

Algoritmo basado en la minimización de la norma  $\ell_1$  para la reconstrucción de una escena tridimensional capturada con el dispositivo Project Tango

Jonathan José Rivera Sinuco

Trabajo de Grado para optar al título de Ingeniero Electrónico

Director:

Henry Arguello Fuentes

PhD. Electrical and Computer Engineering

Codirector:

Samuel Eduardo Pinilla Sánchez

Magíster en Matemáticas

Universidad Industrial de Santander

Facultad de Ingeniería Fisicomecánicas

Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones

Bucaramanga

2017

## Agradecimientos

Me gustaría expresar en estas líneas mis sinceros agradecimientos a todas las personas que con su ayuda han permitido la realización de este proyecto.

En primer lugar, a mi director de proyecto, Dr. Henry Arguello, por permitirme hacer parte de su grupo, y darme la oportunidad de realizar este trabajo.

A mi codirector Samuel Pinilla, por su paciencia, comprensión y su invaluable ayuda. Este proyecto no podría haberse realizado sin sus consejos y asesoría.

A mi familia por su apoyo incondicional. A mis padres por la educación y los valores que me hacen ser quien soy, y a mis hermanos por ser un ejemplo y una guía.

A Jenny Oviedo por su amor y compañía, soportando mis frustraciones y preocupaciones.

Además, quisiera agradecer a todas aquellas personas que directa o indirectamente se han visto involucradas a lo largo de este proyecto. Amigos, compañeros y profesores.

## Tabla de Contenidos

Introducción . . . . .	11
1 Métodos de Escaneo tridimensional . . . . .	14
1.1 Visión estereoscópica . . . . .	14
1.2 Telemetría láser . . . . .	15
1.3 Escáner de luz estructurada . . . . .	17
2 Project Tango . . . . .	19
2.1 Percepción de profundidad . . . . .	19
2.2 Hardware . . . . .	20
3 Muestreo Compresivo . . . . .	23
4 Captura y Procesamiento de los Datos . . . . .	26
4.1 Proceso de captura con Tango . . . . .	26
4.2 Sincronización del color y la profundidad . . . . .	29
4.3 Construcción del cubo de datos . . . . .	30
4.4 Matriz de sensado y decimación . . . . .	35
5 Resultados de la Reconstrucción tridimensional . . . . .	40
5.1 Algoritmo de reconstrucción . . . . .	40
5.2 Simulaciones . . . . .	42
5.2.1 Reconstrucción basado en la minimización Norma $\ell_1$ . . . . .	42
5.2.2 Matrix Completion. . . . .	43
5.3 Método tradicional de reconstrucción. Norma $\ell_2$ . . . . .	45
6 Caracterización del Dispositivo Tango . . . . .	48
7 Conclusiones . . . . .	53
Referencias Bibliográficas . . . . .	54

## Lista de Tablas

Tabla 1	Hardware Dispositivo Project Tango . . . . .	22
Tabla 2	Tabla comparativa para 4 iteraciones . . . . .	46

## Lista de Figuras

Figura 1	Visión Estereoscópica . . . . .	16
Figura 2	Telemetría Láser . . . . .	16
Figura 3	Representación de las cargas en el sensor . . . . .	17
Figura 4	Ejemplo de patrones de luz estructurada . . . . .	18
Figura 5	Montaje escáner de luz estructurada . . . . .	18
Figura 6	Vista interna de los componentes, Adaptado de Sam Lionheart, Walter Galan / Project Tango Tablet Teardown from <a href="https://es.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148">https://es.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148</a> . . . . .	21
Figura 7	Vistazo al interior de la Tango, Adaptado de Sam Lionheart / Project Tango Tablet Teardown from <a href="https://es.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148">https://es.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148</a> . . . . .	21
Figura 8	Exterior del dispositivo Tango . . . . .	22
Figura 9	Muestreo compresivo de imágenes (CS). Aquí $\mathbf{g}$ es el vector de medidas compresivas, $\mathcal{H}$ es la matriz del sistema, $\Psi$ es la base de representación escasa y $\theta$ es la secuencia o vector de coeficientes escasos. . . . .	24
Figura 10	Tango APIs . . . . .	27
Figura 11	Percepción de profundidad del dispositivo Tango . . . . .	27
Figura 12	Proyección bidimensional de una punto tridimensional . . . . .	28
Figura 13	(a) muestra la imagen RGB de la escena, (b) representa la nube de puntos, y (c) la sincronización entre la imagen RGB y la nube de puntos . . . . .	29
Figura 14	Cubo de datos . . . . .	30
Figura 15	Planos $\mathbf{A}_{xy}$ , $\mathbf{A}_{xz}$ y $\mathbf{A}_{zy}$ . . . . .	31
Figura 16	Búsqueda exhaustiva del algoritmo 2 . . . . .	33
Figura 17	Estructura de la matriz $\mathbf{D}$ con $N = 4$ , $M = 4$ , $L = 4$ y $\alpha = 2$ . Donde los puntos blancos representan 1, y los negros 0 . . . . .	36
Figura 18	Imagen intencionalmente mostrada con el fin de ilustrar el proceso de super-resolución. a) Punto capturado, b) Primera Reconstrucción, c) Segunda Reconstrucción, d) Unión del punto capturado y sus reconstrucciones . . . . .	37
Figura 19	(a) Cubo de datos, (b) matriz de sensado $\mathbf{H}$ asociada al cubo de datos. . . . .	38

Figura 20 matriz $\Phi$ que representa el proceso de captura de datos y decimación, donde los puntos blancos representan 1, y los negros 0. . . . .	39
Figura 21 Escenas capturadas por el dispositivo Tango . . . . .	42
Figura 22 Reconstrucción de la escena de la Figura 21a . . . . .	43
Figura 23 Reconstrucción de la escena de la Figura 21b . . . . .	44
Figura 24 Reconstrucción de las escenas usando Matrix Completion . . . . .	45
Figura 25 Reconstrucción de las escenas usando norma euclídea . . . . .	47
Figura 26 Sistema de iluminación utilizado durante la prueba . . . . .	48
Figura 27 Prueba uno, fuente de luz ubicada a uno de los lados del dispositivo . . . . .	49
Figura 28 prueba dos, fuente de luz ubicada frente al dispositivo . . . . .	50
Figura 29 prueba tres, fuente de luz ubicada sobre el dispositivo . . . . .	50
Figura 30 Comparación de los puntos capturados por el dispositivo para la prueba uno . . . . .	51
Figura 31 Comparación de los puntos capturados por el dispositivo para la prueba dos . . . . .	51
Figura 32 Comparación de los puntos capturados por el dispositivo para la prueba tres . . . . .	52

## Resumen

**TITULO:** ALGORITMO BASADO EN LA MINIMIZACIÓN DE LA NORMA  $\ell_1$  PARA LA RECONSTRUCCIÓN DE UNA ESCENA TRIDIMENSIONAL CAPTURADA CON EL DISPOSITIVO PROJECT TANGO.(\*)

**AUTOR:** JONATHAN JOSE RIVERA SINUCO(\*\*)

**PALABRAS CLAVE:** Reconstrucción Tridimensional, Norma  $\ell_1$ , Project Tango.

La reconstrucción de una escena tridimensional tiene como función obtener un modelo digital de la misma, conservando sus propiedades físicas. Este modelo es creado obteniendo una limitada cantidad de información de la escena, la cual se compone de puntos de coordenadas espaciales con sus respectivas intensidades. Este paquete de información llamado nube de puntos, es posteriormente procesado con el fin de producir una imagen digital. Sin embargo, la mayoría de equipos destinados para este trabajo son grandes y costosos, limitando su uso a escenas de gran tamaño. Por tanto, para este trabajo se propone el uso de un moderno dispositivo llamado Project Tango, el cual incorpora un conjunto de tecnologías destinadas al desarrollo de proyectos que interactuen con el mundo real, que además presenta un tamaño y precio reducido. Además se realizaron mediciones con este dispositivo, que fueron posteriormente organizadas en cubos de datos y reconstruidas mediante un algoritmo de optimización que minimice la norma  $\ell_1$  de los datos, obteniendo una escena de mayor resolución en comparación con la escena capturada. Por último se realiza una caracterización del dispositivo Project Tango, determinando la variación de sus mediciones cuando es ubicada una fuente de luz de intensidad variable cerca al dispositivo.

---

(\*) Trabajo de Investigación

(\*\*) Facultad de ingenierías Físico-Mecánicas. Escuela de ingeniería eléctrica, electrónica y telecomunicaciones. Director: Henry Arguello. Codirector: Samuel Pinilla

## Abstract

**TITLE:** THREE-DIMENSIONAL RECONSTRUCTION ALGORITHM MINIMIZING  $\ell_1$  NORM FROM PROJECT TANGO MEASUREMENTS.(\*)

**AUTHOR:** JONATHAN JOSE RIVERA SINUCO.(\*\*)

**KEYWORDS:** 3D Reconstruction,  $\ell_1$  Norm, Project Tango.

The reconstruction of a three-dimensional scene has as an intension to obtain a digital model of the scene, conserving its physical properties. This model is created by obtaining a limited amount of information from the scene, which is composed by points of spatial coordinates with their respective intensities. This information packet called point cloud is subsequently processed in order to produce a digital image. However, most of the equipment intended for this job is large and expensive, limiting its use to large scenes. Therefore, this work proposes the use of a modern device called Project Tango, which incorporates a set of technologies intended for the development of projects that interact with the real world, which also has a size and reduced price. Moreover the captured measurements by this device, were organized into data cubes and reconstructed by an optimization algorithm that minimizes the  $\ell_1$  norm of the data, obtaining a scene of higher resolution compared to the scene captured. Finally, a characterization of the Project Tango device is performed, Calculating the variation of its measurements when it is located a source of light near the device

---

(\*) Research Work

(\*\*) School of Physical-Mecanical Enginnering. Department of Electrical, Electronics and Telecommunications Engineering. Advisor: Henry Arguello. Co-Advisor: Samuel Pinilla

## Introducción

La reconstrucción tridimensional (3D) de una escena consiste en la obtención de las coordenadas espaciales de un objetivo de interés, es decir, dados dos puntos correspondientes al plano imagen se obtienen las coordenadas 3D del punto a un sistema de coordenadas del mundo Beltrán Guerrero and Basañez Villaluenga (2008). Este mapa tridimensional es una copia digital de una escena o objeto físico, la cual permite visualizar y manipular la escena. por medio de esta copia digital. Además los modelos tridimensionales tienen importantes aplicaciones en diversos campos como en medicina, para el análisis de escenas forenses Thali et al. (2003), detección de cáncer de mama Shen et al. (2013), en arqueología, para la documentación, estudio y reparación de piezas y sitios históricos Plets et al. (2012), De Reu et al. (2013). Además es usada en geología para crear modelos montañosos, ubicar fuente de recursos naturales y determinar fallas geológicas Zanchi et al. (2009), Yanlin et al. (2011), Cartwright and Huuse (2005). Además, es usado en áreas en crecimiento como en la realidad virtual, la cual pretende cambiar la manera en la que las personas interactúan con el entorno, logrando importantes aplicaciones en entretenimiento Lopes et al. (2015), educación Ott et al. (2015), medicina Soeiro et al. (2016) entre otros.

Diversos métodos se han desarrollado con el fin de recrear un entorno tridimensional, los más comunes para medir y reconstruir una escena tridimensional son visión estereoscópica, telemetría láser y escáner de luz estructurada Saaidi et al. (2014) Sarbolandi et al. (2015). Estos métodos mencionados están basados en el uso de sistemas ópticos. En particular, la visión estereoscópica es

un método que permite determinar la ubicación espacial de un objeto mediante la correspondencia de dos puntos en las imágenes, similar a como lo hace el ojo humano Ann et al. (2016). Por otro lado, la telemetría láser determina la distancia de un objeto midiendo el tiempo de recorrido de un pulso láser. Por último el escáner de luz estructurada adquiere los datos proyectando un haz de luz sobre el objeto y capturando su reflejo mediante un dispositivo de carga acoplada (CCD por sus siglas en inglés) Salvi et al. (2010).

Las técnicas tradicionales de adquisición de datos están regidas por la tasa de Nyquist, la cual establece que una señal puede ser reconstruida si la frecuencia de muestreo es al menos el doble del ancho de banda de la señal Donoho (2006), Davenport et al. (2011). La tasa de Nyquist se convierte entonces en una limitación en el desarrollo de sistemas de adquisición sofisticados, pues la cantidad de muestras adquiridas es tan alta que dificulta su almacenamiento y transmisión Vivek et al. (2015). Sin embargo, en el año 2006 una teoría llamada muestreo compresivo (CS, por sus siglas en inglés de Compressed Sensing), asegura que una señal puede ser reconstruida con un número mucho menor de muestras que los requeridos por la tasa de Nyquist Candès and Wakin (2008). Específicamente, el muestreo compresivo resuelve el problema matemático expresado como  $\mathbf{y} = \mathbf{H}\mathbf{x}$ , donde  $\mathbf{y} \in \mathbb{R}^m$  se conoce como las medidas,  $\mathbf{H} \in \mathbb{R}^{m \times n}$  como la matriz de muestreo y  $\mathbf{x} \in \mathbb{R}^n$  la señal de interés, asumiendo que  $m \ll n$ .

Es importante notar que dependiendo del método de adquisición, de las limitaciones del sistema, de los errores en la transmisión o del almacenamiento y manipulación de los datos, el número de muestras obtenidas por un sistema de adquisición puede ser escaso, incumpliendo la tasa de Nyquist y causando que los algoritmos de reconstrucción fallen en la estimación de la señal de interés, resaltando la importancia de CS. Por estas razones, la teoría de muestreo compresivo ha sido aplicada en diversos campos como imágenes por resonancia magnética Lustig et al. (2008), radares Herman and Strohmer (2009) y redes de comunicación Huang et al. (2013).

Para solucionar el problema de la escasez de datos considerando las limitaciones anteriores, se estiman datos nuevos a partir de un conjunto discreto de los datos adquiridos, esta técnica matemática es conocida como interpolación. Los métodos de interpolación más comunes son el de mínimos cuadrados y de error absoluto, llamados también norma  $\ell_2$  y norma  $\ell_1$  respectivamente Martínez and Bouza (2013). Los métodos tradicionales utilizan la norma  $\ell_2$  (norma euclidiana) como estimador de aquellos puntos no medidos. Por otro lado, la norma  $\ell_1$  minimiza el error absoluto, la cual es una métrica menos susceptible ante mediciones atípicas Bogdan et al. (2013). Además, cabe resaltar, que la norma  $\ell_1$  promueve soluciones escasas Tusell (2011), donde la escasez hace referencia a que la señal tiene un pequeño número de muestras diferentes de cero Candès and Romberg (2007). Es importante notar, que los métodos tradicionales mencionados, utilizan interpolación por mínimos cuadrados. Sin embargo las nuevas estimaciones producidas por estos algoritmos basados en nube de puntos no es la mejor cuando la señal capturada es escasa Moenning and Dodgson (2003), Rusu et al. (2009). Además, utilizar la norma  $\ell_1$  para la reconstrucción tridimensional evita el problema de suavizado, ya que la norma  $\ell_2$  penaliza fuertemente las variaciones atípicas, obligando a considerar superficies lisas Avron et al. (2010), contrario a la norma  $\ell_1$  que no se ve limitada por ello. Por tanto, este proyecto de grado desarrolla e implementa un algoritmo que minimiza la norma  $\ell_1$  para reconstruir una escena tridimensional de mayor resolución a partir de los datos escasos capturados con el dispositivo Project Tango. Para realizar esto, la nube de puntos es sincronizada con la imagen de color de la escena, y organizada un cubo de datos que representa el espacio tridimensional capturado. Los resultados obtenidos presentan un aumento en la resolución espacial de la imagen tridimensional con respecto a la capturada. Finalmente, en este trabajo se realiza una caracterización del dispositivo project Tango, buscando evaluar la capacidad del dispositivo para obtener una nube de puntos cuando las condiciones lumínicas no son las recomendadas por el fabricante.

## 1. Métodos de Escaneo tridimensional

En este capítulo se presentan los métodos tradicionales más usados para reconstruir una escena tridimensional, realizando una breve descripción de su funcionamiento.

Para realizar una reconstrucción 3D se debe generar una imagen de la superficie, una proyección bidimensional del objeto y posteriormente añadir la profundidad de cada punto de la superficie. Existen múltiples métodos que permiten calcular la profundidad, los cuales se clasifican en: de contacto o invasivos y ópticos o no invasivos de Jesús Flores García (2013). Los métodos de contacto son los más antiguos, funcionan desplazando un cabezal sobre una superficie que obtiene las coordenadas correspondientes. Este sistema requiere que la superficie a escanear sea lo suficientemente rígida para no sufrir daños al contacto con el cabezal. Este sistema no suele ser utilizado en el escaneo de superficies complejas debido a la dificultad para llegar a todos los puntos de la superficie Saíz (2010). Por otro lado, los métodos ópticos tienen la característica de no entrar en contacto directo con la superficie. Estos métodos se clasifican en activos (emiten radiación al medio) o pasivos (captura radiación directamente del medio), entre los métodos podemos nombrar la visión estereoscópica, la telemetría láser, el escáner de luz estructurada y el escáner de holografía conoscópica Lopez Mendoza et al. (2013).

### 1.1 Visión estereoscópica

El término visión estereoscópica es usado para describir el problema de obtener un punto espacial en una escena tridimensional, partiendo de las imágenes capturadas por dos o más cámaras desde

diferentes puntos de vista Chávez Morales and Flores Parra (2011). Los algoritmos de reconstrucción que usan esta técnica deben solucionar el problema de correspondencia de píxel. Esto es, dado un píxel  $x$  en la primera imagen, se debe encontrar el correspondiente píxel  $x'$  en la segunda imagen, para todos los píxeles de la imagen, como se observa en la Figura 1. El problema de correspondencia puede ser resuelto de varias maneras, usando diferencia absoluta (AD), suma de las diferencias absolutas (SAD), diferencias cuadráticas (SD) o suma cuadrática de las diferencias (SSD) Zhang (2013). Para una ventana fija en una imagen, el píxel  $p_l$  con coordenadas  $(x_l, y)$  e intensidad  $I_l(x_l, y)$  en la imagen izquierda y el píxel candidato  $p_r$  con desplazamiento  $s$  en la imagen de la derecha, su diferencia absoluta se define como

$$AD(x_l, y, s) = |I_l(x_l, y) - I_r(x_l - s, y)|. \quad (1)$$

La suma de las diferencias absolutas mide la sumatoria de las diferencias absolutas en una región de tamaño  $(2w + 1) \times (2w + 1)$ , centrado en el píxel  $p_l$  de la imagen izquierda y su región correspondiente en la imagen derecha, la cual es calculada como

$$SAD(x_l, y, s) = \sum_{i=-w}^w \sum_{j=-w}^w |I_l(x_l + i, y + j) - I_r(x_l + i - s, y + j)| \quad (2)$$

De manera similar la diferencias cuadráticas y la suma cuadrática de las diferencias se define como sigue

$$\begin{aligned} SD(x_l, y, s) &= (I_l(x_l, y) - I_r(x_l - s, y))^2, \\ SSD(x_l, y, s) &= \sum_{i=-w}^w \sum_{j=-w}^w (I_l(x_l + i, y + j) - I_r(x_l + i - s, y + j))^2, \end{aligned} \quad (3)$$

## 1.2 Telemetría láser

Este método consiste en emitir un pulso láser y medir el reflejo producido por la superficie, de manera que, conocido el tiempo transcurrido entre la emisión y la recepción, y siendo la velocidad

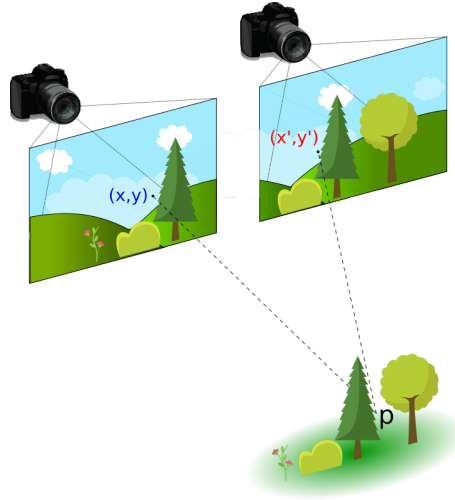


Figura 1: Visión Estereoscópica

de la luz aproximadamente constante, es posible calcular la distancia del objeto Kim et al. (2009). Además, esta técnica es conocida como "tiempo de vuelo". Sin embargo, la velocidad de la luz es tan alta que los componentes electrónicos fallan al calcular la diferencia de tiempo Orta Méndez (2015). Por esto, es común que el escáner mida el desfase de pulso láser recibido con respecto al emitido, produciendo así un plano de profundidad donde cada píxel codifica la distancia de un punto en el espacio. La Figura 2 ilustra el principio de funcionamiento del escáner. Se observa que el haz de luz emitido, tiene un desfase  $\Delta\phi$  con respecto al recibido.

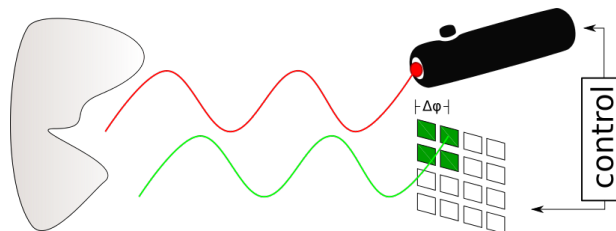


Figura 2: Telemetría Láser

Para detectar la diferencia de fase, se emite un haz de luz que puede ser generado como pulsos o como una señal continua modulada (CW, por sus siglas en inglés de Continuous wave). Las fuentes

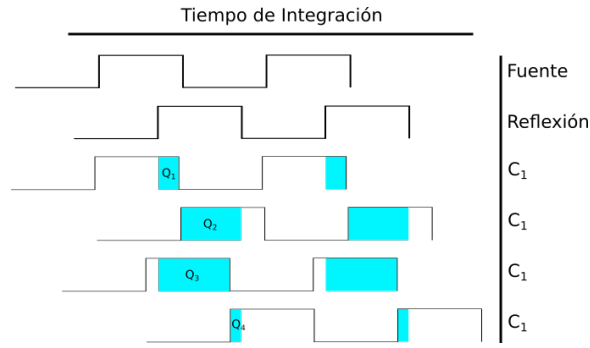


Figura 3: Representación de las cargas en el sensor

CW generalmente son de señales sinusoidales o cuadradas, esta última es más común por la facilidad de producirlas mediante circuitos digitales. Cuando se trabaja con señales continuas (CW), se toman múltiples muestras por medida, cada una de ellas separadas 90 grados, como se ilustra en la Figura 3. De esta manera el desfase entre el haz de luz emitido y el recibido puede calcularse como

$$\Delta \phi = \arctan \frac{Q_3 - Q_4}{Q_1 + Q_2} \quad (4)$$

Ademas, la distancia del objeto puede ser determinada por

$$d = \frac{c \Delta \phi}{4\pi f}, \quad (5)$$

donde  $c$  es la velocidad de la luz,  $f$  la frecuencia del haz emitido, y  $\Delta \phi$  el desfase.

### 1.3 Escáner de luz estructurada

El sistema de luz estructurada utiliza un patrón láser y mide la deformación que este sufre al iluminar un objeto Rodrigues et al. (2013). Esta técnica proyecta un patrón, el cual esta constituido por franjas claras y oscuras diseñadas de tal forma que a cada píxel se le asigna un código definido. La Figura 4 presenta un ejemplo de un patrón que cambia con cada captura aumentando la cantidad de franjas en cada toma. De esta forma es posible obtener un código binario que define la estructura del objeto. La Figura 5 presenta el montaje tradicional para la técnica del sistema de luz estructurada. La expresión que permite obtener la coordenada  $(x, y, z)$  asociadas a un píxel  $p$  en la cámara con

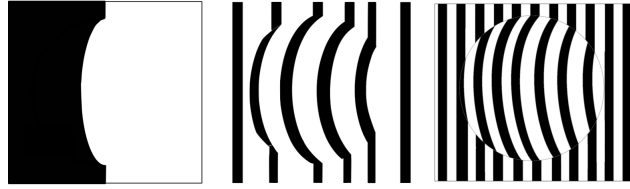


Figura 4: Ejemplo de patrones de luz estructurada

coordenadas  $(v, h)$  se define como

$$x = D_p - \frac{D_p D_s}{v P D_p + W_n} \tag{6}$$

$$y = \frac{h P D_p D_s}{v P D_p + W_n} \tag{7}$$

$$z = \frac{W_n D_s}{v P D_p + W_n} \tag{8}$$

donde  $D_p$  es la distancia del proyector al origen,  $D_s$  es la distancia de la cámara al proyector,  $v$  y  $h$  son las coordenadas horizontal y vertical del píxel en la cámara,  $P$  es el tamaño del objeto en el plano de la cámara, y  $W$  es el grosor entre franjas sucesivas del patrón de luz.

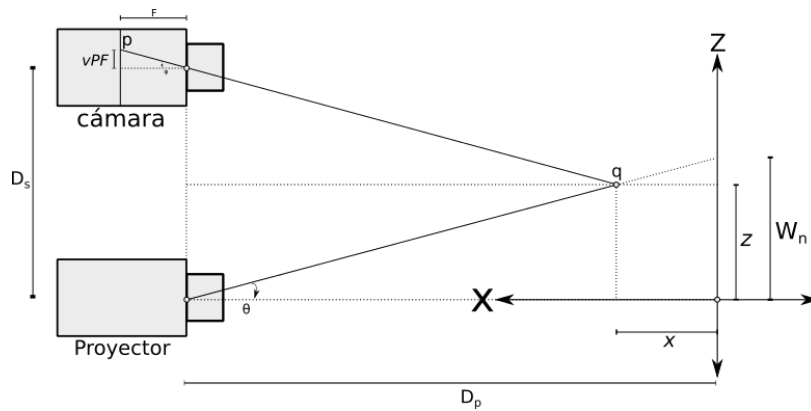


Figura 5: Montaje escáner de luz estructurada

## 2. Project Tango

"La tecnología de project Tango brinda al dispositivo móvil la capacidad de navegar el mundo físico de manera similar a como lo hacemos los humanos mediante la adición de equipo de visión avanzada, procesamiento de imágenes y sensores ópticos especiales" (Google Tango Developers, 2017).

Project Tango es un moderno dispositivo móvil con características de hardware y software únicas, que lo dotan con la capacidad de ubicarse espacial y temporalmente Keralia et al. (2014). El dispositivo realiza más de 250 mil mediciones por segundo, actualizando su posición y rotación constantemente, siendo capaz de medir, modelar el entorno que lo rodea Keralia et al. (2014). Tango ofrece a los dispositivos móviles este tipo de comprensión mediante el uso de tres tecnologías principales: Seguimiento de movimiento, Aprendizaje de área y percepción de profundidad. Esta última característica utilizada en el desarrollo de este proyecto.

### 2.1 Percepción de profundidad

Project Tango es un sistema de escaneo 3D conformado por diferentes sensores que permiten recrear entornos tridimensionales mediante algoritmos de procesamiento de imágenes y visión computarizada, incluidos en el núcleo del sistema operativo. Para este propósito, Tango utiliza un emisor y receptor conformado por un proyector láser infrarrojo (IR, por sus siglas en inglés) y un sensor CCD (cámara RGB-IR). Este sensor mide el desfase del haz de luz recibido con el haz emitido, generando

así, una nube de puntos que representa el entorno tridimensional. Además, el sensor realiza mediciones cada  $200ms$ , lo cual es una frecuencia de muestreo de  $5Hz$ . Por otro lado, Tango presenta dificultades al realizar mediciones sobre algunos objetos, los cuales por su forma geométrica o su material, no retornan el pulso infrarrojo, como son los objetos orgánicos o luminiscentes Google (2015).

Tango usa el sensor RGB-IR para crear una imagen apareada del entorno registrado en la nube de puntos, con la imagen RGB de la escena, creando un mapa interactivo del entorno, usando comúnmente en aplicaciones de realidad aumentada. Además Tango trabaja con diversos sensores para su ubicación espacial (GPS, giroscopio, compás, acelerómetro), que permite almacenar la ubicación espacial de cada punto de la nube con respecto a su posición en el mundo real.

## 2.2 Hardware

El dispositivo cuenta con un procesador Nvidia Tegra K1 cuatro núcleos a  $2.3\text{ Ghz}$ ,  $4\text{ Gb}$  de memoria ram Elpida,  $128\text{ Gb}$  de memoria flash SanDisk, conectividad wifi 802.11.n y bluetooth 4.0 a través del chip Broadcom. Además cuenta con un receptor GNSS de múltiples constelaciones Broadcom, el cual le permite al dispositivo mejorar su ubicación espacial cuando opera en interiores. En la Figura 6a podemos observar la ubicación de estos chips.

La parte trasera del dispositivo cuenta con 2 cámaras, una cámara ojo de pez, y una cámara de  $4MP$  RGB-IR, junto con el proyector infrarrojo ubicado en el lado opuesto, como se observa en la Figura 6b.

En la Figura 7, se muestra la parte interna de la Tango, entre ellos su tarjeta madre mostrada en la Figura 6a, las cámaras traseras, la batería de 2 celdas, su módulo de comunicación, antenas, disipadores de calor, pantalla táctil y carcasa.

La parte exterior del dispositivo luce como en la Figura 8, donde se puede observar la ubicación de los diferentes elementos que componen el dispositivo, entre ellos sus pines de carga, ranuras para

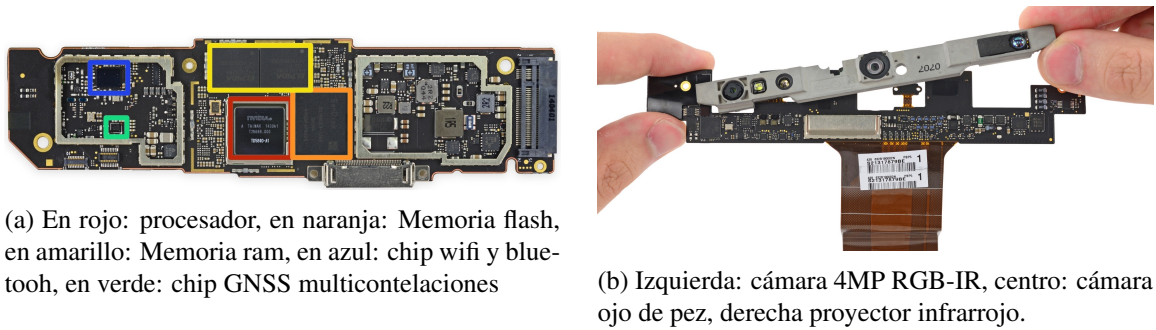


Figura 6: Vista interna de los componentes, Adaptado de Sam Lionheart, Walter Galan / Project Tango Tablet Teardown from <https://es.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148>

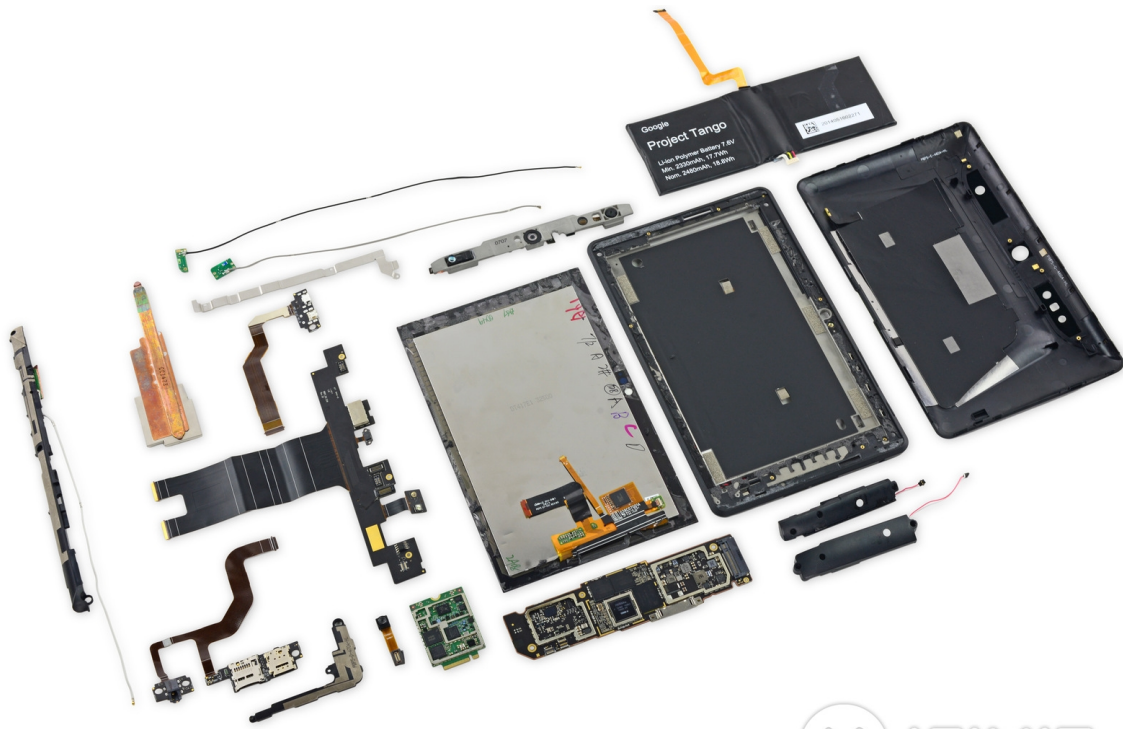


Figura 7: Vistazo al interior de la Tango, Adaptado de Sam Lionheart / Project Tango Tablet Teardown from <https://es.ifixit.com/Teardown/Project+Tango+Tablet+Teardown/28148>

micro SD y nano Sim, puerto de audio, botones de encendido y volumen, y las cámaras delantera y traseras. En la Tabla 1 se presenta un resumen del hardware con el que cuenta el dispositivo Project Tango.

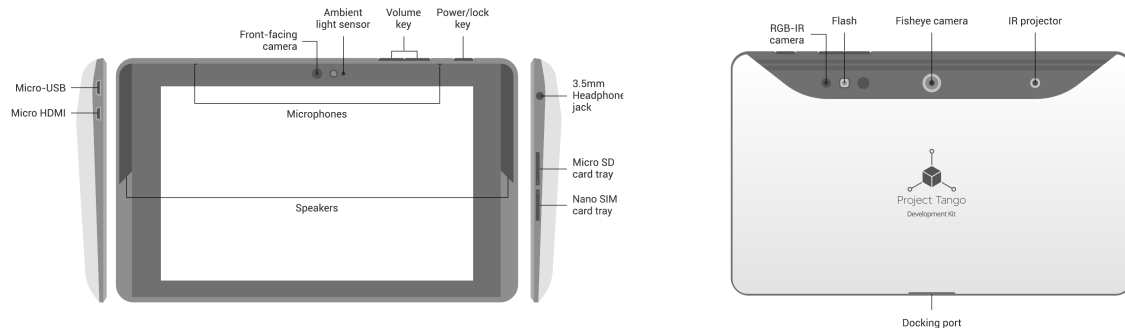


Figura 8: Exterior del dispositivo Tango

Tabla 1: Hardware Dispositivo Project Tango

Hardware	especificaciones
Pantalla	7.02" 1920 × 1200 HD IPS (323 ppi)
Dimensiones	119.77 × 196.33 × 15.36 mm
Peso	370 g
Cámaras	Sensor de 4MP RGB-IR, píxel $2\mu\text{m}$ Frontal de 1MP
Wireless	4G LTE Wi-Fi (2.4 Ghz / 5 Ghz), Wifi 802.11 1/b/g/n NFC
Memoria	128 Gb de almacenamiento interno 4 Gb Ram
Puertos	Micro HDMI USB 3.0 Micro SD Nano Sim
Batería	4960 mAH (celdas $2 \times 2480$ )
Procesador	NVIDIA Tegra K1 w/192 núcleos CUDA
Sensores	Cámara de seguimiento de movimiento Sensor de profundidad 3D Acelerómetro Sensor de luz ambiente Barómetro Compás GPS Giroscopio

### 3. Muestreo Compresivo

Las técnicas tradicionales de adquisición de datos están limitadas por la tasa de Nyquist, la cual establece que para poder reconstruir una señal la frecuencia de muestreo debe ser por lo menos el doble del ancho de banda de la señal. Con el desarrollo de sistemas de sensado cada vez más sofisticados, la tasa de Nyquist se convierte en un problema, debido a que la cantidad de muestras adquiridas es tan alta que complica su posterior almacenamiento y transmisión Li et al. (2013). La teoría del muestreo compresivo (CS por sus siglas en inglés de Compressive Sensing) permite reconstruir una señal a partir de un número menor de muestras que el exigido por el teorema de Nyquist-Shannon Eldar and Kutyniok (2012). Para hacer esto posible, CS se basa en dos principios fundamentales: la escasez que está relacionada con la señal de interés, y la incoherencia que está ligada al método de sensado utilizado.

La escasez indica que una imagen puede ser representada en una base apropiada con una pequeña cantidad de coeficientes Paredes and Arce (2011), Quezada et al. (2016). Entonces, Sea una imagen  $\mathcal{F} \in \mathbb{R}^{N^2 \times L}$  o  $\mathbf{f} \in \mathbb{R}^{N^2 \hat{A} \hat{u} L}$  en su forma vectorial (como se ilustra en la Figura 9),  $\mathbf{f}$  puede ser representada en una base ortonormal como

$$\mathbf{f} = \sum_{i=1}^n \theta_i \boldsymbol{\psi}_i, \quad (9)$$

donde  $\boldsymbol{\theta}$  es la secuencia de coeficientes escasos de  $\mathbf{f}$ , tal que  $\theta_i = \langle \mathbf{f}, \boldsymbol{\psi}_i \rangle$ . De acuerdo a esta notación se puede expresar  $\mathbf{f}$  como es producto matricial entre  $\boldsymbol{\Psi}$  y  $\boldsymbol{\theta}$ , donde  $\boldsymbol{\Psi}$  es una matriz de  $N^2 L \times N^2 L$ , donde  $n = N^2 L$ . Formalmente, se dice que  $\mathcal{F}$  es escasa en una base de representación  $\boldsymbol{\Psi}$ , si  $\mathbf{f} = \boldsymbol{\Psi} \boldsymbol{\theta}$  puede ser aproximada por una combinación lineal de  $s$  vectores de  $\boldsymbol{\Psi}$  con  $s \ll n$ .

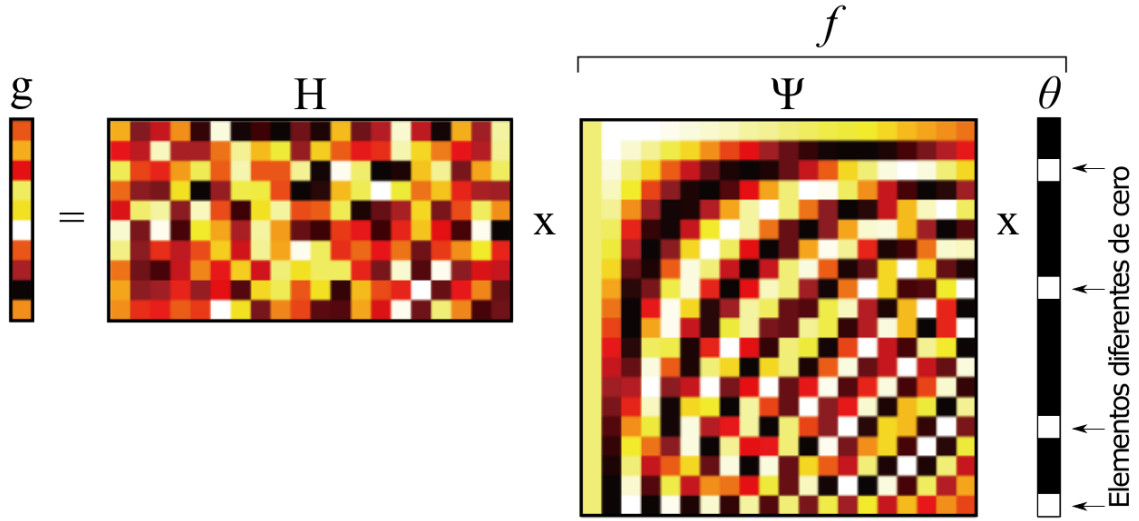


Figura 9: Muestreo compresivo de imágenes (CS). Aquí  $g$  es el vector de medidas compresivas,  $\mathcal{H}$  es la matriz del sistema,  $\Psi$  es la base de representación escasa y  $\theta$  es la secuencia o vector de coeficientes escasos.

CS permite que  $f$  sea recuperada a partir de  $m$  proyecciones aleatorias con alta probabilidad cuando  $m > s \log n \ll n$ . Así, la escasez indica que cuando una imagen tiene una expansión escasa, es posible descartar los coeficientes más pequeños sin perder gran cantidad de información de la imagen.

Por otro lado, la incoherencia hace referencia a la estructura de las formas de onda de muestreo utilizada en CS, que a diferencia de la imagen de interés, tiene una representación densa en la base  $\Psi$ . Dadas dos bases ortonormales  $\langle \mathcal{H}, \Psi \rangle$ , donde  $\mathcal{H} \in \mathbb{R}^{N(N+L-1) \times N^2 L}$  es una matriz que corresponde a la base de muestreo para  $f$ , y  $\Psi$  es la base de representación para  $f$  en el dominio escaso. La coherencia entre las bases ortonormales  $\mathcal{H}$  y  $\Psi$  se define como

$$\mu \langle \mathcal{H}, \Psi \rangle = \sqrt{N^2} \max_{1 \leq i, j \leq N^2} \langle \mathcal{H}_i, \Psi_j \rangle \quad (10)$$

De forma general, la coherencia mide la mayor correlación entre dos elementos cualesquiera de  $\mathcal{H}$  y  $\Psi$  Donoho and Huo (2001). Si  $\mathcal{H}$  y  $\Psi$  poseen elementos correlacionados, la coherencia es grande, de otra forma, es pequeña. Cabe resaltar que CS se centra en los pares con baja coherencia.

Con el fin de recuperar la imagen a partir de las mediciones compresivas  $\mathbf{g}$ , se debe resolver el problema inverso de CS minimizando la siguiente función costo

$$\tilde{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{g} - \mathcal{H}\Psi\theta\|_2^2 + \lambda \|\theta\|_1. \quad (11)$$

Los algoritmos de reconstrucción CS pueden clasificarse en cinco enfoques computacionales Tropp and Wright (2010). Los algoritmos de búsqueda codiciosa (algoritmos greedy) obtienen una estimación escasa de  $\theta$  de manera iterativa mediante la identificación de componentes que permitan obtener la mejor reconstrucción en cada iteración. Entre los algoritmos que implementan este enfoque se encuentran: OMP (Orthogonal Matching Pursuit) Tropp and Gilbert (2007), StOMP (OMP por etapas) Donoho et al. (2012), y CoSaMP (Compressive Sampling Matching Pursuit) Needell and Tropp (2009). El segundo enfoque obtiene una representación escasa de la imagen multispectral resolviendo el problema de optimización convexa mostrado en la Ecuación (11). Este enfoque es implementado por los métodos de gradiente descendente como el algoritmo de reconstrucción escaso a través de aproximación separable (SpaRSA) Wright et al. (2009), TwIST Bioucas-Dias and Figueiredo (2007) y GPSR Figueiredo et al. (2007). La tercera categoría de algoritmos se basa en un enfoque bayesiano, el cual asume una distribución a-priori de los coeficientes de la señal que se quiere reconstruir Ji et al. (2008). Estos algoritmos encuentran un estimador de un máximo a-posteriori que incorpora la observación Ji et al. (2008). El cuarto enfoque consiste en relajar el problema de la norma  $\ell_0$  a un problema no convexo e intentar identificar un punto estacionario Chartrand (2007). En la ultima categoría están los algoritmos de búsqueda exhaustiva, los cuales realizan una estimación de la representación escasa de  $\hat{\mathbf{y}}$  buscando todas las posibles soluciones Miller (2002). Una revisión de estos algoritmos y su análisis de complejidad computacional pueden encontrarse en Tropp and Wright (2010)

## 4. Captura y Procesamiento de los Datos

En este capítulo se presentará el procedimiento que propone este trabajo para la reconstrucción de una imagen tridimensional, iniciando en el proceso de captura de los datos desde la Tango, posteriormente se ilustra como se realizó la adaptación de la nube de puntos para su procesado, y finalmente se presenta el algoritmo para la reconstrucción de la imagen tridimensional.

### 4.1 Proceso de captura con Tango

Project Tango tiene la capacidad de medir su posición espacial, haciendo posible que cuando la Tango se desplace o rote, pueda conocerse la dirección y la magnitud de su desplazamiento. Haciendo uso de esta característica, combinada con su capacidad de medir profundidad, la cual es determinada usando un sensor infrarrojo (IR por sus siglas en inglés de InfraRed) como se mencionó en la sección 2.1, es posible crear una nube de puntos donde cada punto contiene una coordenada  $x, y, z$  relativa al dispositivo.

Con el fin de procesar los datos capturados, Tango hace uso de diversos servicios para informar a las aplicaciones de las mediciones realizadas en su entorno, los cuales están disponibles en tres interfaces de programación de aplicaciones (API por sus siglas en inglés de Application Programming Interface) diferentes, Unity, Java API y C API. Cada una de las APIs tiene características que la hacen especial para diferentes tipos de aplicaciones. Por ejemplo, Unity es ideal si se desea crear un juego de realidad aumentada debido al uso de un motor gráfico para su desarrollo, Java API es ideal para crear cualquier tipo de aplicación android, y C API es el indicado cuando se requiere un manejo más flexible de los servicios del dispositivo. En la Figura 10 se observa el nivel de cada API

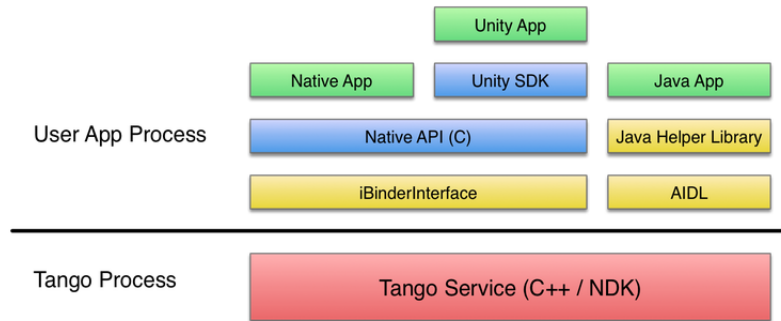


Figura 10: Tango APIs

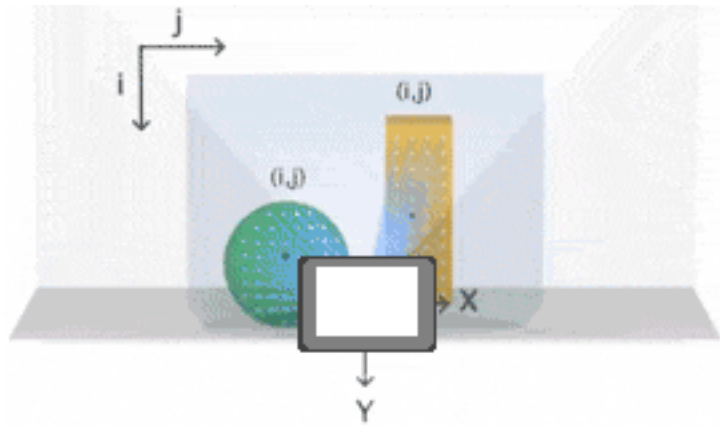


Figura 11: Percepción de profundidad del dispositivo Tango

en la pila de desarrollo. Para el desarrollo de este proyecto, se hizo uso de Java API y C API.

Por otro lado, cada punto de una nube de puntos capturada por Tango, tiene coordenadas  $x, y, z$  que esta determinada por el sistema de referencia, como se observa en la Figura 11. Esta nube de puntos está contenida en un vector de tipo coma flotante y cuya estructura esta dada por

$$[x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n]. \quad (12)$$

Además, es necesario proyectar la escena tridimensional en el plano bidimensional del dispositivo, con el fin de obtener el píxel correspondiente a cada punto de la nube de puntos para posteriormente asociarlo a la imagen RGB. La Figura 12 muestra como un punto de coordenadas  $x, y, z$  es proyectado sobre el plano imagen obteniendo una posición  $x', y'$ . La proyección de un punto sobre

el plano imagen se expresa como

$$\frac{x'}{x} = \frac{z'}{z} \quad y \quad \frac{y'}{y} = \frac{z'}{z}, \quad (13)$$

donde  $z'$  es la distancia focal. Además, de la Ecuación (13) es posible concluir

$$x' = f_x \times \frac{x}{z} \quad y \quad y' = f_y \times \frac{y}{z} \quad (14)$$

donde  $f_x$  y  $f_y$  son constantes que derivan de factores geométricos, como el ángulo de visión, la distancia focal, y la relación de aspecto en el plano imagen Feldmar et al. (1995). Note que si el ori-

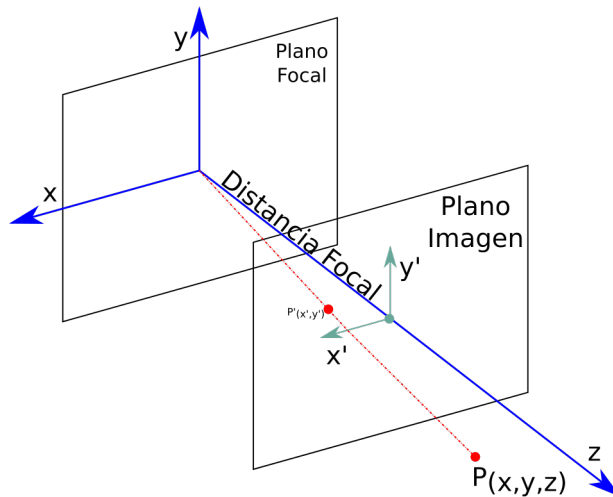


Figura 12: Proyección bidimensional de una punto tridimensional

gen del sistema de coordenadas de imagen no está en el centro del plano imagen, el desplazamiento  $c_x, c_y$  desde el origen hasta el centro del plano debe incluirse en las ecuaciones de proyección. Así, la Ecuación (15) puede reescribirse como

$$x' = f_x \cdot \frac{x}{z} + c_x, \quad y' = f_y \cdot \frac{y}{z} + c_y. \quad (15)$$

Finalmente, con la posición en píxeles de cada proyección de la nube de puntos, se logra crear una imagen bidimensional que es posteriormente asociada con la imagen RGB.

## 4.2 Sincronización del color y la profundidad

Para conseguir un modelo digital del entorno no solo basta con obtener una nube de puntos de su profundidad, también es necesario conseguir la imagen de color de la escena que representa. La Figura 13 muestra el procedimiento de sincronización de la nube de puntos y la imagen de color.

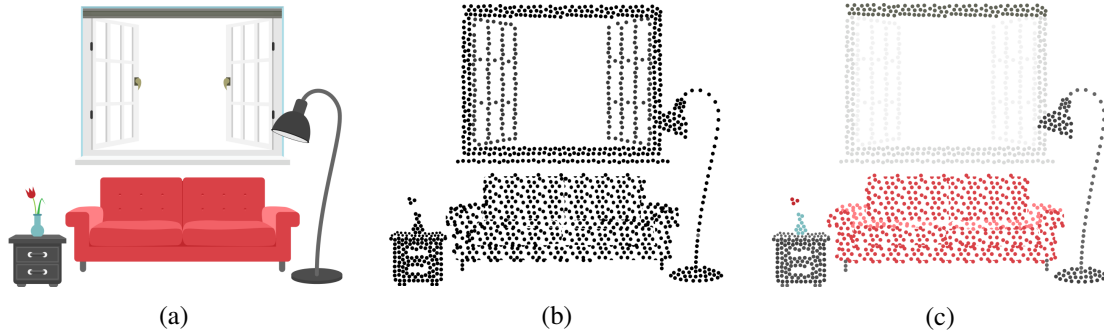


Figura 13: (a) muestra la imagen RGB de la escena, (b) representa la nube de puntos, y (c) la sincronización entre la imagen RGB y la nube de puntos

El dispositivo Tango realiza las capturas de color y profundidad en diferentes instantes de tiempo, por ello es necesario realizar una transformación de los datos para obtener las dos mediciones (color y profundidad), referenciadas al mismo instante de tiempo. Sin embargo, Tango realiza este proceso de manera automática. Por ejemplo, la Figura 13a muestra una imagen RGB de una escena tridimensional capturada por la Tango en un tiempo  $t_1$ , un instante de tiempo después, en  $t_2$ , el dispositivo captura la nube de puntos, la cual debe ser referenciada al tiempo  $t_1$ , como se ilustra en la Figura 13b. Finalmente, estas dos imágenes son unidas obteniendo una nube de puntos asociada a su imagen de color como en la Figura 13c. Por otro lado, los datos son almacenados en un vector cuya estructura esta dada por

$$[x_1, y_1, z_1, a_1, x_2, y_2, z_2, a_2, \dots, x_n, y_n, z_n, a_n] \quad (16)$$

donde  $x, y, z$  son las coordenadas obtenidas, y  $a$  es el código en hexadecimal del color asociado a ese punto. la Ecuación 17 presenta un ejemplo de un vector con 4 puntos

$$[1, 1, 3, FFFF00, 1, 2, 4, 002255, 2, 1, 2, 00FFFF, 2, 1, 4, 2A7FFF]. \quad (17)$$

### 4.3 Construcción del cubo de datos

Una vez se obtiene la nube de puntos, esta se organiza en un cubo que representa el espacio tridimensional de la escena. Además, el tamaño del cubo está determinado por la cantidad de valores únicos en cada coordenada. Por ejemplo la Figura 14 es la representación de un cubo de datos que contiene tres valores únicos en el eje  $x$ , dos valores únicos en el eje  $y$ , y 6 valores únicos en el eje  $z$ . Además, representar la nube de puntos de esta manera, permite manipular la imagen tridimensional, como si se tratara de un conjunto de imágenes bidimensionales, lo cual facilita su posterior reconstrucción.

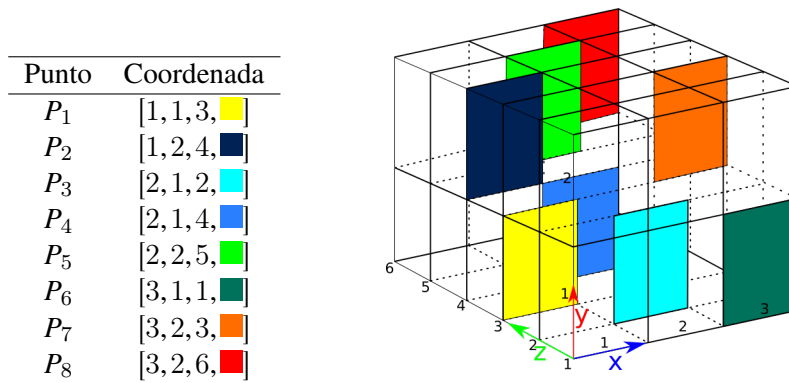
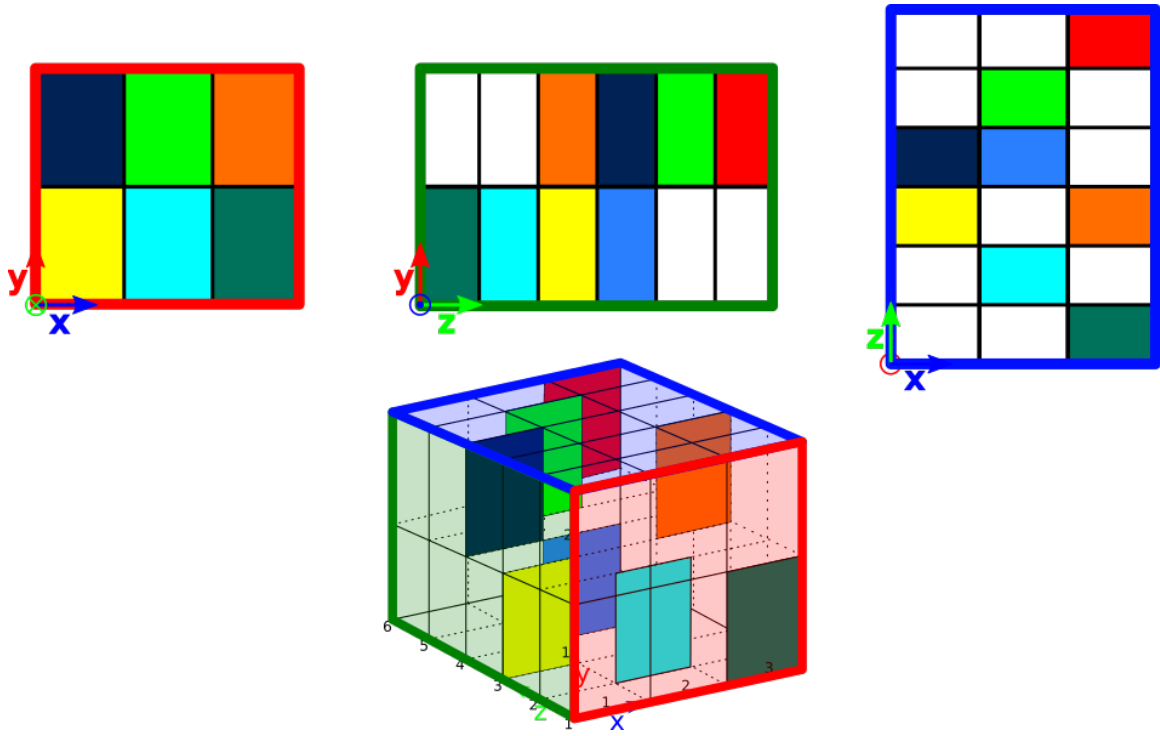


Figura 14: Cubo de datos

El procedimiento utilizado para crear el cubo de datos consistió en realizar proyecciones de la nube de puntos sobre los planos,  $A_{xy}$ ,  $A_{xz}$  y  $A_{zy}$ , y luego transformar estas 3 vistas en un cubo tridimensional. La Figura 15 muestra los planos asociados al ejemplo de la Figura 14.

Figura 15: Planos  $A_{xy}$ ,  $A_{xz}$  y  $A_{zy}$ 


---

**Algorithm 1** Cloud2Plane(*nube*, *eje<sub>1</sub>*, *eje<sub>2</sub>*)

**Require:** Matriz con la nube de puntos  $\mathbf{P} = [x_1, y_1, z_1, a_1; \dots; x_n, y_n, z_n, a_n]$ , Ejes de coordenadas  $c_1, c_2$

**Ensure:** Matriz  $\mathbf{A}$

```

1: Orden  $\leftarrow$  sortrows( $\mathbf{P}, c_1$ ) // ordenar de menor a mayor la matriz respecto a  $c_1$ 
2: cor  $\leftarrow$  Orden(:,  $c_1$ )
3: vcolor  $\leftarrow$  Orden(:, 4)
4: [unicoc1, posc1]  $\leftarrow$  unique(Orden(:,  $c_1$ ))
5: [unicoc2, posc2]  $\leftarrow$  unique(Orden(:,  $c_2$ ))
6:  $t_x \leftarrow$  size(unicoc1)
7:  $t_y \leftarrow$  size(unicoc2)
8:  $n \leftarrow$  size(cor)
9:  $\mathbf{A} \leftarrow$  zeros( $t_y, t_x$ )
10: for  $i = 1, n$  do
11:    $\mathbf{A}(\text{posc2}[i], \text{posc1}[i]) \leftarrow$  vcolor[ $i$ ]
12: end for
13: return  $\mathbf{A}$ 

```

---

Por otro lado, el Algoritmo 1 transforma la nube de puntos en los tres planos que contienen la proyección de los datos como se ilustró en la Figura 15. Note, que el Algoritmo 1 tiene como

entrada una matriz  $\mathbf{P} \in \mathbb{R}^{n \times 4}$ , donde  $n$  es la cantidad de puntos que componen la nube de puntos, la cual es procesada de la siguiente manera:

1. Se ordena la matriz  $\mathbf{P}$  de manera descendente respecto a una de las coordenadas (línea 1).
2. Se extrae el vector  $\mathbf{cor}$  con las coordenadas ordenadas y el color asociado a ellas  $\mathbf{vcolor}$  (líneas 2 y 3).
3. Se determina los valores no repetidos en los ejes  $c_1$  y  $c_2$  y sus posiciones en el vector (líneas 4 y 5).
4. Se crea una matriz  $\mathbf{A} \in \mathbb{R}^{t_y \times t_x}$ , donde  $t_y$  y  $t_x$  es la cantidad de valores no repetidos del vector ordenado (línea 9).
5. Finalmente la matriz  $\mathbf{A}$  es llenada con los colores asociados a la posición de los valores no repetidos (línea 11).

Además, note que el Algoritmo 1 solo tiene un ciclo de longitud  $n$  (línea 10), por tanto su complejidad temporal está dada por  $O(n)$ .

Una vez creadas las matrices que representan los 3 planos, se procede a crear el cubo de datos. Tomando como base cada valor de la matriz  $\mathbf{A}_{xy}$ , se realiza una búsqueda exhaustiva de dicho valor en los planos  $\mathbf{A}_{xz}$  y  $\mathbf{A}_{zy}$ . Una vez encontrada la similitud en las 3 matrices, se ubicará el valor hallado en el cubo de datos en la correspondiente posición. Por ejemplo, en la Figura 16, se realiza la búsqueda del vóxel azul en los planos  $\mathbf{A}_{xy}$ ,  $\mathbf{A}_{xz}$  y  $\mathbf{A}_{zy}$ , encontrándose en las posiciones  $x = 1$ ,  $y = 2$  y  $z = 4$ .

Además, el proceso de construcción del cubo de datos es realizado por el Algoritmo 2.

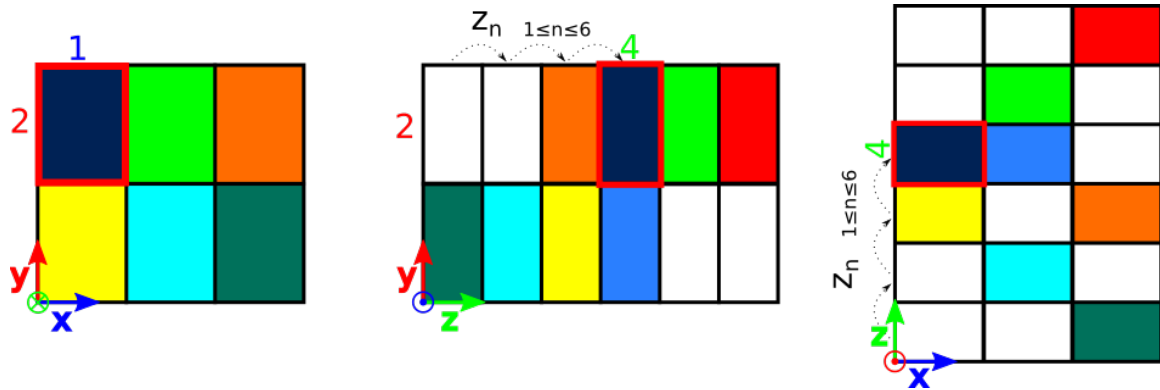


Figura 16: Búsqueda exhaustiva del algoritmo 2

**Algorithm 2** Plane2Cube(plano)**Require:** Matriz de los planos  $xy, xz, zy$ **Ensure:** Matriz 3D  $C$ 

```

1:  $[y, x, v] \leftarrow find(xy)$  //posición y valor de los elementos no nulos
2:  $n \leftarrow size(x)$ 
3: for  $i = 1, n$  do
4:    $color \leftarrow v(i)$ 
5:    $posx \leftarrow x(i)$ 
6:    $posy \leftarrow y(i)$ 
7:    $[z, v2] \leftarrow find(zy(posy, :))$ 
8:    $n2 \leftarrow size(z)$ 
9:   for  $j = 1, n2$  do
10:     $color2 \leftarrow v2(i)$ 
11:     $posz \leftarrow z(i)$ 
12:    if  $color == color2$  then
13:       $C(posy, posx, posz) \leftarrow color$  //asignación del color en la matriz 3D
14:    end if
15:  end for
16: end for
17: return  $C$ 

```

El Algoritmo 2, tiene como entrada 3 matrices de los planos  $A_{xy}$ ,  $A_{xz}$  y  $A_{zy}$ , las cuales son usadas para crear el cubo de datos. Una vez ingresadas las matrices mencionadas, se procede de la siguiente manera:

1. Se crea el vector  $v_1$  con los valores diferentes de cero de la matriz  $A_{xy}$  (línea 1).
2. para cada valor de  $v_1$  se hace

- (a) Se obtiene el color, y su posición (fila y columna) del punto en la matriz  $\mathbf{A}_{xy}$  (lineas 4, 5 y 6).
- (b) Se crea un vector  $\mathbf{v}_2$  con los valores diferentes de cero de la matriz  $\mathbf{A}_{zy}$  solo para la fila en la cual se ubica el punto (linea 7).
- (c) Para cada valor de  $\mathbf{v}_2$  se hace
  - i. obtiene su color y su posición en la matriz  $\mathbf{A}_{zy}$  (linea 10 y 11).
  - ii. se compara con el color de la matriz  $\mathbf{A}_{xy}$  (linea 12).
  - iii. Si son iguales se asigna el color en la respectiva posición en el cubo  $\mathbf{C}$ , de lo contrario se continua la búsqueda (linea 13).

Note que el Algoritmo 2 usa dos ciclos anidados de tamaño  $n$  para realizar la búsqueda (lineas 3 y 9), comparar y crear el cubo 3D. Así se concluye que su complejidad computacional es  $O(n^2)$ .

Como se presentó en el Capítulo 2, una captura típica realizada con el dispositivo Tango, contiene más de quince mil puntos, los cuales generan un cubo de grandes dimensiones, cuyo tamaño en memoria es de decenas de Gigabytes. Es evidente que estas matrices son difíciles de operar en un computador común. Para solucionar este inconveniente se hace necesario dividir el problema. Crear pequeños cubos de datos, operarlos y posteriormente liberar el espacio en memoria para un nuevo cubo. Este hecho hace que el algoritmo de reconstrucción se deba aplicar tantas veces como cubos de datos se generen. Es importante además, que las coordenadas  $x,y,z$  asociadas a cada vóxel del cubo de datos se actualicen cada vez que se crea un nuevo cubo, con la finalidad que una vez se reconstruya la imagen tridimensional, los nuevos puntos generados sean asociados a una coordenada física consistente con el vóxel que los generó. El Algoritmo 3 realiza este proceso de fragmentación del cubo de datos capturado, extrayendo las coordenadas y asociando el cubo reconstruido a ellas.

---

**Algorithm 3** setCoordinates( $\mathbf{RGB}_{\text{Original}}, \text{DatosMedidos}, \mathbf{RGB}, \text{delta}$ )

---

**Require:** Matriz RGB del cubo de datos  $\mathbf{RGB}_{\text{original}} = [R_1, G_1, B_1; \dots; R_n, G_n, B_n]$ , Matriz con la nube de puntos  $\mathbf{P} = [x_1, y_1, z_1, a_1; \dots; x_n, y_n, z_n, a_n]$ , Matriz RGB de los puntos reconstruidos  $\mathbf{RGB} = [R_1, G_1, B_1; R_n, G_n, B_n]$ , desplazamiento de las nuevas coordenadas  $\text{delta}$

**Ensure:** MatrizReconstruccion

```

1:  $\text{senal} \leftarrow \text{sparse}(\mathbf{RGB}_{\text{Original}}(:))$ 
2:  $\text{NumeroVoxels} \leftarrow \text{size}(\text{find}(\text{senal}, 1))$  //Cantidad de voxels diferentes de cero en el cubo
3: for  $i \leftarrow 1, \text{NumeroVoxels}$  do
4:    $\text{posicion} \leftarrow \text{find}(\text{DatosMedidos}(:, 4 : 6), \mathbf{RGB}_{\text{Original}}(i, :))$ 
5:    $\text{coordenadasXYZ} \leftarrow \text{DatosMedidos}(\text{posicion}, 1 : 3)$ 
6:    $\text{MatrizReconstruccion} \leftarrow [\text{coordenadasXYZ} + \text{delta}, \mathbf{RGB}]$ 
7: end for
8: return MatrizReconstruccion

```

---

El Algoritmo 3 recibe como parámetros de entrada la matriz  $\mathbf{RGB}_{\text{original}}$ , la nube de puntos, la matriz  $\mathbf{RGB}_{\text{reconstruida}}$  y la distancia de los nuevos puntos, los cuales son procesados como sigue:

1. Se vectoriza la matriz  $\mathbf{RGB}_{\text{original}}$  (línea 1).
2. Se determina la cantidad de vóxeles diferentes de cero (línea 2).
3. Para cada vóxel distinto de cero
  - (a) Búsqueda del vóxel en la nube de puntos (línea 4).
  - (b) Extracción de las coordenadas correspondientes al vóxel (línea 5).
  - (c) Asignación nuevas coordenadas a los puntos reconstruidos (línea 6).

Por otro lado, dado que el algoritmo 3 tiene un ciclo de tamaño  $n$ , su complejidad computacional es  $O(n)$

#### 4.4 Matriz de sensado y decimación

Como se mencionó en el Capítulo 3 para reconstruir una imagen a partir de un conjunto de mediciones escasas debemos resolver el problema inverso de CS expresado en la Ecuación 11, dando

solucion a la expresion  $\mathbf{g} = \mathbf{H}\mathbf{f}$ , donde  $\mathbf{f} \in \mathbb{R}^n$  y  $\mathbf{g} \in \mathbb{R}^m$  son la forma vectorizada de la matriz  $\mathbf{F}$  y  $\mathbf{G}$  respectivamente, y  $\mathbf{H}$  es la matriz de sensado.

Asumiendo que el cubo de datos representa una versión de baja resolución de la imagen a reconstruir, el proceso de sensado puede ser reescrito como  $\mathbf{g} = \mathbf{H}\mathbf{D}\mathbf{z}$ , donde  $\mathbf{D}$  es una matriz de decimación y  $\mathbf{z} \in \mathbb{R}^p$  con  $p > n$ , es una versión de alta resolución de  $\mathbf{f}$ .

La matriz de decimación  $\mathbf{D} \in \{0, 1\}^{\frac{NML}{\alpha^2} \times NML}$ , donde  $\alpha$  es el factor de decimación y  $N, M, L$  son las dimensiones del cubo de datos. Además la estructura de  $\mathbf{D}$  esta dada por

$$\mathbf{D} = \begin{bmatrix} \text{sec}(\alpha) & & & & 0_{1 \times NML-3\alpha} \\ & 0_{1 \times \alpha} & \text{sec}(\alpha) & & 0_{1 \times NML-4\alpha} \\ & \vdots & & \ddots & \vdots \\ & & & & 0_{1 \times NML-4\alpha} & \text{sec}(\alpha) & 0_{1 \times \alpha} \\ & & & & 0_{1 \times NML-3\alpha} & & \text{sec}(\alpha) \end{bmatrix} \quad (18)$$

donde  $\text{sec}(\alpha)$  es un vector cuya secuencia de 1 y 0 esta dado el factor de decimación  $\alpha$ . El vector  $\text{sec}(\alpha)$  se define como

$$\text{sec}(\alpha) = [1_1 \ 1_2 \ \dots \ 1_\alpha \ 0_1 \ 0_2 \ \dots \ 0_\alpha \ 1_1 \ 1_2 \ \dots \ 1_\alpha] \quad (19)$$

Además,  $0_{1 \times NML-3\alpha}$  y  $0_{1 \times \alpha}$  son vectores nulos de longitud  $NML - 3\alpha$  y  $\alpha$  respectivamente.

La Figura 17 ilustra la estructura de la matriz  $\mathbf{D}$ , con  $N = 4$ ,  $M = 4$ ,  $L = 4$  y  $\alpha = 2$ . Note que cada



Figura 17: Estructura de la matriz  $\mathbf{D}$  con  $N = 4$ ,  $M = 4$ ,  $L = 4$  y  $\alpha = 2$ . Donde los puntos blancos representan 1, y los negros 0

fila de la matriz  $\mathbf{D}$  tiene  $2\alpha$  elementos distintos de cero, esto es debido a la asunción de que cada punto de la imagen de baja resolución es el resultado de la suma de las intensidades de sus puntos vecinos. Obtener una imagen de mayor resolución a partir de una de baja resolución es llamado superresolución. Además, este proceso de decimación puede seguir escalándose, obteniendo cada vez una imagen de mayor resolución, como se ilustra en la Figura 18.

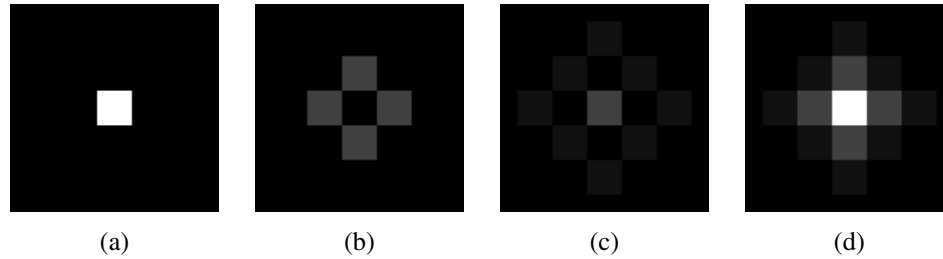


Figura 18: Imagen intencionalmente mostrada con el fin de ilustrar el proceso de superresolución. a) Punto capturado, b) Primera Reconstrucción, c) Segunda Reconstrucción, d) Unión del punto capturado y sus reconstrucciones

La matriz de sensado  $\mathbf{H} \in \{0, 1\}^{\beta \times \frac{NML}{\alpha^2}}$ , representa el proceso de captura del dispositivo, donde  $\beta$  es el número de valores diferentes de cero del cubo de datos. Su estructura es única para cada cubo debido a que depende de la cantidad y la posición de los valores capturados. La Figura 19a presenta un ejemplo de la construcción de  $\mathbf{H}$ , con un cubo de datos de dimensiones  $M = 4$ ,  $N = 4$  y  $L = 4$ , con factor de decimación  $\alpha = 2$  y con 7 voxels diferentes de cero, y la matriz de sensado  $\mathbf{H} \in \{0, 1\}^{7 \times 16}$  presentada en la Figura 19b.

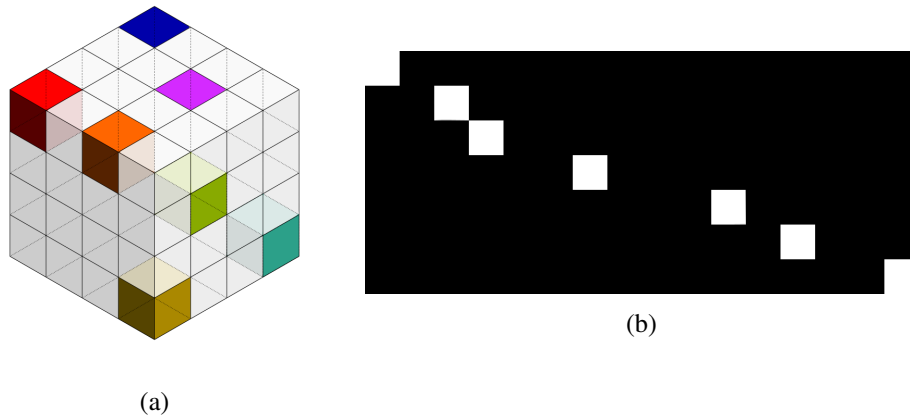


Figura 19: (a) Cubo de datos, (b) matriz de sentido  $\mathbf{H}$  asociada al cubo de datos.

---

**Algorithm 4** *getSensingMatrix(cubo,alpha)*

---

**Require:** cubo de datos  $\mathbf{C}$ , factor decimación  $\alpha$

**Ensure:** Matriz de sentido

```

1:  $\mathbf{x} \leftarrow \text{sparse}(\mathbf{C}(:))$  vectorización del cubo de datos
2:  $l \leftarrow \text{nthroot}(\text{size}(\mathbf{x}), 3)$  tamaño de los lados del cubo
3: for  $i = 1 : 2l : \text{size}(\mathbf{x})$  do
4:    $pos_1 = i/2$ 
5:   for  $j = 1 : 2 : l - 1$  do
6:     for  $k = 1, \alpha^2$  do
7:        $p[k] \leftarrow \mathbf{x}(k)$ 
8:       if  $p[k] > 0$  then
9:          $n \leftarrow n + 1$ 
10:         $pos \leftarrow (j + pos_1)/2$ 
11:         $H(n, pos) \leftarrow 1$  asignación en la matriz de sentido
12:       end if
13:      $j \leftarrow j + 2$ 
14:   end for
15: end for
16:    $i \leftarrow \alpha \times l$ 
17: end for
18: return  $\mathbf{H}$ 

```

---

Por otro lado, el Algoritmo 4 realiza el proceso de construcción de la matriz  $\mathbf{H}$ . Además, este procesa el cubo de datos de entrada  $\mathbf{C}$  y el factor de decimación  $\alpha$ , de la siguiente manera:

1. Se vectoriza el cubo de datos  $\mathbf{C}$  (línea 1).
2. Se obtiene el tamaño  $l$  de cada lado del cubo (línea 2).

3. Variando la posición en el cubo en secciones de tamaño  $\alpha^2$  hacer (lineas 3, 5 y 6).
  - (a) Verificar si algún punto de la sección es mayor a cero. (linea 8)
  - (b) Si es cierto se obtiene su posición y se ubica en la matriz  $\mathbf{H}$  (linea 9, 10 y 11).

El algoritmo 4 utiliza tres ciclos anidados de tamaño  $n$ , así se puede concluir que su complejidad temporal es  $O(n^3)$ , siendo el método con mayor complejidad computacional.

Finalmente, considere la matriz  $\Phi \in \{0,1\}^{\beta \times NML}$ , que es construida como  $\Phi = \mathbf{H}\mathbf{D}$ , donde  $\beta$  es la cantidad de voxels diferentes de cero, y  $NML$  las dimensiones del cubo de datos. Por tanto, el proceso de decimación y captura de los datos usando el dispositivo Tango puede modelarse como

$$\mathbf{g} = \mathbf{H}\mathbf{D}\mathbf{f} = \Phi\mathbf{f}. \quad (20)$$

La Figura 20, presenta un ejemplo de la matriz  $\Phi \in \{0,1\}^{7 \times 64}$ , usando  $\mathbf{H}$  de la Figura 19b y  $\mathbf{D}$  de la Figura 17. Así, el problema inverso planteado en la Ecuación (20), asumiendo que la imagen deseada  $\mathbf{f}$  es escasa, se puede resolver por el siguiente problema de optimización

$$\mathbf{f} = \underset{\mathbf{z} \in \mathbb{R}^{N^3}}{\operatorname{argmin}} \|\mathbf{g} - \Phi\mathbf{z}\|_2^2 + u \|\mathbf{z}\|_1 \quad (21)$$

donde el término  $\|\mathbf{g} - \Phi\mathbf{z}\|_2^2$  promueve la fidelidad de los datos y el término  $u \|\mathbf{z}\|_1$  promueve la esparsidad de la solución.

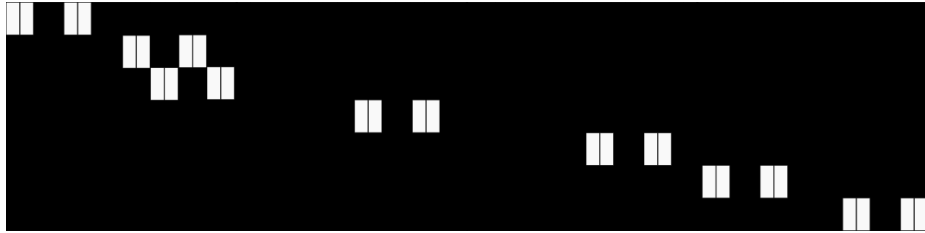


Figura 20: matriz  $\Phi$  que representa el proceso de captura de datos y decimación, donde los puntos blancos representan 1, y los negros 0.

## 5. Resultados de la Reconstrucción tridimensional

Con base en los algoritmos presentados en el capítulo anterior, este capítulo plantea un algoritmo final que reconstruya la escena tridimensional. La sección 5.1 presenta el algoritmo final condensado, la sección 5.2 muestra los resultados obtenidos minimizando la norma  $\ell_1$  y usando MC. Además, la sección 5.3, presenta los resultados obtenidos usando una técnica de interpolación tradicional (norma euclídea).

### 5.1 Algoritmo de reconstrucción

El propósito de esta sección es describir el algoritmo de reconstrucción, Algoritmo 5, que realiza el proceso de captura modelado en (20) y resuelve el problema de optimización en (21).

**Algorithm 5** Algoritmo Reconstrucción 3D**Require:** nube de puntos  $\mathbf{P}$ , Numero Iteraciones  $d$ , Tamaño cubo de datos  $t$ **Ensure:** Matriz Reconstrucción  $\mathbf{M}$ 


---

```

1: for  $i = 1$  hasta  $d$  do
2:    $\text{OrdenX}, \text{OrdenY}, \text{OrdenZ} \leftarrow \text{sortrows}(\mathbf{P}, 1), \text{sortrows}(\mathbf{P}, 2), \text{sortrows}(\mathbf{P}, 3)$ 
3:    $\text{lx}, \text{ly}, \text{lz} \leftarrow \text{unique}(\text{OrdenX}(:, 1)), \text{unique}(\text{OrdenY}(:, 2)), \text{unique}(\text{OrdenZ}(:, 3))$ 
4:    $\text{alfa}, \text{delta} \leftarrow 0.001, 2$ 
5:    $\text{PlanoXY}_{\text{rojo}}, \text{PlanoZY}_{\text{rojo}}, \text{planoXZ}_{\text{rojo}} \leftarrow \text{Cloud2Plane}(P, 1, 2), \text{Cloud2Plane}(P, 3, 2), \text{Cloud2Plane}(P, 1, 3)$ 
6:   //repetir para verde y azul
7:   for  $z=1:\text{lz}$  do
8:     for  $y=1:\text{ly}$  do
9:       for  $x=1:\text{lx}$  do
10:         $\text{seccionxy} \leftarrow \text{PlanoXY}(y : t \times y, x : t \times x)$ 
11:         $\text{seccionzy} \leftarrow \text{PlanoZY}(y : t \times y, z : t \times z)$ 
12:         $\text{seccionxz} \leftarrow \text{PlanoXZ}(z : t \times z, x : t \times x)$ 
13:         $\text{Cubo}_{\text{rojo}} \leftarrow \text{Plane2Cube}(\text{seccionxy}, \text{seccionzy}, \text{seccionxz})$ 
14:        //repetir para verde y azul
15:         $\mathbf{H} \leftarrow \text{getSensingMatrix}(\text{Cubo}_{\text{rojo}}, \text{alfa})$ 
16:         $\mathbf{D} \leftarrow \text{Down1}(t, \text{alfa})$ 
17:         $\Phi \leftarrow \mathbf{H}\mathbf{D}$ 
18:         $\text{Imagen}_{\text{rojo}} \leftarrow \text{GPSR}(\text{Cubo}_{\text{rojo}}, \Phi)$ 
19:        //repetir para verde y azul
20:         $[\text{vro}], [\text{vgo}], [\text{vbo}] \leftarrow \text{find}(\text{Cubo}_{\text{rojo}}), \text{find}(\text{Cubo}_{\text{verde}}), \text{find}(\text{Cubo}_{\text{azul}})$ 
21:         $\text{RGB}_{\text{original}} \leftarrow [\text{vro}, \text{vgo}, \text{vbo}]$ 
22:         $[\text{vr}], [\text{vg}], [\text{vb}] \leftarrow \text{find}(\text{Imagen}_{\text{rojo}}), \text{find}(\text{Imagen}_{\text{verde}}), \text{find}(\text{Imagen}_{\text{azul}})$ 
23:         $\text{RGB}_{\text{rec}} \leftarrow [\text{vr}, \text{vg}, \text{vb}]$ 
24:         $\text{Ptemp} \leftarrow \text{setCoordinates}(\text{RGB}_{\text{original}}, \text{OrdenX}, \text{RGB}_{\text{rec}}, \text{delta})$ 
25:       end for
26:     end for
27:   end for
28:    $\mathbf{P} \leftarrow \text{Ptemp}$ 
29: end for

```

---

El Algoritmo 5 usa como entrada la nube de puntos  $\mathbf{P}$ , el número de reconstrucciones  $d$  que realizara el algoritmo, y el tamaño del cubo de datos  $t$  de la siguiente manera:

- Se ordena la nube de puntos  $\mathbf{P}$  respecto a cada coordenada (línea 2).
- Se determinan los valores únicos de cada coordenada (línea 3).
- Se crean las matrices de los tres planos para cada color ( $R, G, B$ ) (línea 5).
- Se toman secciones de los planos de tamaño  $t$  (línea 10, 11 y 12).

- Se crean los cubos de datos, sobre los cuales se realizara la reconstrucción. (línea 13).
- Construcción de la matriz de decimación y sentido (línea 15 y 16).
- Reconstrucción del cubo usando el algoritmo GPSR para cada color (línea 18)
- Almacenamiento la reconstrucción. (línea 26).
- Se repite el procedimiento hasta cubrir la totalidad de la nube de puntos.

## 5.2 Simulaciones

La evaluación del desempeño del algoritmo fue realizada sobre Matlab 2015a, en un AMD A10 7300 a 1.9 GHz, 4 núcleos y 8 Gb de memoria RAM, usando Linux Mint como sistema operativo. Las medidas tomadas por el dispositivo Tango, junto con la matriz de sentido  $\mathbf{H}$ , y la matriz de decimación  $\mathbf{D}$ , fueron usadas como parámetros de entrada al algoritmo de reconstrucción GPSR. En las siguientes secciones se presenta el algoritmo de reconstrucción y los resultados obtenidos.

**5.2.1 Reconstrucción basado en la minimización Norma  $\ell_1$ .** La Figura 21 presenta las imágenes capturadas por el dispositivo Tango. Además se utilizó el algoritmo de reconstrucción presentado en la sección 5.1, para obtener imágenes de mayor resolución a las capturadas.

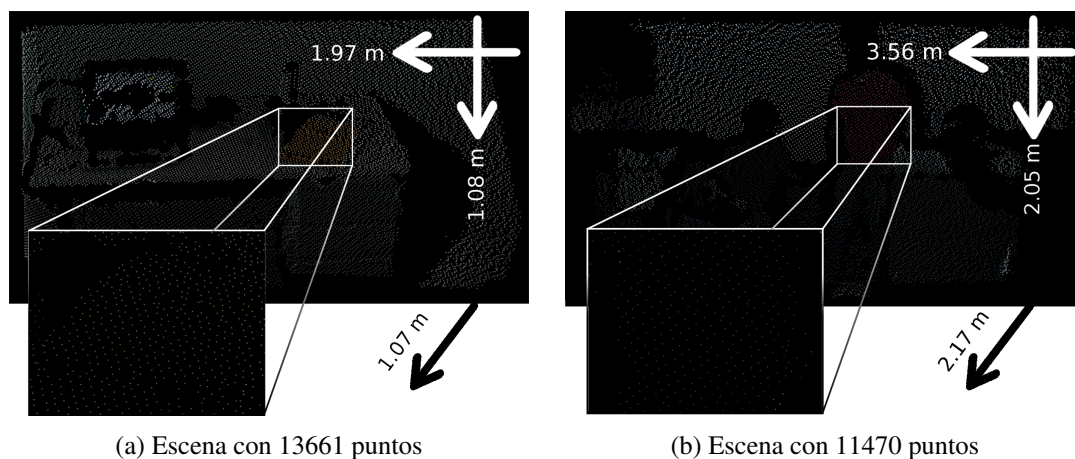


Figura 21: Escenas capturadas por el dispositivo Tango

Los Resultados obtenidos presentados a continuación varían en la cantidad de iteraciones realizadas por el algoritmo para reconstruir la escena, este proceso es similar al ilustrado en la Figura 18. La Figura 22 y Figura 23 presentan 4 imágenes de mayor resolución obtenidas con el Algoritmo 5. Las reconstrucciones muestran una escena tridimensional de mayor resolución, manteniendo de manera consistente las dimensiones originales de la escena.

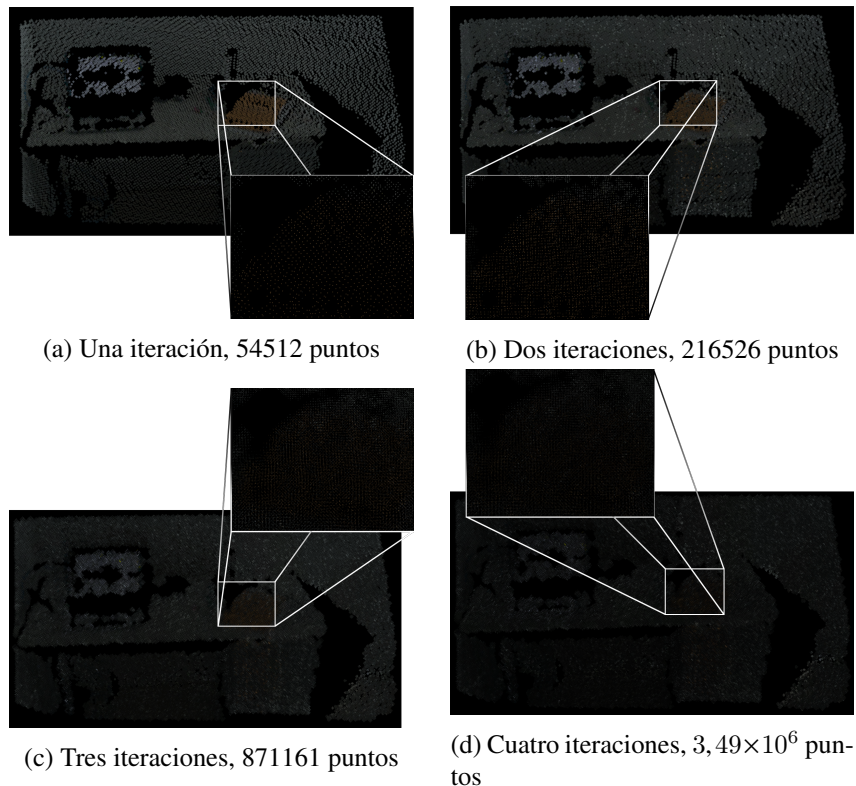


Figura 22: Reconstrucción de la escena de la Figura 21a

**5.2.2 Matrix Completion.** Considere nuevamente el proceso de obtener las matrices de proyección  $A_{xy}$ ,  $A_{xz}$  y  $A_{yz}$ , ilustrado en la Figura 15. Note que la cantidad de elementos conocidos de dichas matrices son escasos, entonces podemos considerar que  $A_{xy}$ ,  $A_{xz}$  y  $A_{yz}$  son de bajo rango y escasas. Recientemente, se a propuesto un método basado en CS, para completar dichos elementos desconocidos, llamado Matrix Completion (MC), el cual permite completar una matriz con solo un pequeño numero de muestras de la misma Keshavan et al. (2010).

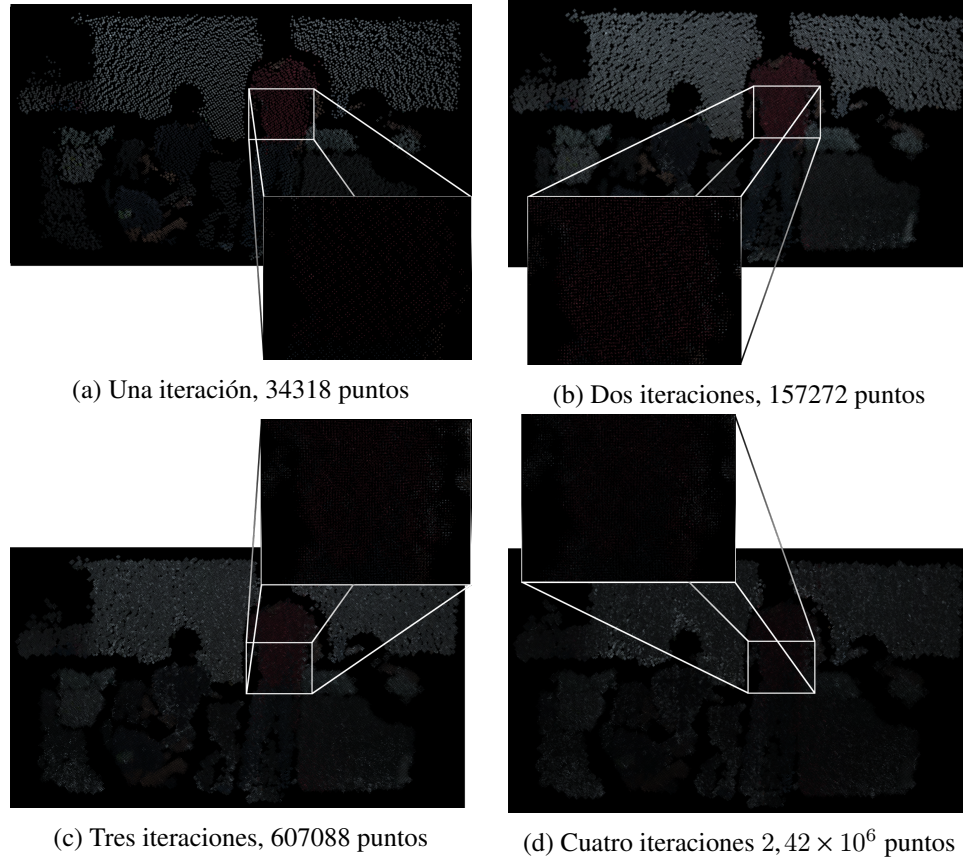


Figura 23: Reconstrucción de la escena de la Figura 21b

Sea una matriz de datos  $\mathbf{A} \in \mathbb{R}^{m \times n}$  generada por las observaciones de una matriz de bajo rango  $\mathbf{D} \in \mathbb{R}^{m \times n}$ . Entonces, el problema de completar una matriz de bajo rango y escasa como  $\mathbf{A}$  puede ser resuelto mediante el siguiente problema de optimización

$$\mathbf{D} = \underset{\mathbf{A} \in \mathbb{R}^{m \times n}}{\operatorname{argmin}} \|\mathbf{A}\|_* \quad (22)$$

donde  $\|\mathbf{A}\|_* = \sum_{k=1}^n \sigma_k \mathbf{A}$  es llamada la norma nuclear Candes and Recht (2012). Considerando que el problema de reconstrucción planteado pretende un aumento en la resolución de la escena, MC se plantea como una posible solución. Para ello, se hizo uso del algoritmo TFOCS Becker et al. (2011). Con el fin de usar MC para la reconstrucción de una imagen tridimensional de mayor resolución a la capturada, el Algoritmo 5 fue modificado. La modificación consistió en reemplazar el algoritmo

GPSR por TFOCS en la línea 18. Además, las matrices  $\mathbf{D}$  y  $\mathbf{H}$  no son necesarias, por este motivo, son eliminadas las líneas 15, 16 y 17.

Los resultados obtenidos en la reconstrucción de la escena usando MC son presentados a continuación. La Figura 24 muestra la reconstrucción para las escenas de la Figura 21, presentando un aumento en la resolución cercana al 300%, manteniendo inalterables las dimensiones de la escena.

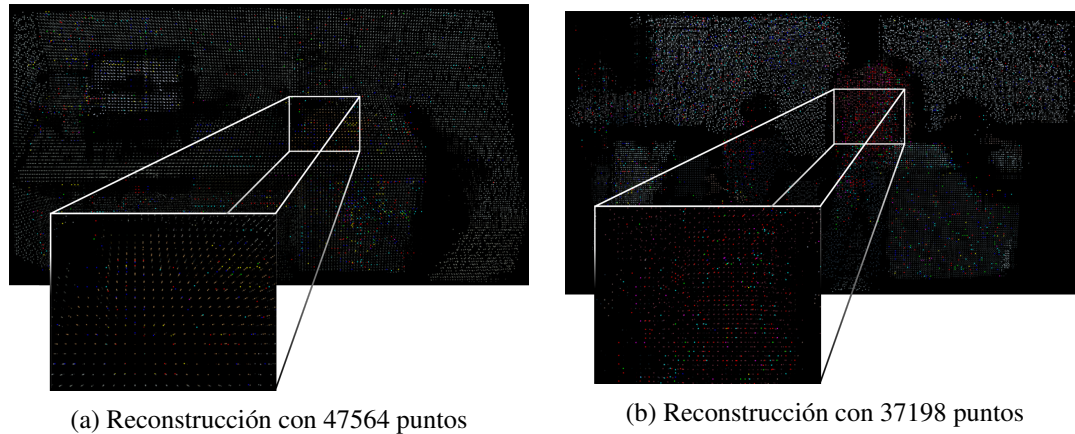


Figura 24: Reconstrucción de las escenas usando Matrix Completion

Los tiempos de calculo junto al aumento en la resolución de las imágenes, para los dos algoritmos son presentados en la Tabla 2. Es posible apreciar que el tiempo de cómputo cuando se minimiza la norma  $\ell_1$  de los datos, crece de manera exponencial con la cantidad de iteraciones que se realicen, haciendo que sea inviable incluso para un número pequeño de iteraciones. Sin embargo, el aumento en la resolución de la reconstrucción para una y dos iteraciones fue de aproximadamente de 4 y 16 veces comparada con la escena original, en un tiempo considerablemente bajo. Comparando estos resultados con los obtenidos usando MC, podemos observar que MC tardo cerca de 11 veces mas para obtener un aumento similar en la resolución de la escena.

### 5.3 Método tradicional de reconstrucción. Norma $\ell_2$

La norma euclídea o norma  $\ell_2$ , es el método mas común de interpolación, la cual es utilizada en casi todos los campos de la ingeniería. La norma  $\ell_2$  se define como

Tabla 2: Tabla comparativa para 4 iteraciones

## Personas Original 11470

Iteraciones	Cantidad de puntos	Tiempo
Una	34318	407.044 s
Dos	157272	2044.736 s
Tres	607088	>3 [h]
Cuatro	2420424	>8 [h]
Matrix Completion	37198	4735.299 [s]

## Escritorio Original 13661 puntos

Iteraciones	Cantidad de puntos	Tiempo
Una	54512	484.852 [s]
Dos	216526	2447.268 [s]
Tres	871161	>4 [h]
Cuatro	34992754	>10 [h]
Matrix Completion	47564	5255.062 [s]

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}, \quad (23)$$

Si la norma euclídea es utilizada para calcular una diferencia vectorial, se le llama distancia euclidiana, definida como

$$\|x_1 - x_2\|_2 = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2}, \quad (24)$$

o en su forma cuadrada

$$\|x_1 - x_2\|_2^2 = \sum_{i=1}^n (x_{1i} - x_{2i})^2. \quad (25)$$

Para resolver el problema de optimización planteado en la Ecuación (21), minimizando la norma  $\ell_2$ , solo es necesario eliminar el término  $u \|z\|_1$ , haciendo  $u = 0$ . Los resultados obtenidos minimizando la norma  $\ell_2$  presentan la misma cantidad de puntos que usando la norma  $\ell_1$ . Sin embargo, sus intensidades son menores, presentado una escena opaca y difícilmente visible, como se observa en la Figura 25.

En resumen, el algoritmo presentado reconstruye una escena tridimensional mostrando un aumento en la resolución espacial de la escena comparada con la capturada por el dispositivo Tango.

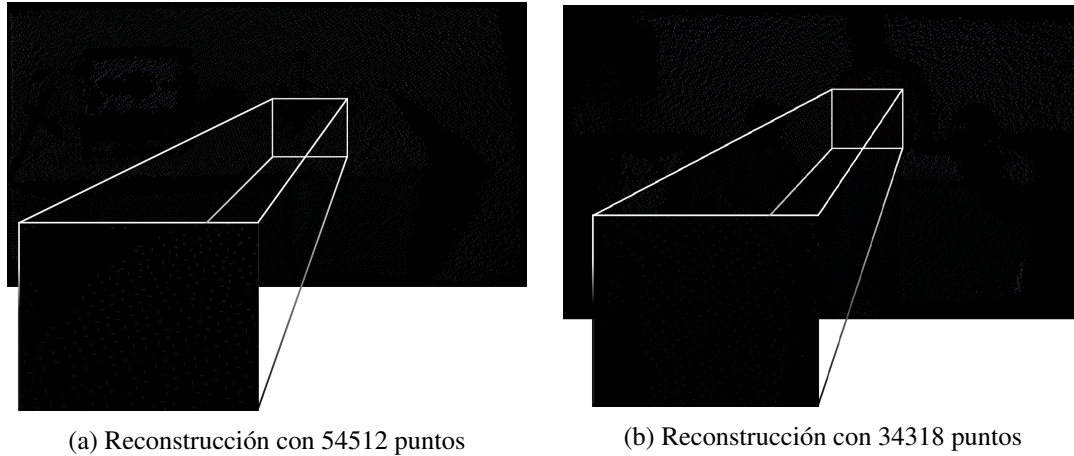


Figura 25: Reconstrucción de las escenas usando norma euclídea

Los resultados obtenidos minimizando la norma  $\ell_1$  muestran que la reconstrucción puede ser escalada obteniendo cada vez una escena de mayor resolución, significando esto un incremento en el tiempo de reconstrucción, el cual crece de manera exponencial con el número de iteraciones como lo se muestra en la Tabla 2. Por otro lado, se presentó Matrix Completion como un método alternativo para solucionar el problema, obteniendo de esta manera una reconstrucción con una cantidad de puntos similar a los obtenidos minimizando la norma  $\ell_1$ . Los resultados en ambos casos presentan un aumento en la resolución de la imagen. Además se mostraron los resultados obtenidos usando la norma  $\ell_2$ , presentando una menor calidad en la reconstrucción en comparación con los métodos propuestos.

Es importante en el proceso de captura de los datos tener en cuenta las condiciones en las cuales el dispositivo funciona de manera óptima, el siguiente capítulo describe las pruebas de iluminación realizadas con el fin de caracterizar el dispositivo y lograr determinar las condiciones en las que el dispositivo falla.

## 6. Caracterización del Dispositivo Tango

La caracterización de la Tango buscó evaluar la capacidad del dispositivo para obtener una nube de puntos, cuando las condiciones de iluminación son adversas a las indicadas por el fabricante para un correcto funcionamiento Google (2015). Dado que el dispositivo usa un conjunto emisor-receptor infrarrojo para crear un mapa de su entorno, la prueba consistió en iluminar la escena con una fuente de luz y observar la precisión de las mediciones realizadas por el dispositivo. La prueba fue realizada en el laboratorio de óptica del grupo de investigación HDSP de la Universidad Industrial de Santander.

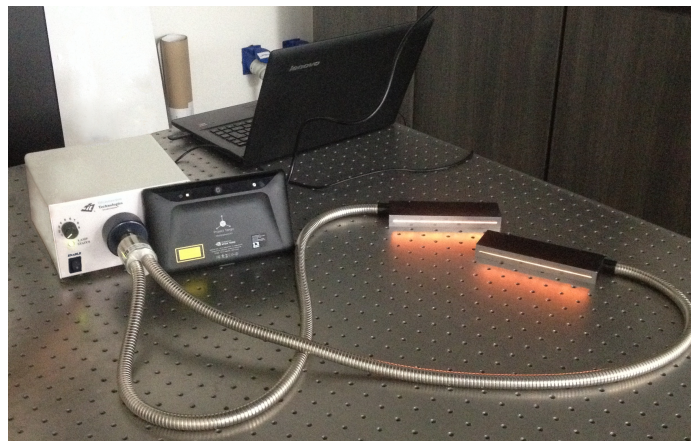


Figura 26: Sistema de iluminación utilizado durante la prueba

La Figura 26 presenta los elementos utilizados en el proceso de caracterización. Dichos elementos son una fuente de luz, un par de lámparas incandescentes y el dispositivo Tango. La fuente de luz

utilizada es una 3900e-DJ. Además las lámparas utilizadas son lightline 9140, las cuales permiten simular la iluminación natural dentro de nuestra escena.

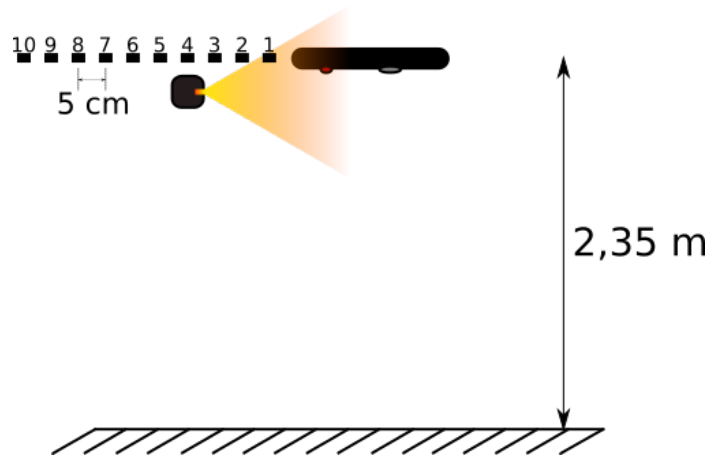


Figura 27: Prueba uno, fuente de luz ubicada a uno de los lados del dispositivo

La prueba de caracterización se dividió en tres etapas. La primera consistió en ubicar la fuente de luz en uno de los lados del dispositivo, variando la distancia de 5 *cm* a 50 *cm*, en pasos de 5 *cm*, y la intensidad de la luz emitida por ella, como se ilustra en la Figura 27. Para la segunda prueba, la fuente de luz se ubicó frente al dispositivo, variando de igual manera la distancia desde 5 *cm* a 50 *cm*, en pasos de 5 *cm*, y la intensidad, como lo ilustra la Figura 28. La tercera prueba se realizó ubicando la fuente sobre el dispositivo, y variando su intensidad para dos ubicaciones diferentes 0 *cm* y 50 *cm*, como se observa en la Figura 29.

La captura utilizada como referencia fue realizada en completa oscuridad, para garantizar una baja interferencia en las mediciones hechas por el dispositivo y utilizando como objetivo para la medición de las distancias, una pared del laboratorio, con el fin de maximizar la cantidad de puntos medidos. Los puntos capturados en esta prueba fueron 15971. Esta cantidad es utilizada como referencia para las demás pruebas.

La Figura 30 y Figura 31 representan la variación en la cantidad de puntos capturados por el dispositivo a medida que la fuente de luz se aleja de él, los resultados son presentados para cinco

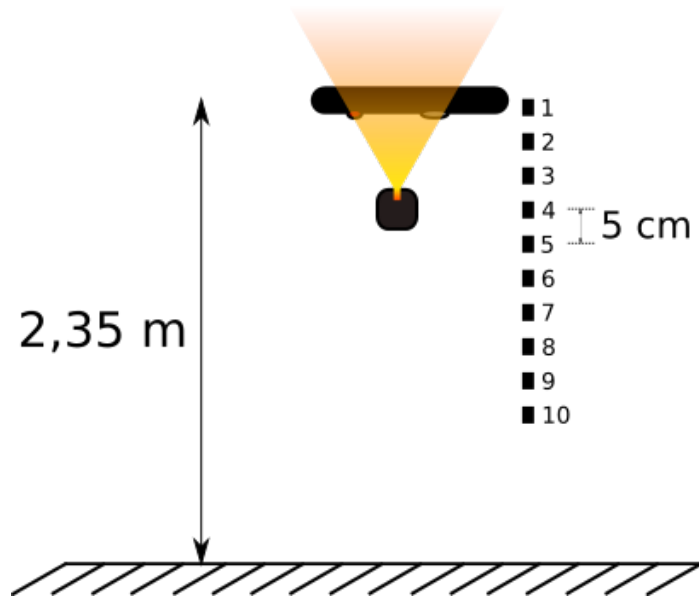


Figura 28: prueba dos, fuente de luz ubicada frente al dispositivo

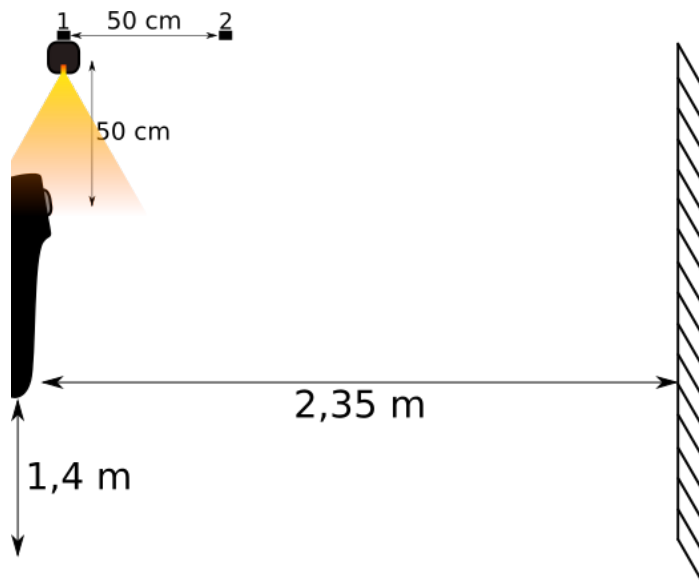


Figura 29: prueba tres, fuente de luz ubicada sobre el dispositivo

valores distintos de intensidad, siendo uno la menor y cuya ubicación es la mostrada en la Figura 27 y Figura 28 respectivamente. Note que en la Figura 30 las variaciones en la cantidad de puntos capturados por el dispositivo fue pequeña en cualquiera de los casos, y siempre por debajo de la referencia, este resultado es esperado, debido a que la fuente de luz está ubicada perpendicularmente

al sensor infrarrojo. En contraste, la Figura 31 muestra una variación significativa en el numero de

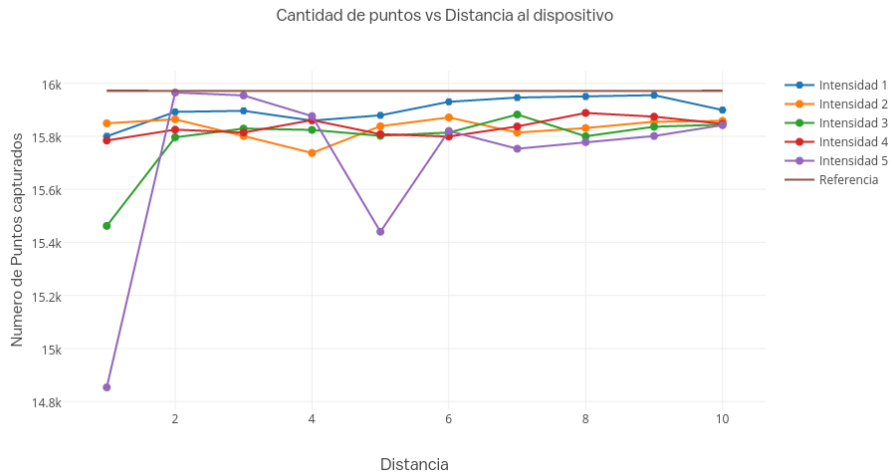


Figura 30: Comparación de los puntos capturados por el dispositivo para la prueba uno

puntos cuando la fuente esta paralela al sensor infrarrojo, disminuyendo entre 26.11% y 58.39% en el peor caso (circulo rojo). Además, podemos observar que exceptuando la prueba con la máxima intensidad, cuando la fuente de luz estaba a 30cm de la Tango, las variaciones en las mediciones fueron inferiores a 4%. (circulo celeste).

Por último, la Figura 32 presenta los resultados obtenidos cuando la fuente de luz esta ubicada como se ilustra en la Figura 29. Es posible apreciar que los resultados obtenidos en esta prueba

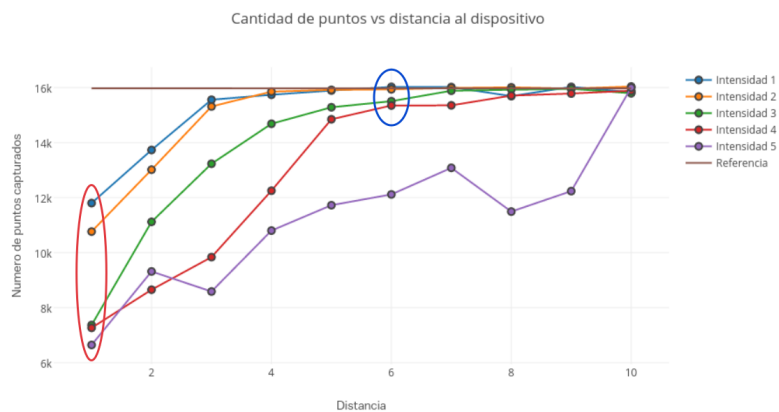


Figura 31: Comparación de los puntos capturados por el dispositivo para la prueba dos

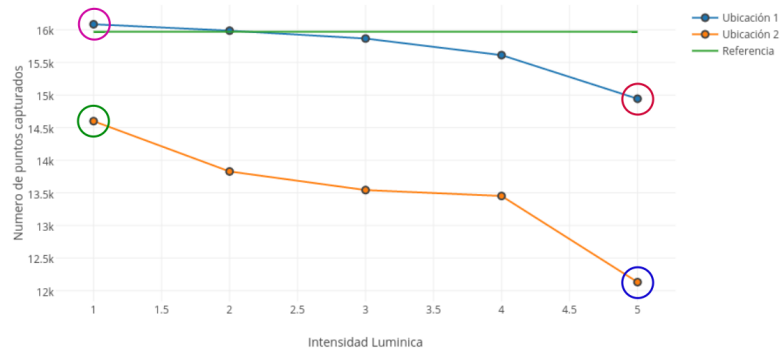


Figura 32: Comparación de los puntos capturados por el dispositivo para la prueba tres

concuerdan con los resultados de las pruebas uno y dos, ya que en la posición uno, obtuvo una disminución de puntos entre 0 y 7% (círculos rosa y rojo respectivamente). En contraste, en la prueba dos, la disminución de puntos estuvo entre 8.58 y 24.04% (círculos verde y azul respectivamente), debido a que en esta ubicación la luz procedente de la fuente llegaban al sensor con una menor inclinación, afectando negativamente el funcionamiento del mismo. En resumen, el dispositivo Tango presenta un bajo rendimiento, en términos de precisión, cuando es iluminado de manera directa. Sin embargo, los resultados obtenidos en la prueba dos (Figura 28) sugieren que si la fuente de luz está suficientemente alejada del dispositivo, las variaciones presentadas en las mediciones son bajas. Además, la prueba uno (Figura 27) indica que si la fuente de luz está ubicada de manera perpendicular a la Tango, sus mediciones presentan una disminución menor al 4% respecto a la referencia.

## 7. Conclusiones

Se presentó un algoritmo para la reconstrucción tridimensional de una imagen capturada por el dispositivo Tango, minimizando la norma  $\ell_1$ . Los resultados obtenidos minimizando la norma  $\ell_1$  muestran que la reconstrucción puede ser escalada obteniendo cada vez una escena de mayor resolución. Las resoluciones de las imágenes tridimensionales reconstruidas fueron de hasta 2561 veces en comparación con la capturada por el dispositivo Tango. Además, se presentó Matrix Completion como un método alternativo para reconstruir imágenes tridimensionales de mayor resolución a la capturada. Además las resoluciones obtenidas fueron similares a las conseguidas minimizando la norma  $\ell_1$  para una iteración. En contraste se realizó la reconstrucción minimizando la norma  $\ell_2$ . Los resultados obtenidos de esta manera muestran una escena opaca y difícilmente visible. Se realizó además la caracterización del dispositivo project Tango, mostrando la manera en la cual una fuente de luz puede afectar su correcto funcionamiento. Las pruebas de laboratorio sugieren que si la fuente de luz esta ubicada perpendicularmente a la Tango la disminución en la nube de puntos capturada es inferior al 4%. Por otra parte, si esta se encuentra paralela, la disminución máxima presentada fue de casi un 60%.

## Referencias Bibliográficas

- Ann, N. Q., Achmad, M. H., Bayuaji, L., Daud, M. R., and Pebrianti, D. (2016). Study on 3d scene reconstruction in robot navigation using stereo vision. In *Automatic Control and Intelligent Systems (I2CACIS), IEEE International Conference on*, pages 72–77. IEEE.
- Avron, H., Sharf, A., Greif, C., and Cohen-Or, D. (2010).  $\ell_1$ -sparse reconstruction of sharp point set surfaces. *ACM Transactions on Graphics (TOG)*, 29(5):135.
- Becker, S. R., Candès, E. J., and Grant, M. C. (2011). Templates for convex cone problems with applications to sparse signal recovery. *Mathematical programming computation*, 3(3):165–218.
- Beltrán Guerrero, D. and Basañez Villaluenga, L. (2008). Técnicas y algoritmos para la adquisición, transmisión y visualización de escenas 3d.
- Bioucas-Dias, J. M. and Figueiredo, M. A. (2007). A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2992–3004.
- Bogdan, M., Berg, E. v. d., Su, W., and Candes, E. (2013). Statistical estimation and testing via the sorted  $\ell_1$  norm. *arXiv preprint arXiv:1310.1969*.
- Candes, E. and Recht, B. (2012). Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119.
- Candes, E. and Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969.
- Candès, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30.
- Cartwright, J. and Huuse, M. (2005). 3d seismic technology: the geological  $\text{hubble}$ . *Basin Research*, 17(1):1–20.
- Chartrand, R. (2007). Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710.
- Chávez Morales, G. E. and Flores Parra, D. A. (2011). Diseño e implementación de un sistema de visión estereoscópica para el posicionamiento del brazo robótico crs-a255.
- Davenport, M. A., Duarte, M. F., Eldar, Y. C., and Kutyniok, G. (2011). Introduction to compressed sensing. *Preprint*, 93(1):2.
- de Jesús Flores García, A. (2013). Escaneo 3d en un campo de visión de 360 utilizando la técnica de proyección de franjas y un arreglo de espejos. Master's thesis, Centro de investigaciones en optica, AC.

- De Reu, J., Plets, G., Verhoeven, G., De Smedt, P., Bats, M., Cherretté, B., De Maeyer, W., Deconynck, J., Herremans, D., Laloo, P., et al. (2013). Towards a three-dimensional cost-effective registration of the archaeological heritage. *Journal of Archaeological Science*, 40(2):1108–1121.
- Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306.
- Donoho, D. L. and Huo, X. (2001). Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862.
- Donoho, D. L., Tsaig, Y., Drori, I., and Starck, J.-L. (2012). Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 58(2):1094–1121.
- Eldar, Y. C. and Kutyniok, G. (2012). *Compressed sensing: theory and applications*. Cambridge University Press.
- Feldmar, J., Ayache, N., and Betting, F. (1995). 3d-2d projective registration of free-form curves and surfaces. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 549–556. IEEE.
- Figueiredo, M. A., Nowak, R. D., and Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of selected topics in signal processing*, 1(4):586–597.
- Google (2015). *Google Tango Developers*. <https://developers.google.com/tango/overview/depth-perception>.
- Herman, M. A. and Strohmer, T. (2009). High-resolution radar via compressed sensing. *IEEE transactions on signal processing*, 57(6):2275–2284.
- Huang, H., Misra, S., Tang, W., Barani, H., and Al-Azzawi, H. (2013). Applications of compressed sensing in communications networks. *arXiv preprint arXiv:1305.3002*.
- Ji, S., Xue, Y., and Carin, L. (2008). Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356.
- Keralia, D., Vyas, K., and Deulkar, K. (2014). Google project tango-a convenient 3d modeling device. *International Journal of Current Engineering and Technology*, 4(5):3139–3142.
- Keshavan, R. H., Montanari, A., and Oh, S. (2010). Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998.
- Kim, Y. M., Theobalt, C., Diebel, J., Kosecka, J., Miscusik, B., and Thrun, S. (2009). Multi-view image and tof sensor fusion for dense 3d reconstruction. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1542–1549. IEEE.
- Li, S., Da Xu, L., and Wang, X. (2013). Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *IEEE Transactions on Industrial Informatics*, 9(4):2177–2186.
- Lopes, P., Ion, A., and Kovacs, R. (2015). Using your own muscles: realistic physical experiences in vr. *XRDS: Crossroads, The ACM Magazine for Students*, 22(1):30–35.

- Lopez Mendoza, P., Martinez Rivera, I., Latorre Astudillo, O., Rojas Mondaca, O., and Maluenda Bolados, P. (2013). Uso de escaner laser 3d en la reproduccion de restos oseos: una evaluacion desde la zooarqueologia, paleontologia y la conservacion-restauracion= use of 3d laser scanner in the reconstruction of bones remains: assessments from zooarchaeology, paleontology and conservation-restoration. *Conserva: Revista del Centro Nacional de Conservacion y Restauracion*, (18):59–70.
- Lustig, M., Donoho, D. L., Santos, J. M., and Pauly, J. M. (2008). Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72–82.
- Martínez, L. C. and Bouza, C. N. (2013). La norma  $\ell_1$  como alternativa a la norma  $\ell_2$  en el ajuste de la regresión.
- Miller, A. (2002). *Subset selection in regression*. CRC Press.
- Moening, C. and Dodgson, N. A. (2003). A new point cloud simplification algorithm. In *Proc. Int. Conf. on Visualization, Imaging and Image Processing*, pages 1027–1033.
- Needell, D. and Tropp, J. A. (2009). Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321.
- Orta Méndez, L. A. (2015). Telemetría láser.
- Ott, M., FREINA, L., et al. (2015). A literature review on immersive virtual reality in education: state of the art and perspectives. In *Conference proceedings of eLearning and Software for Education «(eLSE)*, number 01, pages 133–141. Universitatea Nationala de Aparare Carol I.
- Paredes, J. L. and Arce, G. R. (2011). Compressive sensing signal reconstruction by weighted median regression estimates. *IEEE Transactions on Signal Processing*, 59(6):2585–2601.
- Plets, G., Gheyle, W., Verhoeven, G., De Reu, J., Bourgeois, J., Verhegge, J., and Stichelbaut, B. (2012). Three-dimensional recording of archaeological remains in the altai mountains. *Antiquity*, 86(333):884–897.
- Quezada, E., Arribasplata, R., and Merchán, F. (2016). Fusión de imágenes con múltiples puntos de enfoque basado en sensado compresivo. *Prisma*, 4(1):38–41.
- Rodrigues, M., Kormann, M., Schuhler, C., and Tomek, P. (2013). Structured light techniques for 3d surface reconstruction in robotic tasks. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 805–814. Springer.
- Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. (2009). Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1–6. IEEE.
- Saaidi, A., Satori, K., et al. (2014). Multi-view passive 3d reconstruction: Comparison and evaluation of three techniques and a new method for 3d object reconstruction. In *Next Generation Networks and Services (NGNS), 2014 Fifth International Conference on*, pages 194–201. IEEE.
- Saíz, M. (2010). Reconstrucción tridimensional mediante visión estéreo y técnicas de optimización. *Madrid: Universidad Pontificia de Comillas*.

- Salvi, J., Fernandez, S., Pribanic, T., and Llado, X. (2010). A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8):2666–2680.
- Sarbolandi, H., Lefloch, D., and Kolb, A. (2015). Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer vision and image understanding*, 139:1–20.
- Shen, Y., Chen, H., Yao, C., and Qiao, Z. (2013). Super-resolution reconstruction of compressed sensing mammogram based on contourlet transform1. In *Proc. of SPIE Vol*, volume 8750, pages 87500M–1.
- Soeiro, J., Cláudio, A. P., Carmo, M. B., and Ferreira, H. A. (2016). Mobile solution for brain visualization using augmented and virtual reality. In *Information Visualisation (IV), 2016 20th International Conference*, pages 124–129. IEEE.
- Thali, M. J., Braun, M., Wirth, J., Vock, P., and Dirnhofer, R. (2003). 3d surface and body documentation in forensic medicine: 3-d/cad photogrammetry merged with 3d radiological scanning. *Journal of forensic sciences*, 48(6):1356–1365.
- Tropp, J. A. and Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666.
- Tropp, J. A. and Wright, S. J. (2010). Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958.
- Tusell, F. (2011). Análisis de regresión. introducción teórica y práctica basada en r.
- Vivek, P., Veenus, P., Dharun, V., and Sivasankar, K. (2015). The implications of compressive sensing in signal processing. In *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2015 International Conference on*, pages 517–521. IEEE.
- Wright, S. J., Nowak, R. D., and Figueiredo, M. A. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493.
- Yan-lin, S., Ai-ling, Z., You-bin, H., and Ke-yan, X. (2011). 3d geological modeling and its application under complex geological conditions. *Procedia Engineering*, 12:41–46.
- Zanchi, A., Francesca, S., Stefano, Z., Simone, S., and Graziano, G. (2009). 3d reconstruction of complex geological bodies: Examples from the alps. *Computers & Geosciences*, 35(1):49–69.
- Zhang, N. (2013). *Cellular GPU Models to Euclidean Optimization Problems: Applications from Stereo Matching to Structured Adaptive Meshing and Traveling Salesman Problem*. PhD thesis, Université de Technologie de Belfort-Montbéliard.