

# Detección de Puertas en Nubes de Puntos 3D con Color para Navegación Autónoma en Interiores

B. Quintana<sup>1</sup>, S. A. Prieto<sup>1</sup>, A. Adán<sup>1</sup>, F. Bosché<sup>2</sup>

<sup>1</sup> 3D Visual Computing and Robotics Lab, Universidad de Castilla-La Mancha. Ciudad Real, España.  
{Blanca.Quintana, Samuel.Prieto, Antonio.Adan}@uclm.es

<sup>2</sup> School of Energy, Geoscience, Infrastructure and Society, Heriot-Watt University. Edinburgh, U. K.  
f.n.bosche@hw.ac.uk

## Resumen

*Este artículo presenta un novedoso método capaz de detectar puertas, tanto cerradas como abiertas mediante procesamiento de nubes de puntos adquiridas con un láser-escáner 3D. La técnica propuesta utiliza información geométrica (coordenadas XYZ) e información de color (RGB/HSV) proporcionada por un conjunto calibrado de un láser-escáner 3D y una cámara digital, por lo que podría decirse que nuestro método está desarrollado sobre un espacio 6D. Esta característica hace que sea lo suficientemente robusto ante oclusión y leves cambios sobre el color o la iluminación de la escena. El método ha sido probado tanto en escenarios reales como simulados proporcionando resultados prometedores.*

**Palabras Clave:** Modelado de interiores, Posicionamiento de puertas 3D, Procesado de datos 3D, Modelos de edificios 3D.

## 1. INTRODUCCIÓN

La detección de puertas es una funcionalidad que un robot debe disponer para llevar a cabo tareas de navegación autónoma en edificios (ya que debe pasar por puertas de una habitación a otra) o manipulación (ya que debe empujar puertas entreabiertas o interaccionar con un tirador de la puerta). El trabajo presentado forma parte de un proyecto mayor, cuyo objetivo es desarrollar un sistema autónomo capaz de digitalizar interiores de edificios habitados. Por lo tanto, el sistema debe tratar con escenarios complejos compuestos por varias habitaciones colindantes en una planta de un edificio. En general, el proceso para escanear (y modelar) dichos escenarios es el siguiente: 1) el robot realiza un escaneado completo de la primera habitación, 2) se reconoce una puerta de paso (abierta o cerrada), 3) se interacciona con la puerta para abrirla si fuera necesario, 4) el robot pasa por la puerta y se sitúa en una posición cercana con el objetivo de adquirir el primer escaneo de la siguiente habitación. Como puede comprobarse, la detección de puertas es un elemento clave en este proceso.

A pesar de que la detección de puertas ya se ha tratado anteriormente por otros autores, este artículo presenta un método único con las siguientes características: (1) proporciona de manera precisa tanto el tamaño como la localización de la puerta en un sistema mundo de coordenadas 3D; (2) es válido tanto para puertas abiertas como cerradas; (3) es capaz de tratar con oclusión; y (4), usa información geométrica en 3D e información de color, lo que le permite detectar y localizar puertas de manera robusta.

## 2. ESTADO DEL ARTE

La facilidad de uso y el bajo coste de los sistemas de visión artificial han hecho que los sistemas basados en imágenes 2D sean muy comunes en aplicaciones de detección de puertas ([1]-[6]). Por ejemplo, Xiaodong et al. [1] definen un modelo 2D de un marco de puerta, compuesto por dos líneas horizontales y dos verticales entre cuatro esquinas. Después, dicho modelo es utilizado para buscar composiciones similares dentro de la imagen. Dado que el algoritmo está basado en características geométricas, es invariable ante cambios de color e iluminación y es capaz de detectar puertas abiertas y cerradas, así como puertas de cristal. Marwa M. Shalaby et al. [2] usan un método similar, basado igualmente en un modelo 2D geométrico. Sin embargo, únicamente muestran resultados con puertas cerradas. Andreopoulos et al. [3] también hacen uso de una técnica basada en detección de bordes y esquinas. Su método reconoce la puerta, detecta y localiza el tirador e identifica el estado de la puerta (abierta o cerrada). Para detectar y localizar el tirador, se usa un algoritmo de aprendizaje entrenado con un conjunto de 1,500 ejemplos. Wei Chen et al. [4] proponen un método basado en aprendizaje con una red neuronal convolucional. No mencionan ni hacen ningún tipo de referencia de si su método es capaz de detectar puertas entreabiertas o completamente abiertas. La propuesta de Soohwan Kim et al. [5] está basada en la detección de un tirador para poder identificar una puerta, por lo que su método no es válido para puertas abiertas. Finalmente, Rafiq Sekkal et al. [6] parten de la hipótesis de que se encuentran en un corredor y que, por lo tanto, son capaces de extraer las líneas correspondientes a la intersección entre las

paredes con el suelo y el techo. Con estas líneas detectadas en la imagen, proceden a extraer el punto de fuga de la escena, con el que generan un modelo 3D aproximado de la misma. Una vez que se dispone del modelo 3D, el algoritmo busca dos líneas consecutivas ubicadas en el plano de la pared, para después verificar si ambas líneas se encuentran separadas una determinada distancia impuesta a-priori e identificar dicho par de líneas como una puerta. Según los autores, este método “ad-hoc” es capaz de detectar tanto puertas abiertas como cerradas – lo cual sería algo discutible si se eliminan las fuertes hipótesis iniciales.

Los métodos basados en imágenes 2D generalmente son ineficientes y producen una alta cantidad de falsos positivos debido a la presencia de otros objetos en las paredes (como por ejemplo ventanas o cuadros) o a la de muebles cercanos a las paredes (por ejemplo, radiadores o estanterías). Para reconocer puertas con una mayor precisión y fiabilidad, algunos métodos optan por usar sensores 3D y procesar nubes de puntos (en algunos casos también junto con imágenes 2D) ([7]-[16]). Goron et al. [7] obtienen nubes de puntos por medio de un sensor láser de rango (LRF en acrónimo inglés), de las cuales extraen planos correspondientes a puertas cerradas aplicando una técnica de RANSAC (Random Sample Consensus). Esta no es una técnica demasiado robusta dado que está basada en la hipótesis de que las puertas no son coplanarias con la pared, lo que generalmente no es cierto. Con una cámara RGB-D para la toma de datos, Sebastian Meyer et al. [8] segmentan la nube de puntos en sub-regiones planas con el uso de un algoritmo de crecimiento de regiones basado en los vectores normales de los puntos. La puerta es identificada si y solo si las dimensiones del plano detectado coinciden con unas dimensiones estándar predefinidas y si dicho plano contiene un tirador. Este método está optimizado para detectar puertas cerradas y de una sola hoja. Karthik Mahesh et al. [9] proponen un método centrado en la detección de puertas abiertas. Para ello, obtienen información 3D con una cámara estéreo. Tras extraer los posibles planos verticales por medio de RANSAC, se buscan las puertas en los huecos existentes en dichos planos (es decir, regiones con ausencia de datos). El método presentado por Ting Han Yuan et al. [10] es capaz de identificar puertas abiertas y entreabiertas, haciendo uso de la información de profundidad de una pared proporcionada por una cámara RGB-D. El ángulo de apertura de la puerta es calculado a través de la forma del hueco que se genera entre la pared y la puerta, por lo que asumen que el sensor estará colocado perpendicularmente a la puerta. De manera similar, Matthew Derry y Brenna Argall [11] utilizan un sensor RGB-D para identificar huecos en las paredes correspondientes a puertas. DaWei Dai et al. [12] se basan en la misma idea, con el mismo tipo de sensor

montado en un robot, para detectar la puerta y la posición relativa del robot con respecto a ésta. Esta información es usada posteriormente en la navegación del robot a la hora de atravesar la puerta. Adiwahono et al. [14] proponen algo diferente. A partir de los datos obtenidos con un LRF en 2D, y asumiendo que la puerta no se encuentra en el mismo plano que la pared, seleccionan una serie de candidatos a puerta. Cada candidato se compone de un conjunto de puntos que se ajustan a una línea recta de una determinada longitud. A continuación, el robot se aproxima a cada uno de los candidatos y procede a escanearlos con una cámara RGB-D. Si en el escaneo se detecta un tirador (haciendo coincidir la nube de puntos con un modelo previamente conocido) el candidato pasa a ser identificado como una puerta. Diaz-Vilariño et al. [15] llevan a cabo la extracción de puertas cerradas aplicando la Transformada de Hough Generalizada sobre ortoimágenes, provenientes de una nube de puntos con información de color obtenida con un escáner 3D. Este método, basado en la detección de rectángulos, solo es capaz de detectar puertas cerradas. Banerjee et al. [16] desarrollan un sistema para su participación en el DARPA Robotics Challenge, permitiendo a un robot Atlas detectar puertas, así como abrirlas. En primer lugar, pares consecutivos de líneas verticales son detectados en una imagen 2D de la escena (usando un detector de bordes Canny). Posteriormente, las líneas son recalculadas en un espacio 3D con la ayuda de la técnica RANSAC. Si entre un par de líneas existe una superficie plana, dicho conjunto es etiquetado como una puerta. La detección del tirador es llevada a cabo por medio de segmentación de color, bajo la hipótesis de que el tirador posee un color completamente diferente al de la puerta.

### 3. CONTEXTO EN LA DETECCIÓN DE PUERTAS

Como ha sido mencionado en la Sección I, la detección de puertas es una de las muchas tareas que lleva a cabo nuestro sistema para generar modelos geométricos 3D de interiores. El robot, con un láser-escáner 3D montado sobre él, se mueve autónomamente con la ayuda de un sistema de posicionamiento local y de un algoritmo Next-Best-Scan (NBS). El algoritmo de NBS calcula las coordenadas de la siguiente mejor posición a la que debe desplazarse el robot para tomar un nuevo escaneo. El algoritmo ha sido diseñado teniendo en cuenta el objetivo final del proceso de escaneado, que es la adquisición del mayor número posible de datos de elementos estructurales (SEs por sus siglas en inglés) visibles del edificio, que son suelo, techo y paredes. Cuando el sistema considera que una habitación ha sido completamente escaneada se ejecuta el algoritmo de detección de puertas, permitiendo de esta manera que el robot pueda acceder

al resto de habitaciones que deben ser digitalizadas. El sistema propuesto tiene una gran relevancia en el área de la generación de modelos semánticos 3D de interiores habitualmente conocidos como *as-is/as-built 3D Building Information Models* (BIMs) en nomenclatura inglesa [17].

Antes de que el robot se desplace a una habitación contigua, nuestro sistema genera un modelo semántico 3D simplificado en un espacio voxelizado etiquetado, con un tamaño de vóxel de  $20 \times 20 \times 20 \text{ cm}^3$ , de la habitación en la que se encuentra actualmente. En [18] se pueden encontrar detalles acerca del espacio voxelizado y del algoritmo NBS. Esto quiere decir que las partes esenciales de los elementos estructurales de la escena (es decir, los SEs) son reconocidos y están disponibles en el sistema de coordenadas global, siendo así posible utilizar los datos asociados a las paredes modeladas en el proceso de detección de puertas. Concretamente, las entradas para nuestro algoritmo de detección de puertas son: el conjunto de vóxeles que intersectan los polígonos de las paredes - que pueden tener etiquetas de Ocupados, Ocluidos y Aperturas (para ésta hemos usado la palabra anglosajona *Openings*) -, y la nube de puntos 3D con color asociada a esos vóxeles. La sección siguiente explica el significado de estas etiquetas.

## 4. DETECCIÓN DE PUERTAS ABIERTAS

### 4.1 DETECCIÓN DE VÓXELES OPENING

Nuestro algoritmo NBS es ejecutado en un espacio buna nueva habitación. El espacio voxelizado, junto con sus respectivas etiquetas, es actualizado cada vez que se ejecuta el algoritmo NBS desde una nueva posición del escáner. El algoritmo detecta el suelo, el techo y las paredes (SEs) en el espacio discretizado y asigna automáticamente a cada uno de los vóxeles una de las siguientes etiquetas:

- Los vóxeles ocupados pueden tener una de las siguientes tres etiquetas (hemos mantenido nomenclatura anglosajona en algunos casos que no tienen una traducción admisible en castellano):

- \* *Clutter*: el vóxel contiene puntos que no pertenecen a ningún SE;

- \* *Estructura-ocupado*: el vóxel contiene puntos pertenecientes a un SE;

- \* *Outlier*: el vóxel contiene puntos situados fuera del límite detectado de la habitación.

- Los vóxeles no-ocupados pueden ser etiquetados de la siguiente forma:

- \* *Clutter-ocluído*: El vóxel no ha sido visto desde ninguna posición (porque ha estado siempre

ocluído por un vóxel ocupado) y no pertenece a ningún SE.

- \* *Estructura-ocluído*: El vóxel no ha sido visible desde ninguna posición y pertenece a un SE.

- \* *Vacío*: El vóxel ha sido visto (es decir, no ha sido ocluido por un vóxel ocupado desde alguna de las posiciones del robot), y no pertenece a un SE.

- \* *Opening*: El vóxel no ocupado ha sido visible y pertenece a un SE.

En la Figura 1 se puede ver un ejemplo de un espacio etiquetado.

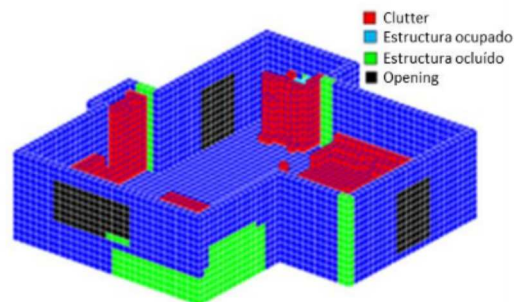


Figura 1: Espacio voxelizado y etiquetado de una habitación.

## 4.2 RECONOCIMIENTO Y POSICIONAMIENTO DE PUERTAS ABIERTAS

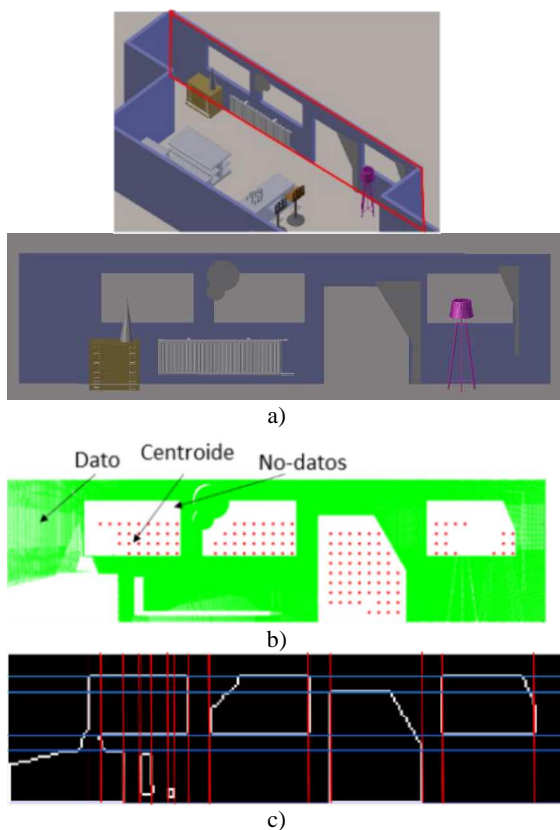
El algoritmo de detección de puertas se centra en la detección del área de las paredes (los polígonos correspondientes a cada pared y los vóxeles y puntos 3D asociados). El etiquetado de los vóxeles nos ayuda a delimitar aproximadamente la posición de las aperturas en las paredes, que son posteriormente procesadas para reconocer puertas abiertas (entre otros tipos de aperturas). Si la pared no contiene vóxeles *Opening*, entonces el sistema procede con el reconocimiento de puertas cerradas (ver Sección 5).

Para localizar de manera precisa las aperturas de la pared en el espacio 3D no es suficiente sólo con considerar los vóxeles, sino que hay que tener en cuenta el conjunto  $S$  de todos los puntos escaneados asociados a la pared. El primer paso es calcular el polígono  $R$  que mejor se ajusta al conjunto  $S$  mediante un algoritmo RANSAC. A continuación,  $S$  y los centroides de los vóxeles *Opening* son proyectados sobre  $R$ , convirtiendo  $R$  en una imagen 2D  $I$  que contiene centroides de los vóxeles, los puntos (datos) y el resto de área sin datos (no-datos). Es importante remarcar que los píxeles correspondientes a zonas sin datos pueden ser debidos a oclusiones o a aperturas (como puertas o ventanas) en la pared. En la Figura 2 b) se muestra un ejemplo de la imagen  $I$ .

Asumiendo que las puertas son rectangulares, lo cual es una hipótesis razonable, el primer paso del

algoritmo propuesto para la detección de puertas abiertas consiste en la extracción de un conjunto de rectángulos candidatos en  $I$ . Para ello, lo primero es extraer líneas horizontales y verticales en la imagen  $I$ , con una técnica basada en histogramas laterales, y considerar todos los rectángulos formados por la intersección de pares de líneas verticales con pares de líneas horizontales (Figura 2 c)). Esto se realiza ya que asumimos que puede haber oclusión en la pared. El conjunto de rectángulos obtenido es reducido tras imponer una serie de restricciones de tamaño. A continuación, se lleva a cabo un agrupamiento de los centroides de los vóxeles por medio de un algoritmo de crecimiento de regiones. Para cada uno de los grupos obtenidos, se calcula qué rectángulo es el mejor candidato, eligiendo de entre todos los rectángulos que contienen el mayor número de centroides (pueden ser varios), aquel que tiene menor área. Esto permite identificar la apertura que mejor se ajusta a cada clúster de centroides. Por último, sólo aquellos rectángulos que tienen uno de sus lados al nivel del suelo son considerados como candidatos a puerta. En la Figura 2 d) y 2 e) se muestran una figura de los centroides agrupados y las aperturas detectadas.

Después de llevar a cabo el procedimiento descrito en el párrafo anterior, los rectángulos elegidos en la imagen  $I$  se convierten en polígonos verticales en el plano de la pared, posicionando así todas las aperturas, incluidas las puertas, en el espacio 3D de coordenadas globales.



## 5 DETECCIÓN DE PUERTAS CERRADAS

El espacio voxelizado aporta información esencial para la detección de puertas abiertas, pero no para la detección de puertas cerradas. Esto es debido a que normalmente éstas son co-planares con la pared y aparecerían, al igual que el resto de la pared, etiquetadas como vóxeles *Ocupados*. Para detectar puertas cerradas se ha desarrollado un algoritmo de detección 4D (color+profundidad) que es capaz de funcionar en una amplia variedad de casos, incluyendo aquellos en los que no existen colores completamente uniformes en los objetos contenidos en la pared. Esto se debe a que para conseguir un modelo completo de la escena, el sistema recoge datos (puntos+color) desde diferentes posiciones, pudiendo existir variaciones en las condiciones de luz para cada una de las posiciones. Esto produce cambios de color en la nube de puntos completa, que da lugar a colores no-uniformes en las distintas paredes.

Para el desarrollo del algoritmo se han hecho una serie de hipótesis, la cuales se cumplen en la gran mayoría de los casos:

1. Las paredes son planas (esta restricción se impone en el paso de reconocimiento de SEs y modelado).
2. Las paredes tienen un color relativamente homogéneo, siendo aceptables algunas variaciones de color resultantes del fenómeno que se ha descrito anteriormente.
3. Las puertas están definidas por rectángulos con lados verticales y horizontales (es la misma restricción que se realizó para el caso de puertas abiertas).

Las entradas del algoritmo propuesto son: el plano de la pared calculado mediante RANSAC (como se discutió en la Sección 4) y los puntos 3D con información de color asociados al plano. El algoritmo

se divide en dos pasos: la segmentación de las áreas visibles de la pared y la detección de la puerta.

## 5.1 SEGMENTACIÓN DE LAS ÁREAS VISIBLES DE LA PARED

Utilizando los datos de entrada, lo primero que se hace es crear una ortoimagen RGB-D  $J_{CP}$  de la pared, donde cada píxel tiene asignado un color RGB y una distancia ortogonal a la pared. A continuación se detalla el proceso de segmentación de las áreas visibles de la pared que es llevado a cabo sobre esta imagen (proceso ilustrado en la Figura 3).

En primer lugar se realiza un muestreo de la imagen  $J_{CP}$  utilizando una rejilla de  $20 \times 20$  con pequeñas ventanas cuadradas de  $5 \times 5$  píxeles de tamaño. Para cada ventana se analiza la distribución de los valores RGB-D del conjunto de píxeles en su interior  $\{v\}$ , y aquellas para las que la desviación estándar de alguno de los cuatro componentes sea mayor que un determinado umbral (en nuestro caso 0.20) son descartadas. Este proceso asegura que sólo las ventanas que son *coherentes* tanto en el dominio del color como en el de la profundidad (por ejemplo, la ventana no está localizada en el borde de un marco) se mantienen (ver Figura 3 b)).

Cada ventana *coherente*  $m$  se representa por el valor medio de los valores RGB-D de sus 25 píxeles  $\bar{v}_m$ , y a continuación se emplea un algoritmo adaptativo *k-means* para agrupar las ventanas  $\{\bar{v}_m\}$  en  $k$  grupos, siendo  $k$  calculado por el propio algoritmo (ver Figura 3 c)). Posteriormente se mejora la consistencia dentro de cada grupo eliminando cualquier ventana que tenga un valor *silhouette*  $\delta$  mayor que un valor umbral razonable ( $|\delta| > 0.7$ ). El valor *silhouette* para un miembro de un grupo es una medida que indica cómo de similar es ese miembro al resto de miembros del grupo, en comparación con los miembros de los otros grupos. Los valores de *silhouette* varían en un rango de -1 a +1 (ver Figura 3 d)).

Finalmente se identifica el conjunto de píxeles de  $J_{CP}$  asociados con el  $i$ -ésimo clúster  $\{\bar{v}_m\}_{i,i=1..k}$  mediante una técnica de asociación (*matching*) impuesta en las cuatro componentes. La Figura 3 e) muestra los resultados finales con tres imágenes binarias que representan los píxeles (en blanco) asociados a cada segmento. Por último, se establece que el área de la pared es el clúster que contiene el mayor número de píxeles localizados en el borde de la imagen.

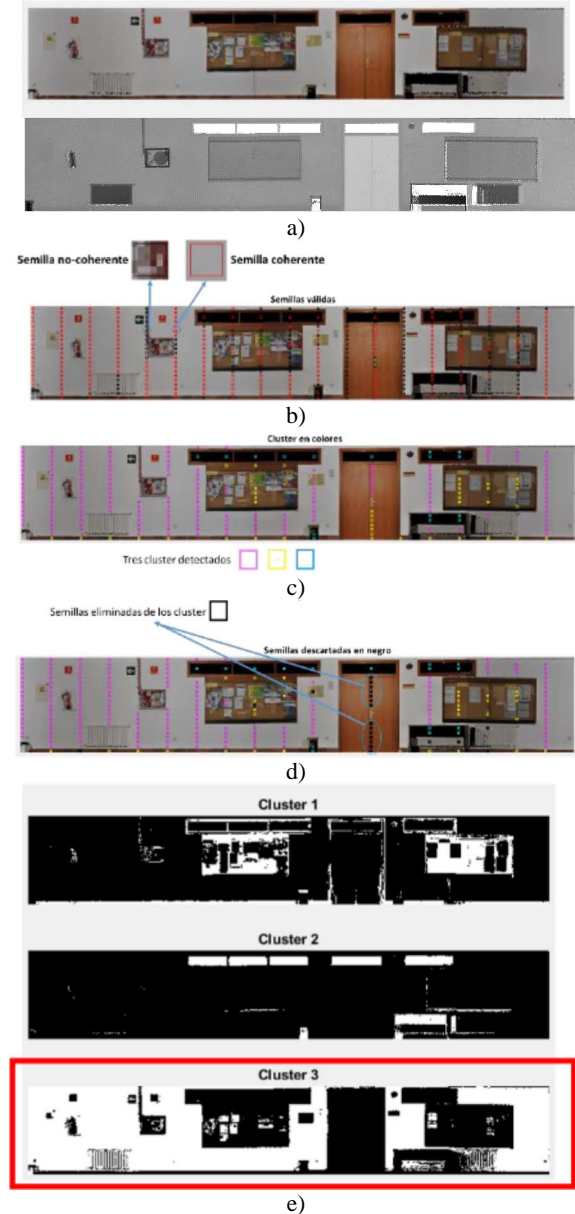


Figura 3: Detección del área de la pared. a) Imagen RGB e imagen de profundidad. b) Búsqueda de ventanas coherentes (llamadas semillas en la figura). c) Grupos de semillas. d) Eliminación de semillas no consistentes de los grupos. e) Identificación de segmentos y del área de la pared (marcada en rojo).

## 5.2 ALGORITMO DE RECONOCIMIENTO

La detección de puertas cerradas es una tarea especialmente compleja cuando la puerta está parcialmente ocluida o cuando las paredes contienen una gran cantidad de objetos de diferentes colores. Utilizando nuestro espacio de color-profundidad se consigue ser más eficiente bajo estas situaciones.

Para llevar a cabo la detección de puertas cerradas presentamos un método basado en discontinuidades en el espacio 4D RGB-D. En nuestra propuesta, tanto la

componente de color como la de profundidad se procesan por separado, con la imagen  $J_{CP}$  descompuesta en dos imágenes  $J_C$  (color) y  $J_P$  (profundidad), recombinando a continuación los resultados. Primero se aplica un operador gradiente a la imagen  $J_C$  que calcula el máximo ratio de cambio en el píxel (gradiente) a lo largo de las dimensiones espectrales [19]. Esto es seguido de un proceso de binarización de la imagen utilizando la técnica de umbralización global de Otsu. El método de Otsu selecciona el valor umbral que minimiza la varianza intraclase de los píxeles blancos y negros. Como resultado de este proceso se obtiene la imagen binaria  $J'_C$  (Figura 4 a)). A la imagen  $J_P$  (profundidad) se le aplica un detector de bordes Canny, obteniendo así una segunda imagen binaria  $J'_P$ . Por último, se combinan (con un operador OR) las imágenes  $J'_C$  y  $J'_P$  formando una única imagen gradiente  $J'_{CP}$ . La Figura 4 ilustra este proceso.

En la imagen  $J'_{CP}$  los píxeles blancos representan discontinuidades en el espacio color-profundidad. Esto permite la detección de los marcos de las puertas como discontinuidades en el dominio de color, en el de profundidad, o en ambos. Para ello, detectamos líneas rectas en  $J'_{CP}$  (utilizando el mismo método que en la Sección 4). Estas líneas contienen las discontinuidades de color-profundidad de la pared, y por ello, también contienen partes de los contornos de las hipotéticas puertas. La palabra “parte” se utiliza debido a que pueden existir oclusiones (las cuales están permitidas en nuestro contexto).

De forma similar a lo que se hacía para la detección de puertas abiertas, se calculan todos los posibles rectángulos definidos por dos pares de líneas verticales y horizontales. Dado que se están buscando rectángulos que delimiten puertas, se impone que uno de los lados del rectángulo debe estar situado a la altura del suelo, además de ciertas restricciones en cuanto al tamaño de los rectángulos. Estas restricciones disminuyen el tamaño del conjunto de rectángulos  $\{r\}$  que va a ser evaluado.

Un rectángulo de  $\{r\}$  es considerado como una puerta (marco de la puerta) si cumple las siguientes condiciones:

1. Consistencia de color y profundidad. Tras haber aplicado un proceso de agrupamiento adaptativo *k-means* sobre la imagen de color y profundidad en  $r$ , establecemos que el color y la profundidad dominantes deben cubrir por separado un cierto porcentaje  $\alpha_1$  del área de la puerta. Se ha fijado este valor a  $\alpha_1=55\%$ .
2. Oclusión del marco de la puerta. Un porcentaje  $\alpha_2$  de la longitud de cada uno de los lados de  $r$  debe contener información de discontinuidad, es decir, píxeles blancos de  $J'_{CP}$ . El valor umbral ha sido fijado a  $\alpha_2=70\%$ . Por lo tanto se permite un 30% de oclusión.
3. Consistencia de localización. El área encerrada por el rectángulo no interseca el área de pared

identificada en el proceso descrito en la sección 5.2, con un margen de  $\alpha_3=3\%$ .

4. El rectángulo no es contenido por ningún otro rectángulo que verifique las condiciones 1, 2 y 3.

Los parámetros  $\alpha_1$ ,  $\alpha_2$  y  $\alpha_3$  han sido definidos empíricamente mediante experimentación con 40 casos distintos con una oclusión de hasta el 50%. La Figura 5 ilustra las salidas en cada paso de la segunda parte del algoritmo de detección de puertas cerradas. En este caso, la única puerta existente ha sido correctamente detectada.

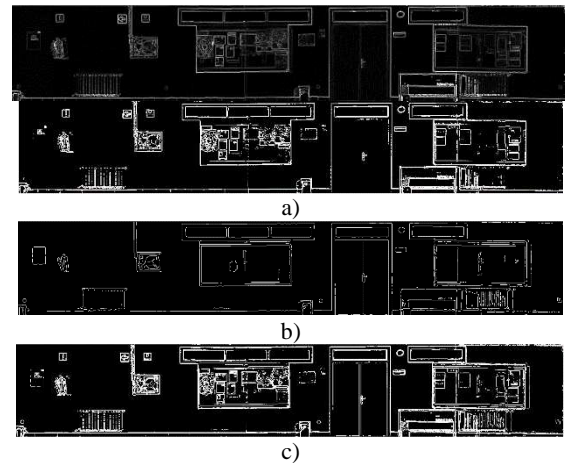


Figura 4: Generación de imagen de discontinuidad combinada. a) Imagen de gradiente  $J_C$  y  $J'_C$ . b)  $J'_P$ . c) Imagen integrada  $J'_{CP}$ .

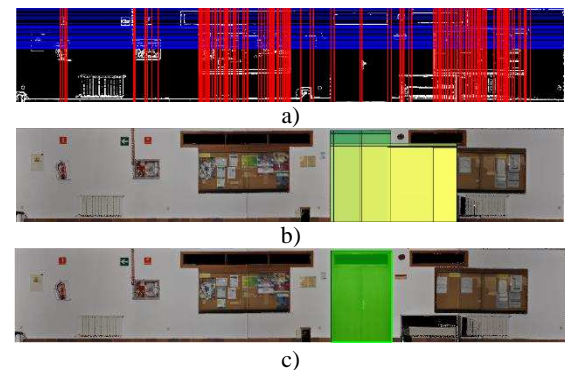


Figura 5: Ejemplo de detección de puertas. a) Líneas verticales y horizontales detectadas en  $J'_{CP}$ . b) Conjunto de rectángulos  $\{r\}$ . Los rectángulos candidatos están superpuestos en la imagen en diferentes colores. c) Rectángulo que contiene la puerta en color verde.

## 6. EXPERIMENTACIÓN

### 6.1 TEST DEL MÉTODO EN UN ENTORNO SIMULADO

Para medir la eficiencia y exactitud del método presentado se han llevado a cabo una experimentación en un entorno simulado. El escenario ha sido generado mediante la herramienta software Blender, mientras que la simulación del escáner y la toma de datos se ha

realizado con su *add-on* Blensor [20]. Este *add-on* permite la simulación de un escáner 3D similar al nuestro (Riegl VZ-400), con lo que se consigue trabajar, en la medida de lo posible, en las condiciones reales. Los datos presentados en esta sección son el resultado de aplicar el algoritmo al escenario mostrado en la Figura 6 a). Dado que se dispone de las dimensiones exactas del modelo, ha sido posible presentar una evaluación de la precisión del método.

El escenario está compuesto de 5 habitaciones adyacentes, conectadas mediante 6 puertas y un total de 9 ventanas. El tamaño de la escena es de 27m × 21m. El total de elementos estructurales es de 66 paredes, 5 techos y 5 suelos. El mobiliario y otro tipo de elementos (cortinas, carteles, cuadros, ...) han sido insertados como un recurso para introducir “clutter” y oclusión a la escena. La oclusión media conseguida en elementos estructurales es de un 14%.

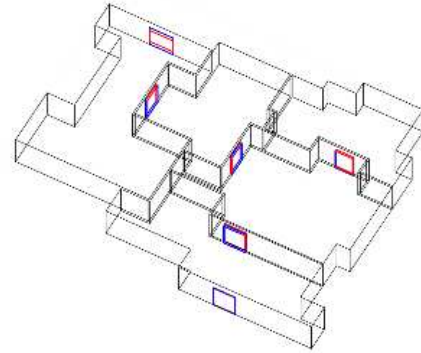
El método propuesto en este artículo ha sido capaz de detectar todas las puertas. La Figura 6 b) muestra las puertas reconocidas (en rojo) superpuestas a su posición exacta en el modelo (en azul). Puede apreciarse como, en algunos casos, existe una pequeña diferencia entre ambos rectángulos. Es por ello por lo que es necesario cuantificar la precisión a la hora de obtener la posición de las puertas.

Los conceptos *Precision* y *Recall* son bien conocidos para evaluar reconocimiento y posicionamiento de objetos. En este caso, usaremos de manera similar dichas medidas para evaluar el nivel de superposición entre los rectángulos equivalentes a la posición exacta de la puerta y a los calculados por el algoritmo. Sean Q y G las áreas de dichos rectángulos, respectivamente. El valor que representa el caso verdadero-positivo ( $t_p$ ) es evaluado como la intersección entre Q y G, mientras que el caso falso-positivo ( $f_p$ ) es evaluado como el área de Q no contenida en G. Por otro lado, el área de G no contenida en Q es considerada como falso-negativo ( $f_n$ ). En la Figura 7 a) puede observarse una representación gráfica de dichos conceptos.

Por lo tanto, *Precision* se define como la fracción de puerta detectada que verdaderamente se corresponde con una puerta real (Ecuación (1)), mientras que *Recall* se corresponde con la fracción de puerta real que ha sido correctamente detectada (Ecuación (2)).



a)



b)

Figura 6. a) Modelo 3D sobre el que ha sido probado el método. b) Resultados de la detección de puertas. Los rectángulos correspondientes a la situación exacta de las puertas se representan en azul, mientras que los calculados por el método están en rojo.

Una medida que combina estas dos cantidades es la media armónica de ambas, *F-measure* ( $F_\beta$ ). Este valor estadístico podría considerarse como una media ponderada de ambos valores (Ecuación (3)).

$$Precision = \frac{t_p}{f_p + t_p} \quad (1)$$

$$Recall = \frac{t_p}{f_n + t_p} \quad (2)$$

$$F_\beta = (1 + \beta^2) \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \quad (3)$$

Además de *Precision* y *Recall* se ha añadido un segundo conjunto de medidas para evaluar el error de modelado. Los errores para cada una de las puertas detectadas han sido considerados de manera separada como la suma de  $f_p$  y  $f_n$ . El error absoluto y relativo quedan recogidos en las ecuaciones (4) y (5).

$$e_{abs} = f_p + f_n \quad (4)$$

$$e_{rel} = \frac{f_p + f_n}{f_p + t_p} \quad (5)$$

Los valores obtenidos en la evaluación del método sobre el caso de ejemplo son los siguientes. Los valores medios de *Precision* y *Recall* son de 0.979 y 0.949, respectivamente. En la Figura 7 b) puede observarse un gráfico que ilustra con más detalle los resultados para cada una de las puertas. Con respecto a los errores, los valores medios han sido  $e_{abs}=0.259$  m<sup>2</sup> y  $e_{rel}=0.081$ . Como resultado de estos valores, se puede decir que el método detecta eficientemente y con una alta precisión las puertas de la escena.

Se ha calculado la media armónica  $F_\beta$  para un intervalo  $\beta=[1/3, 3]$ . Los resultados quedan especificados en el gráfico de la Figura 7 c). Como puede observarse,  $F$  decrece a medida que  $\beta$  también decrece. Para un valor equilibrado de  $\beta = 1$ ,  $F_\beta=0.988$ .

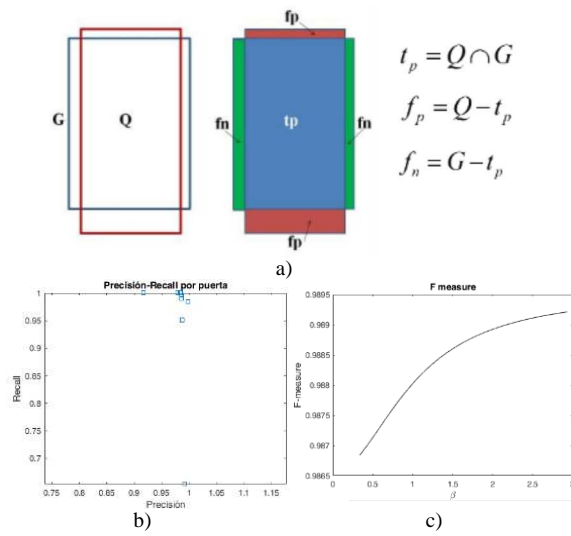


Figura 7. a) Definición de los parámetros  $t_p$ ,  $f_p$  y  $f_n$ . b) Resultados de *Precision* (eje X) y *Recall* (eje Y). c) Evolución de *F-measure* en el rango  $\beta=[1/3, 3]$

## 6.2 DETECCIÓN DE PUERTAS CERRADAS CON UNA PLATAFORMA MÓVIL

Esta sección está dedicada a mostrar los resultados obtenidos en la detección de puertas cerradas con nuestra plataforma experimental.

Se ha implementado un sistema de escaneo autónomo en la plataforma MoPAD (*Mobile Platform for Autonomous Digitization*), compuesta por un robot móvil (Robotnik Guardian) equipado con un láser-escáner 3D (Riegl VZ-400) y 2 LRF (Hokuyo URG-04LX-UG01). El sistema se muestra en la Figura 8. La experimentación ha sido llevada a cabo en la planta de la Escuela Superior de Informática de la Universidad de Castilla-La Mancha. Las dimensiones del escenario son de 8m  $\times$  34m.

El escenario está compuesto por 29 elementos estructurales con 18 puertas. En la experimentación, el

robot es manualmente dirigido a cada una de las habitaciones, procediendo a cerrar las puertas una vez que el robot se encontraba en su interior. El algoritmo NBS calcula las sucesivas posiciones que el robot debe alcanzar en la habitación para realizar una completa digitalización de la escena. A continuación, la nube de puntos es procesada y se reconocen y delimitan todos los elementos estructurales. Cada elemento correspondiente con una pared, junto con el conjunto de puntos (con color) asociado a dicho elemento, son almacenados y enviados al algoritmo de detección de puertas.

El reconocimiento de puertas funcionó correctamente para el 100% de los casos (18/18). La Figura 8 muestra algunas de las paredes y puertas identificadas en la experimentación, algunas de ellas con alta oclusión (paredes 10 y 12 con un 38.3% y 38.8% de oclusión). La Figura 9 muestra los resultados correspondientes a un caso más complejo con dos puertas coplanarias a la pared y con un color muy similar a ésta.

Para evaluar los resultados obtenidos, ha sido necesario construir manualmente un modelo fiable y representativo del escenario (el modelo *ground truth* en nomenclatura anglosajona). Los resultados de *Precision*, *Recall*, error absoluto y relativo están recogidos en la Tabla I. En general, el método proporciona resultados muy alentadores. Los valores medios obtenidos fueron  $Precision=0.994$ ,  $Recal=0.974$ ,  $e_{abs}=0.0928$  (m<sup>2</sup>)  $e_{rel}=0.0331$  lo que demuestra la precisión de las estimaciones calculadas en la posición y tamaño de las puertas detectadas.

TABLA I. RESULTADOS EXPERIMENTALES

| Pared | <i>Precision</i> | <i>Recall</i> | $e_{abs}$ (m <sup>2</sup> ) | $e_{rel}$ |
|-------|------------------|---------------|-----------------------------|-----------|
| #1    | 0,9930           | 0,9957        | 0,0375                      | 0,0113    |
| #2    | 1,0000           | 0,9888        | 0,0376                      | 0,0114    |
| #3    | 1,0000           | 0,9917        | 0,0370                      | 0,0084    |
| #4    | 1,0000           | 0,9849        | 0,0570                      | 0,0153    |
| #5    | 1,0000           | 0,8771        | 0,4052                      | 0,1402    |
| #5    | 1,0000           | 0,8604        | 0,4660                      | 0,1623    |
| #6    | 0,9897           | 0,9951        | 0,0301                      | 0,0151    |
| #7    | 0,9938           | 0,9918        | 0,0562                      | 0,0144    |
| #8    | 0,9912           | 0,9949        | 0,0309                      | 0,0139    |
| #8    | 1,0000           | 0,9760        | 0,0527                      | 0,0245    |
| #9    | 1,0000           | 0,9818        | 0,0624                      | 0,0185    |
| #9    | 0,9587           | 0,9959        | 0,1321                      | 0,0453    |
| #9    | 0,9853           | 1,0000        | 0,0241                      | 0,0147    |
| #10   | 1,0000           | 0,9815        | 0,0430                      | 0,0188    |
| #10   | 1,0000           | 0,9581        | 0,0961                      | 0,0437    |
| #11   | 1,0000           | 0,9863        | 0,0316                      | 0,0138    |
| #11   | 1,0000           | 0,9863        | 0,0315                      | 0,0139    |
| #12   | 0,9945           | 0,9952        | 0,0388                      | 0,0103    |



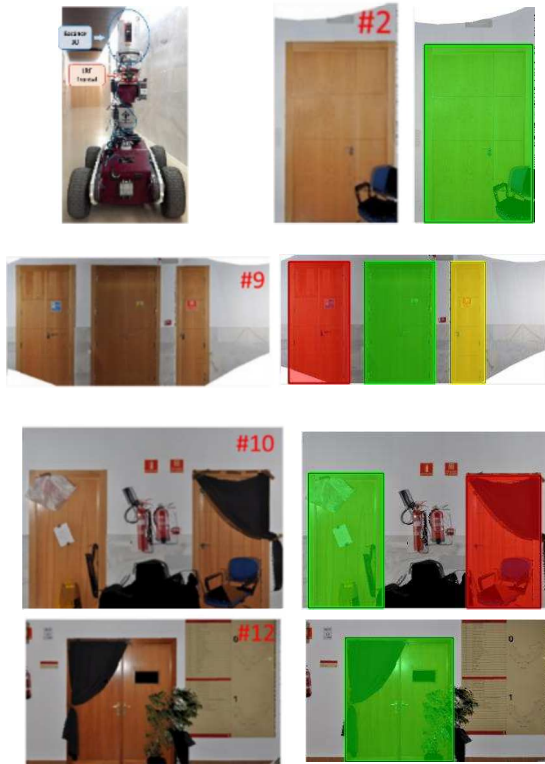


Figura 8. a) MoPAD y ejemplos de puertas detectadas. Paredes 10 y 12 con un 38.3% y 38.8% de oclusión, respectivamente.

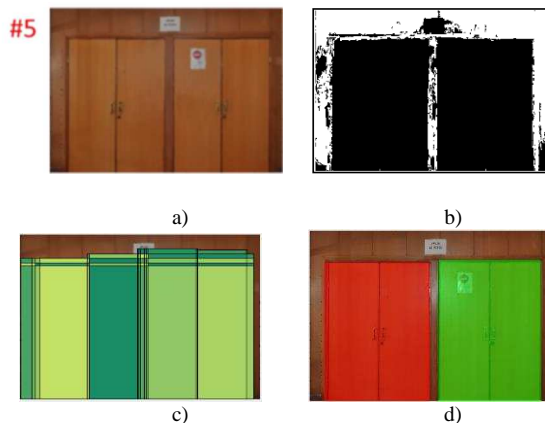


Figura 9. a) Caso complejo: pared coplanaria y de color similar. b) Identificación del área de la pared. c) Posibles rectángulos superpuestos en diferentes colores. d) Rectángulos correspondientes a las puertas, en rojo y verde.

## CONCLUSIONES

En este artículo se ha propuesto un novedoso método para la detección de puertas, tanto cerradas como abiertas, usando nubes de puntos 3D con información de color. El reconocimiento de puertas abiertas está basado en la detección de huecos rectangulares ubicados en las paredes, mientras que la detección de puertas cerradas está basada en la identificación del área visible de la pared y en la posterior búsqueda de áreas rectangulares no pertenecientes a dicha área.

En resumen, el método propuesto en este artículo: (1) proporciona con precisión el tamaño y posición de las puertas en un sistema de coordenadas 3D; (2) proporciona una solución general tanto para puertas abiertas como cerradas; (3) es capaz de reconocer puertas con oclusión; y (4) hace uso de ambas informaciones, geométrica y de color, con el objetivo de proporcionar resultados más robustos ante casos complejos. Se ha validado experimentalmente la eficiencia y precisión del método en un escenario simulado (proporcionando un modelo exacto del mismo) y en un escenario real bajo condiciones de oclusión. Igualmente el método es eficiente ante casos de no uniformidad en el color, que son generados por el solapamiento de varias tomas desde distintos puntos de vista para obtener suficientes datos de la pared. El método muestra resultados alentadores incluso en casos complejos, en los que las puertas se encontraban cerradas, eran coplanarias con la pared y de un color muy similar a ésta. Los futuros trabajos están dirigidos a abordar situaciones más complejas en las que las puertas están entornadas, semiabiertas, de varios colores o con formas no rectangulares.

## Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad del Gobierno Español [proyecto DPI2013-43344-R] y por la Junta de Castilla-La Mancha [proyecto PEII-2014-017-P].

## Referencias

- [1] X. Yang and Y. Tian, "Robust door detection in unfamiliar environments by combining edge and corner features," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, pp. 57–64, 2010.
- [2] M. M. Shalaby, M. A. M. Salem, A. Khamis, and F. Melgani, "Geometric model for vision-based door detection," *Proceedings of 2014 9th IEEE International Conference on Computer Engineering and Systems, ICCES 2014*, pp. 41–46, 2015.
- [3] A. Andreopoulos and J. K. Tsotsos, "Active vision for door localization and door opening using playbot: A computer controlled wheelchair for people with mobility impairments," *Proceedings of the 5th Canadian Conference on Computer and Robot Vision, CRV 2008*, pp. 3–10, 2008.
- [4] W. Chen, T. Qu, Y. Zhou, K. Weng, G. Wang, and G. Fu, "Door recognition and deep learning algorithm for visual based robot navigation," *2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014*, pp. 1793–1798, 2014.
- [5] S. Kim, H. Cheong, D. H. Kim, and S. K. Park, "Context-based object recognition for door detection," *IEEE 15th*

- International Conference on Advanced Robotics: New Boundaries for Robotics, ICAR 2011*, pp. 155–160, 2011.
- [6] Rafiq Sekkal, Franc, Lois Pasteau, Marie Babel, Baptiste Brun, Ivan Leplumcy. “Simple Monocular Door Detection and Tracking” 2013 IEEE International Conference on Image Processing. Melbourne, VIC. pp. 3929–3933, 2013.
- [7] L. C. Goron, L. Tamas, and G. Lazea, “Classification within indoor environments using 3D perception,” *Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on*, pp. 400–405, 2012.
- [8] S. M. Z. Borgsen, M. Schöpfer, L. Ziegler, and S. Wachsmuth, “Automated door detection with a 3D-sensor,” *Proceedings - Conference on Computer and Robot Vision, CRV 2014*, pp. 276–282, 2014.
- [9] K. M. Varadarajan and M. Vincze, “3D room modeling and doorway detection from indoor stereo imagery using feature guided piecewise depth diffusion,” *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 2758–2765, 2010.
- [10] T. H. Yuan, F. H. Hashim, W. M. D. W. Zaki, and A. B. Huddin, “An Automated 3D Scanning Algorithm using Depth Cameras for Door Detection”. *2015 International Electronics Symposium (IES)* pp. 58–61, 2015.
- [11] M. Derry and B. Argall, “Automated doorway detection for assistive shared-control wheelchairs,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1254–1259, 2013.
- [12] D. Dai, G. Jiang, J. Xin, X. Gao, L. Cui, Y. Ou, and G. Fu, “Detecting, locating and crossing a door for a wide indoor surveillance robot,” *2013 IEEE International Conference on Robotics and Biomimetics, ROBIO 2013*, no. December, pp. 1740–1746, 2013.
- [13] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D Point cloud based object maps for household environments,” *Robotics and Autonomous Systems*, 2008.
- [14] A. H. Adiwahono, Y. Chua, K. P. Tee, and B. Liu, “Automated door opening scheme for non-holonomic mobile manipulator,” *International Conference on Control, Automation and Systems*, no. Iccas, pp. 839–844, 2013.
- [15] L. Diaz-Vilariño, K. Khoshelham, J. Martinez-Sanchez, and P. Arias, “3D modeling of building indoor spaces and closed doors from imagery and point clouds,” *Sensors (Switzerland)*, vol. 15, no. 2, pp. 3491–3512, 2015.
- [16] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. Gennert, and T. Padir, “Human-Supervised Control of the ATLAS Humanoid Robot for Traversing Doors,” *IEEE-RAS International Conference on Humanoid Robots*, pp. 722–729, 2015.
- [17] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, “Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques,” *Automation in Construction*, vol. 19, no. 7, pp. 829–843, 2010
- [18] S.A. Prieto, B. Quintana, A. Adán, A.S. Vázquez.” Autonomous as-is Building-Structure Reconstruction from a Probabilistic Scanning Based Robot System”. *Robotics and Autonomous Systems* (Unpublished. 2<sup>nd</sup> revision). <http://isa.esi.uclm.es/index.php/es/publications>.
- [19] Di Zeno, S. "A note on the gradient of a multi-image". *Computer vision, graphics, and image processing*, 33(1), 116-125, 1986.
- [20] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, “BlenSor: Blender sensor simulation toolbox,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6939 LNCS, no. PART 2, pp. 199–208, 2011