ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

Generación de modelos 3D mediante luz estructurada
(Generation of 3D models through structured light)

Para acceder al Título de

Graduado en
Ingeniería de Tecnologías de Telecomunicación

Autor: Fernando Manteca Fernández

Diciembre-2018

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Fernando Manteca Fernández
Director del TFG: Pablo Pedro Sánchez Espeso

Título: "Generación de modelos 3D mediante luz estructurada"
Title: "Generation of 3D models through structured light "

Presentado a examen el día: 3 de diciembre de 2018

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): Villar Bonet, Eugenio
Secretario (Apellidos, Nombre): Ugarte Olano, Iñigo
Vocal (Apellidos, Nombre): Sánchez Espeso, Pablo Pedro

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

AGRADECIMIENTOS

Tengo que agradecer a mi tutor Pablo, por todo lo que me ha ayudado con el proyecto siempre que lo necesitaba y por estar siempre dispuesto para resolver cualquier problema que me pudiera surgir.

A mis compañeros durante la carrera por todo lo que me han ayudado cuando teníamos trabajos y exámenes constantes y por todos los momentos que hemos pasado a lo largo de estos 4 años.

Al grupo Homeless, por todo el apoyo, los buenos momentos, los viajes, todas las experiencias que hemos vivido juntos y sobretodo por todo lo que me ayudaron a desarrollarme como persona durante mi año de Erasmus. Y a todas esas personas con las que pude trabajar en el Politecnico de Turín que me permitieron compartir vivencias y experiencias con personas de diversas nacionalidades y en diferentes idiomas, cosa que me enriqueció mucho como persona.

También agradecer a toda mi familia y amigos por estar ahí siempre que lo he necesitado.

Y el agradecimiento especial a Ana, Jose María y sobretodo a mis padres, que a parte de estar siempre ahí, me han permitido siempre hacer lo que me gustaba y me han apoyado con todo siempre.

Muchas gracias.

Índice general

Índice de figuras

Índice de cuadros

CAPÍTULO 1

Resumen

Hoy en día, la representación en tres dimensiones de objetos es un campo de creciente actividad, tanto en el ámbito industrial como académico. Por ello se han propuesto numerosos métodos para llevar a cabo este tipo de reconstrucciones, entre los que destaca el estudiado en este proyecto, el escáner 3D basado en luz estructurada. Además, para llevar a cabo este proyecto se emplea un sistema que no utiliza un montaje o hardware complicado, requiriendo únicamente un proyector estándar para generar los patrones sobre el objeto y 2 cámaras USB de bajo coste para capturar. Esto hace que el proyecto sea económico al tiempo que permite obtener reconstrucciones con una precisión aceptable. Para el desarrollo de los algoritmos, se emplea el lenguaje de programación C++, así como, funciones de la biblioteca OpenCV.

CAPÍTULO 2

Abstract

Nowadays, 3D representation of objects is a field of growth in industrial and academic fields. For example, numerous methods have been developed to carry out this type of reconstructions, a distinguished example is the one studied in this project, the 3D scanner based on structured light. In addition, to carry out this project it is used a system that does not use a complicated assembly or hardware, requiring a standard projector to generate the patterns on the object and 2 low cost USB cameras to capture. This makes the project economical while allowing a fairly acceptable reconstruction. For the development of the algorithms, the C ++ programming language is used, as well as functions of the OpenCV library.

CAPÍTULO 3

Introducción

3.1. Contexto

El propósito de un escáner 3D es crear una nube de puntos a partir de muestras tomadas en la superficie de un objeto. Estos puntos son empleados para extrapolar la forma del objeto. Actualmente el avance tecnológico en la digitalización 3D está creciendo a pasos agigantados permitiendo digitalizar nuevas y más complicadas formas con mayor precisión, lo que ha permitido ampliar su campo de aplicación, siendo innumerables las aplicaciones que pueden llegar a tener este tipo de tecnologías, desde el área médico-biológico hasta la automoción o robótica.

El mundo físico es tridimensional, pero las cámaras tradicionales solo pueden capturar imágenes en 2 dimensiones. Esto hace que su percepción sea limitada y para tener una mejor visión y caracterización de los objetos a su alrededor es necesario emplear técnicas para obtener modelos 3D, de forma que sea posible obtener una mejor percepción de la realidad y sus características. En los últimos años, estas tecnologías se han popularizado, siendo uno de los temas con mayor interés tecnológico, ya que ofrecen numerosas ventajas respecto a la representación 2D.

La reconstrucción 3D es el proceso por el cual objetos reales son reproducidos en la memoria de una computadora, manteniendo sus características físicas. Existen numerosas técnicas de reconstrucción, cuyo objetivo principal es el de ser capaces de conexionar los puntos que forman las superficies del objeto, a partir de imágenes capturadas por cámaras. En el capítulo 6 se presentan algunos de estos algoritmos.

La eficiencia del algoritmo es la que define la calidad final del mallado. Si partimos de un conjunto de puntos mal representado, existirán puntos definidos que no cumplan las condiciones óptimas para el mallado. Los puntos que se encuentran muy cercanos entre si, los puntos ruidosos o los puntos redundantes no ofrecen ninguna información para la reconstrucción. Por ejemplo, si se quiere representar un cubo en el espacio, simplemente con ocho puntos y doce triángulos sería suficiente; el resto de la información sería redundante.

A continuación se puede observar 1 imagen, en la que se representa un cubo en 2 y 3

dimensiones. Se puede observar perfectamente, como la sensación de volumen aporta una información extra que permite caracterizar perfectamente el objeto.

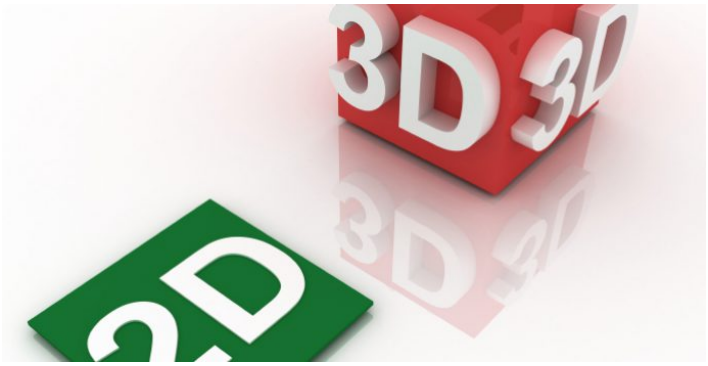


Figura 3.1: Imagen en 2 dimensiones vs imagen en 3 dimensiones

El proceso de digitalización depende en gran medida del modelo que se esté analizando. Por ello se emplean diferentes tecnologías y técnicas dependiendo de la aplicación o las dimensiones del objeto que se quiera estudiar. Por ejemplo para digitalizar algo de tamaño muy grande, en el que se requiera un menor número de detalles en su superficie, se utiliza la tecnología láser, como en el caso de la digitalización de un yacimiento arqueológico. Pero en cambio resultaría muy costoso intentar digitalizar estos trabajos con un escáner de luz estructurada, debido a su limitada área de trabajo.

3.2. Objetivos y motivaciones del proyecto

El objetivo principal que se quiere llevar a cabo con el desarrollo de este trabajo fin de grado, es diseñar e implementar un escáner para la reconstrucción 3D de objetos. Para ello se ha escogido la técnica de luz estructurada. En esta técnica para obtener la forma y características de un objeto, se proyectan patrones de luz sobre el objeto y mediante 2 cámaras se captura la deformación que han sufrido los patrones al proyectarse sobre el objeto. De esta forma, haciendo uso de una serie de algoritmos que serán descritos más adelante, seremos capaces de obtener el modelo 3D del objeto.

Otra de las motivaciones de este trabajo fin de grado, es la de ser capaz de obtener modelos 3D haciendo uso de un hardware muy sencillo, para poder comparar el sistema desarrollado con otros productos del mercado diseñados exclusivamente para este fin y con un coste más elevado. El hecho de escoger la técnica de luz estructurada es debido a que nos permite obtener reconstrucciones 3D bastante robustas con un bajo coste, de forma rápida y con una alta calidad.

3.3. Estructura de la memoria

En cuanto a la estructura del proyecto, esta memoria está dividida en 8 partes diferenciadas:

- En primer lugar, se escriben los agradecimientos.
- Capítulo 1 y 2, constan de un breve resumen, tanto en español como en inglés, del proyecto llevado a cabo.
- Capítulo 3, en el que se va a introducir las bases del modelado 3D y se enuncian las motivaciones y objetivos de este proyecto.
- Capítulo 4, en esta parte se analiza el estado del arte comentando las diferentes tecnologías que existen para obtener modelos 3D de objetos. En especial se contará la técnica de luz estructurada, que es la empleada para llevar a cabo este trabajo fin de grado.
- Capítulo 5, en este apartado se profundiza un poco más en la técnica de luz estructurada, al ser esta la escogida en nuestro proyecto para digitalizar los objetos.
- Capítulo 6, en este capítulo se explica en detalle el proyecto y el sistema que se ha empleado. Se enuncian y detallan los procesos de captura de imágenes, calibración y reconstrucción para la generación de la malla 3D. Se hace hincapié en los algoritmos de triangulación y reconstrucción, que son esenciales. Por otra parte, también se explica como se obtienen las regiones de sombra y la decodificación de los patrones.
- Capítulo 7, en esta parte se muestran los resultados obtenidos y se hace una pequeña comparación con un sistema comercial, mucho más costoso (Intel Real Sense Camera ZR300).
- Capítulo 8, en este último capítulo simplemente se aportan las conclusiones obtenidas con la realización del proyecto, se nombrarán posibles aplicaciones del sistema y algunas mejoras que podrían introducirse.

CAPÍTULO 4

Estado del arte

En el campo de la reconstrucción 3D existen gran cantidad de soluciones tecnológicas. En este apartado se estudiarán algunas de ellas.

En la figura 4.1 se muestra una clasificación de los diferentes tipos de técnicas de reconstrucción que existen. En esta sección se mencionarán las principales diferencias entre unas y otras y se presentarán algunas de sus principales características.

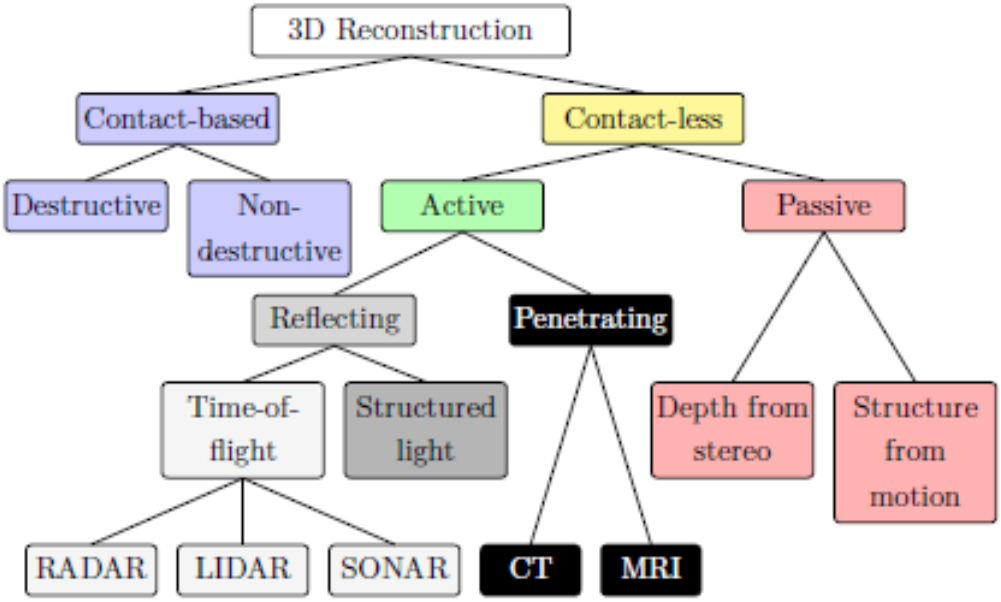


Figura 4.1: Tipos de escáneres para la reconstrucción 3D[5](pag 6)

Observamos una división básica en 2 tipos de técnicas de escaneados en 3D, dependiendo de si hay contacto con el objeto o no:

- **Técnicas basadas en el contacto con el objeto**

  Esta técnica requiere de contacto con el objeto, mediante una serie de sensores sobre su superficie que permiten llevar a cabo la reconstrucción. Existen algunas desventajas muy claras de esta técnica. Por un lado, requieren tocar la pieza, esta puede ser modificada o incluso dañada lo que supone una gran desventaja respecto a otro tipo de técnicas. Por otro lado, la reconstrucción suele ser un proceso lento respecto a otras técnicas que se explicarán a continuación. Su uso principal se sitúa en el control de procesos de fabricación, ya que, permite observar si las piezas cumplen con las características que se les requiere.

  En la siguiente imagen se aprecia perfectamente el funcionamiento de este tipo de escáneres, que incluyen un brazo explorador que recorre el objeto con una punta sensible al contacto.

Figura 4.2: Escáner 3D de contacto

- **Técnicas sin contacto con el objeto**

  Este tipo de reconstructor no entra en contacto directo con la pieza, lo que supone una ventaja para el estudio de objetos frágiles o de gran valor.

  A su vez, los escáneres de contacto se pueden dividir en 2 subgrupos:

  - Activos

    Estos escáneres, emiten algún tipo de señal que es reflejada por el objeto, siendo la señal reflejada analizada para capturar la geometría del objeto en estudio. A su vez estas técnicas se dividen en:

- Tiempo de Vuelo

  Este tipo de sistemas envían un pulso de luz al objeto y cronometran el tiempo que dicho pulso invierte en el viaje desde el emisor hasta el objeto y desde este al receptor. Este tiempo se calcula teniendo en cuenta el valor de la velocidad de la luz. Por tanto, la distancia entre el escáner y la superficie viene dada por la expresión (c*T)/2, donde c='velocidad de la luz' y T= 'tiempo de ida y vuelta del pulso luminoso'.

  Este proceso debe llevarse a cabo para todos los puntos del objeto, moviendo el distanciómetro con algún otro tipo de técnica óptica y/o mecánica.

  Es evidente que la precisión de este tipo de sistemas va a depender de la precisión con la que se pueda medir el tiempo del pulso, que normalmente es del orden de nano/femto segundos. Un ejemplo de este tipo de sistemas es el LIDAR (Light Detector and Ranging).

- Triangulación

  El escáner láser de triangulación 3D, emite un haz de luz láser que incide sobre un objeto. Mediante una cámara se busca el punto donde el láser aparece en el objeto. Dependiendo de la distancia a la que el láser choque con la superficie, el punto aparece en un lugar u otro en el sensor de la cámara, lo que permite calcular la distancia.

  El sistema se basa en que el punto del láser, la cámara y el láser forman un triángulo. Todas las dimensiones del triángulo pueden ser obtenidas a partir de los datos conocidos (distancia entre el láser y la cámara, ángulo de la cámara y el ángulo del emisor láser). Este sistema puede llegar a alcanzar altas precisiones. Con respecto a los escáneres de tiempo de vuelo, los escáneres de triangulación son más precisos, pero tienen un campo de acción de unos cuantos metros, mientras que los de tiempo de vuelo pueden operar en radios de acción de hasta kilómetros. La figura 4.3 muestra un esquema del sistema de triangulación.

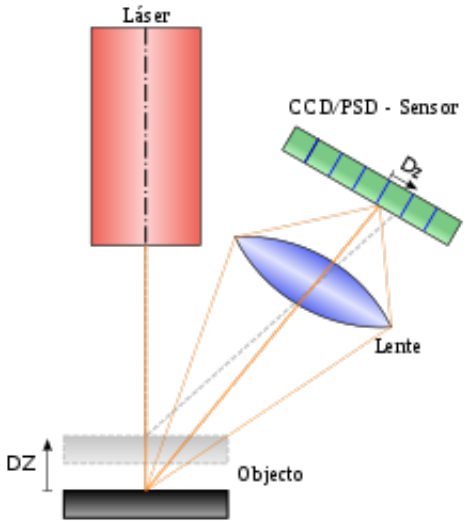


Figura 4.3: Sensor laser de triangulación

A continuación, se muestra la ecuación básica que se emplea para la triangulación:

$R = B\frac{\sin(\theta)}{\sin(\alpha+\theta)}$, donde R es la medida de profundidad, B es la distancia entre el láser y la cámara, θ es el ángulo del láser y α es el ángulo que forma la cámara

- Luz estructurada

  Los escáneres 3D de luz estructurada proyectan un patrón de luz en el objeto y analizan la deformación del patrón producida por la forma del objeto.

  La principal ventaja de este tipo de escáner es la velocidad, ya que en vez de escanear un punto, se escanean múltiples líneas de frame simultáneamente. Este sistema va a ser explicado y analizado a fondo en el próximo capítulo, ya que, es el empleado en el desarrollo del proyecto.

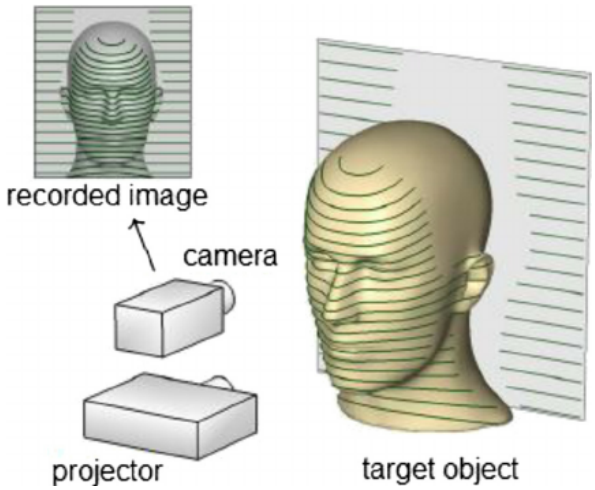
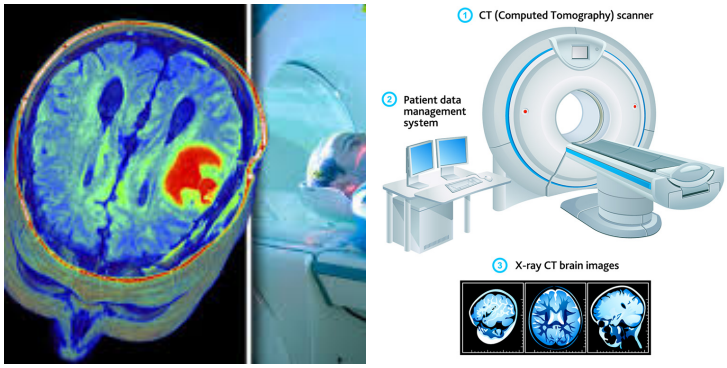


Figura 4.4: Escáner 3D basado en sistema de luz estructurada

- Imagen por resonancia magnética (MRI) y por tomografía computada (CT scanner)

  Estas técnicas son menos empleadas en el ámbito de la visión por computador por lo que se comentarán en menor detalle. En lo que respecta a las imágenes por resonancia magnética, esta técnica utiliza el fenómeno de la resonancia magnética nuclear para obtener información sobre la estructura y composición del objeto a analizar. Su principal aplicación es la generación de imagen médica. En cuanto a la tomografía computada, la técnica se basa en realizar muchas mediciones de rayos X tomadas desde diferentes ángulos, para formar imágenes de áreas específicas de un objeto. Posteriormente mediante procesado, se puede obtener el modelo tridimensional del objeto. En la siguiente figura se aprecian ejemplos de ambas técnicas.



(a) Escáner por resonancia magnética

(b) Escáner por tomografía computada

- Técnicas pasivas

  Los escáneres pasivos no emiten ningún tipo de señal, sino que su funcionamiento se basa en observar la radiación del ambiente reflejada por el objeto. La principal ventaja de este tipo de métodos es que pueden llegar a ser muy baratos, al no necesitar hardware especifico. Dentro de las técnicas pasivas, podemos identificar 2 grupos:

  - Depth from stereo

    Esta técnica requiere de 2 cámaras captando al tiempo una escena, para poder generar el modelo 3D.

    En la imágen 4.6 observamos como las imágenes son proyecciones de puntos 3D en una superficie 2D. Los puntos X_1,X_2,...se proyectan en la imágen de la derecha y de la izquierda en coordenadas diferentes.

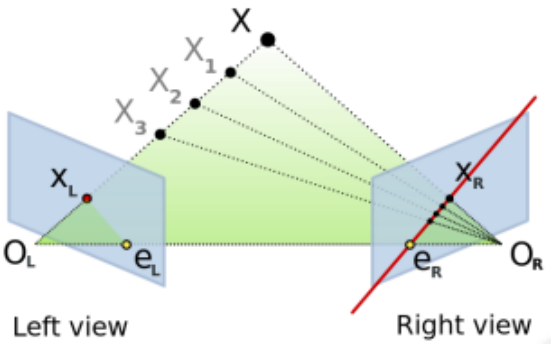
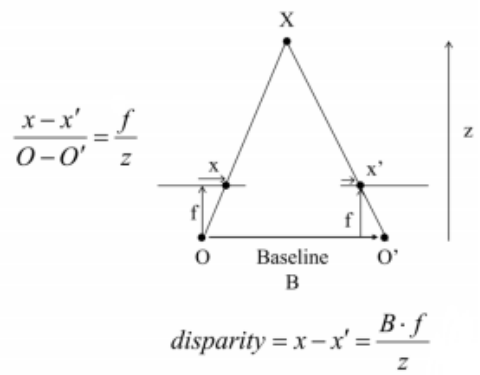


Figura 4.6: Depth from stereo

Para hallar los valores de profundidad, se realiza una triangulación y se determina la distancia (z) como se muestra a continuación.



En este caso, $D = \frac{f*b}{d*ps}$, siendo f= 'longitud focal', b= 'linea de referencia', d= 'valor de disparidad', ps= 'tamaño de pixel' y D='profundidad (depth)'.

La línea de referencia se refiere a la distancia entre la cámara izquierda y la derecha. El valor de disparidad hace referencia a la diferencia entre la posición del píxel entre las imágenes de las 2 cámaras.

La complejidad de este método reside en encontrar el mismo punto en ambas cámaras ('matching' entre imágenes).

- Structure from motion

Esta es una técnica muy efectiva para obtener modelos 3D en caso de movimiento de la cámara o del objeto.La mayoría de los frames de video difieren entre sí debido al movimiento de la cámara o el objeto. Por tanto, los valores de profundidad para un determinado frame pueden estimarse analizando las diferencias entre frames consecutivos.

En resumen, el procedimiento consistirá en la extracción de unos puntos singulares comunes a todas las fotografías tomadas, utilizando para ello el algoritmo SIFT (algoritmo usado en visión artificial para extraer características relevantes de las imágenes). Esto nos permitirá poder hacer una clasificación de los puntos para reconstruir la imagen desde los diferentes puntos de vista posibles.

En la figura 4.7 podemos observar como se va moviendo la cámara para tomar los diferentes frames, modificándose la posición de las esquinas del cubo (puntos singulares).

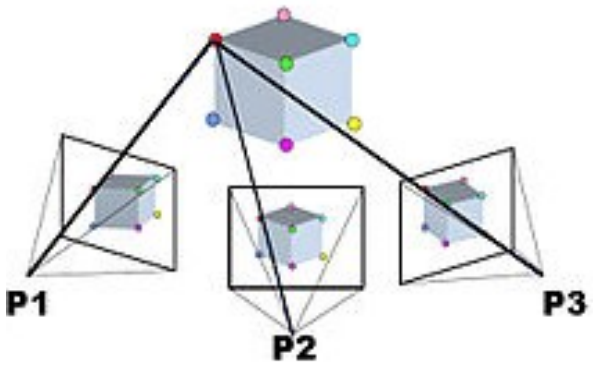


Figura 4.7: Esquemático representando la técnica SFM

El hecho de tener que encontrar una referencia para poder digitalizar la escena, hace que estas técnicas requieran mucho cómputo. Por esto, la técnica de luz estructurada es una mejor elección, ya que se emiten unos patrones, en los que se manda una referencia, la cual es de gran ayuda a la hora de reconstruir la escena.

CAPÍTULO 5

Escáner de luz estructurada

El escáner de luz estructurada está compuesto por un proyector y 1 ó 2 cámaras. En el caso de querer escanear objetos de tamaño muy reducido se requiere también una placa giratoria automatizada, que ayude a capturar el volumen con mayor facilidad. Para la digitalización, el objeto es irradiado con un patrón de luz conocido. El proyector ilumina el objeto con patrones de rayas blancas y negras, colocadas de manera paralela y con anchura variable. Las cámaras registran la imagen que produce el patrón sobre el objeto, creando una secuencia temporal con los diferentes valores de brillo.

La ventaja de los escáneres 3D de luz estructurada es la velocidad: en vez de escanear un punto a la vez, escanean múltiples puntos o un campo de visión entero, lo que reduce o elimina el problema de la deformación por movimiento. Un inconveniente puede ser la mala detección de superficies transparentes o reflectantes, aunque se han propuesto algunas soluciones.

Además, para poder tener buenas reconstrucciones, este tipo de sistema necesita conocer con precisión la localización exacta del proyector, que debe permanecer fijo durante el proceso de captura de las imágenes. Esto genera otra limitación, ya que solo se puede digitalizar la parte iluminada. Esta es una de las principales limitaciones de esta técnica y por ello no es recomendable para escenas u objetos de grandes dimensiones.

A la hora de elegir los patrones que más nos convengan, se deben analizar aquellos que se adapten mejor al objeto de estudio. Por ello se han propuesto diferentes tipos de patrones, como los puntos, rejillas con láseres de diferentes colores, tiras paralelas en blanco y negro, etc. Si se eligen puntos, se debe recorrer el objeto por toda su superficie, tomando una gran cantidad de puntos y pudiendo perder algunas zonas del objeto. En cambio, con otras técnicas en las que se ilumina un conjunto de puntos, la captura de imágenes es mucho más rápida.

Hay varios esquemas para codificar la información en secuencias de patrones, las más populares son las que emplean código binario o las de código de gray. En ambos casos, la información se codifica por separado en los ejes x e y. Por tanto, un proyector con

resolución P_{resx} y P_{resy}, tendra Ncols = log2 (Presx) y Nrows =log2 (Presy). Por ejemplo, el proyector empleado tiene una resolución 1024x768, lo que dará como resultado Ncols = 10 y Nrows = 10, es decir, un total de 20 patrones.

A continuación, se muestra un esquema, con diferentes tipos de patrones, Figura 5.1.

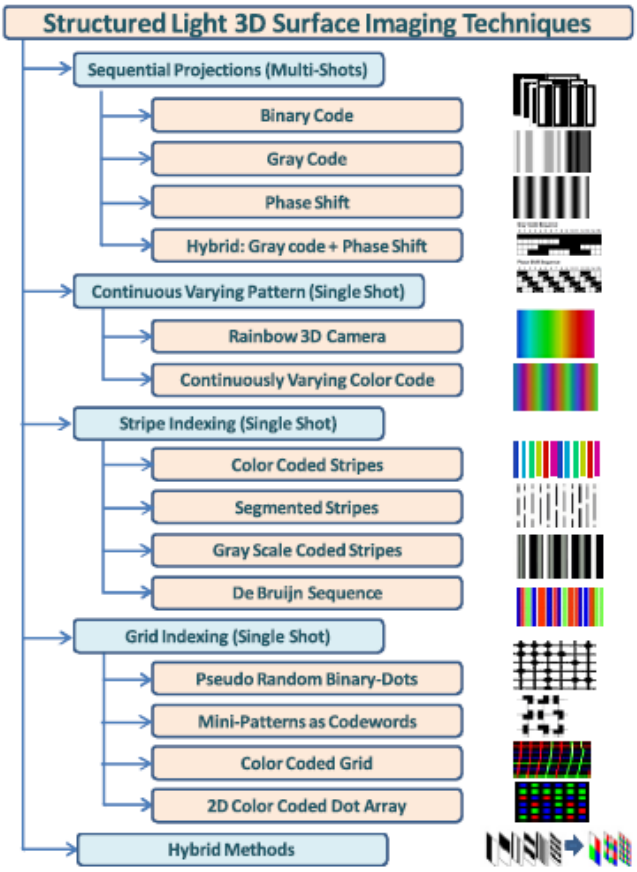


Figura 5.1: Tipos de patrones emitidos

Tal y como se ilustra en la figura anterior existen diferentes tipos de patrones para obtener modelos 3D mediante luz estructurada.

Si el objeto 3D está estático y la aplicación no impone restricciones estrictas sobre el tiempo de adquisición, la técnica de disparo múltiple es más recomendable y nos proporciona resultados más precisos. Sin embargo, si el objeto está en movimiento, se deben utilizar las técnicas de disparo único para adquirir una imagen de la superficie 3D del objeto. Estas técnicas de disparo único se clasifican a su vez en tres categorías, tal y como se aprecia en el esquema. Además existen la posibilidad de combinar diferentes técnicas juntas para lograr algunos beneficios adicionales.

En nuestro proyecto se ha apostado por una técnica de disparo múltiple, ya que el objeto que vamos a analizar se mantiene estático, por tanto, estas técnicas nos pueden dar una mayor precisión.

Ahora se enunciarán y explicarán las diferentes técnicas que existen de generación de patrones, en base al esquema de la figura 5.1.

- Proyecciones secuenciales (disparo múltiple)

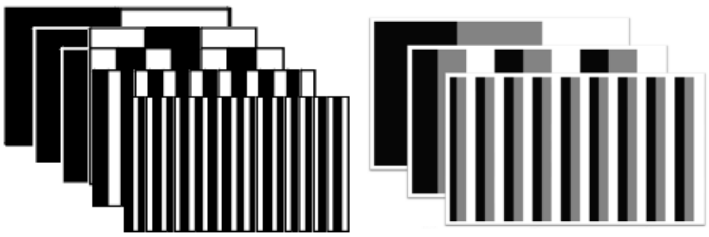
  Código binario

  Los patrones de código binario usan rayas blancas y negras, formando una secuencia de patrones de proyección, de forma que cada punto en la superficie del objeto posee un único código binario diferente al del resto de puntos. Esta técnica es bastante fiable y poco sensible a las características de la superficie. Sin embargo, para obtener buenas resoluciones espaciales, se deben proyectar un alto número de patrones.

  Código gray

  Esta técnica reduce el numero de patrones necesarios para obtener una alta resolución en la imagen 3D. Esto se debe a que se pueden usar distintos niveles de intensidad para generar el patrón, en vez de 2 como en el caso binario.

  En la figura siguiente, podemos apreciar las diferencias entre ambas técnicas, en la que se observa esos diferentes niveles de intensidad en el caso de los códigos de gray.



(a) Patrones de código binario

(b) Patrones de código de gray

Figura 5.2: Estudio comparativo entre códigos binarios y de gray

Código de cambio de fase

En este caso, un conjunto de patrones sinusoidales se proyectan sobre la superficie del objeto. Las intensidades de luz de cada fleco proyectado, viene dada por las siguientes ecuaciones:

$$I_1(x,y) = I_0(x,y) + I_{mod}(x,y)cos(\phi(x,y) - \theta)$$
$$I_2(x,y) = I_0(x,y) + I_{mod}(x,y)cos(\phi(x,y))$$
$$I_3(x,y) = I_0(x,y) + I_{mod}(x,y)cos(\phi(x,y) + \theta),$$

siendo I_0, la componente continua, I_{mod} es la señal de modulación en amplitud, ϕ es la fase y θ es el ángulo constante de cambio de fase.

A partir de la fase envuelta, se puede obtener la fase absoluta:

$$\phi' = \arctan[\sqrt{3}\frac{I_1(x,y) - I_3(x,y)}{2I_2(x,y) - I_1(x,y) - I_3(x,y)}]$$

Las coordenadas 3D se calculan con la diferencia entre la fase medida y el valor de fase desde un plano de referencia. $Z \approx \frac{L}{B} d\alpha \frac{L}{B}(\phi - \phi_0)$

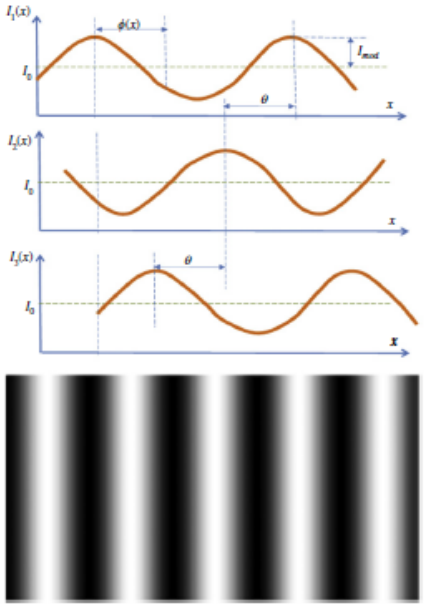


Figura 5.3: Patrones de cambio de fase

La figura 5.4 ilustra el sistema para calcular la profundidad en bases al valor de fase.

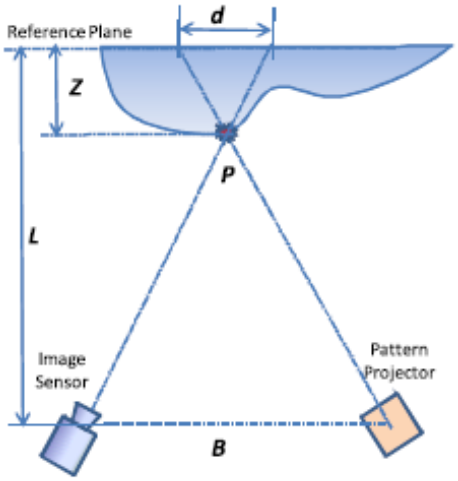


Figura 5.4: Calculo de profundidad en base a la fase

Código híbrido entre gray y de cambio de fase

Hay dos problemas principales que se pueden dar al emplear técnicas de cambio de fase. En primer lugar, los métodos basados en obtener la fase envuelta no dan la fase absoluta. Por otra parte, si dos superficies tienen una discontinuidad de más de 2π, entonces ningún método basado en obtener la fase envuelta reconstruirá correctamente estas dos superficies. Estos problemas se pueden resolver utilizando una combinación de las técnicas que emplean código gray y las técnicas de cambio de fase. La Figura 5.5 muestra un ejemplo de este tipo de combinación en una secuencia de codificación de 32 bandas. El código gray determina el rango absoluto de fase, mientras que el código de cambio de fase ofrece resolución de subpíxeles más allá del número de franjas proporcionadas por el código gay. Sin embargo, los métodos híbridos requieren un mayor número de proyecciones y no se prestan bien a las imágenes en 3D de objetos dinámicos

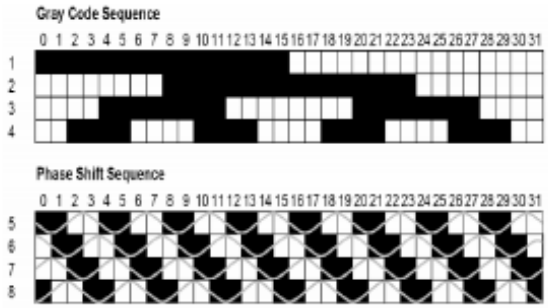


Figura 5.5: Combinación de códigos de gray con códigos de cambio de fase

- Patrones de color variables (disparo simple)

  El principal inconveniente de las técnicas de proyección secuencial es su incapacidad para adquirir el objeto 3D cuando existe movimiento dinámico. En este

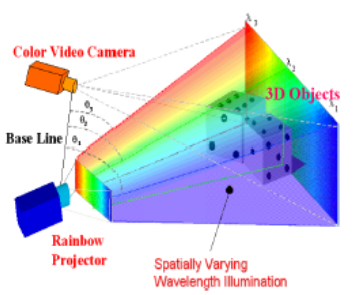
apartado se presentan algunas técnicas de un solo disparo, que soportan el movimiento en el objeto. Estas técnicas aprovechan la información de color y requieren solo una imagen del objeto bajo la iluminación de un patrón en color, para obtener la totalidad de la reconstrucción 3D de la escena.

Cámara 3D de arcoiris

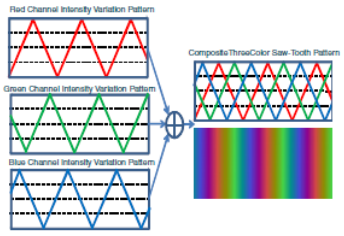
La cámara 3D de arco iris, proyecta una iluminación de longitud de onda variable sobre la superficie del objeto. La geometría fija del proyector establece una correspondencia uno a uno entre el ángulo de proyección de un plano de luz y una longitud de onda espectral particular, proporcionando así puntos de referencia fáciles de identificar en la superficie. Los valores de rango 3D correspondientes para cada píxel individual se pueden calcular utilizando el principio de triangulación. La digitalización es obtenida en una sola instantánea a la velocidad de captura de la cámara, lo que supone que el sistema sea muy rápido.

Patrones de codificación de colores con cambio continuo

Es posible componer varios patrones de colores, que varían continuamente. Esto sirve para codificar la información de ubicación espacial. Se podría realizar esta técnica construyendo un patrón de variación de intensidad para cada canal de color de un proyector, de forma que cuando se sumen los patrones de los canales individuales de color, darían lugar a un patrón de color que varía continuamente. En la figura siguiente se muestran 2 imágenes que ilustran las 2 técnicas explicadas en este apartado.



(a) Cámara 3D de arcoirris



(b) Patrones de codificación de colores con cambio continuo

- Patrones de rayas (disparo único)

Los patrones de rayas, son necesarios para lograr una reconstrucción de superficie 3D robusta, ya que el orden en que se observan las rayas no es necesariamente igual que el orden en que se proyectan las franjas. Esto se puede deber al paralaje inherente que existe en imágenes de superficie 3D basadas en triangulación, o a una posible falta de rayas en la imagen adquirida, normalmente debido a la

oclusión de las características que posee la superficie del objeto.

Patrones de rayas usando colores

Los sensores de imagen de color, tienen normalmente 3 canales independientes. La combinación lineal de estos puede dar lugar a un numero muy elevado de colores. Este hecho sirve para mejorar la precisión de las imágenes en 3D y también para reducir el tiempo de adquisición, pudiendo incluso usarse en tiempo real.

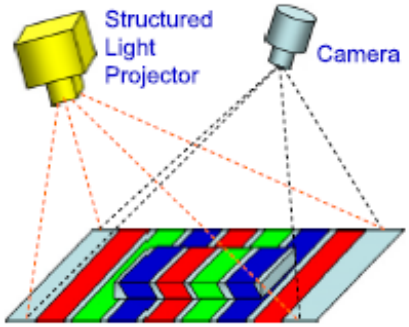


Figura 5.7: Patrones de rayas empleando colores

Patrones de rayas usando segmentos

Para distinguir una franja de otras, se pueden agregar algunos patrones de segmento a cada banda, de manera que, al realizar la reconstrucción 3D, el algoritmo puede utilizar el patrón de segmento único de cada raya para distinguirlos. Este método, solo se aplica a un objeto con una superficie lisa y continua, en el que, la distorsión del patrón debido a la forma de la superficie no es grande.

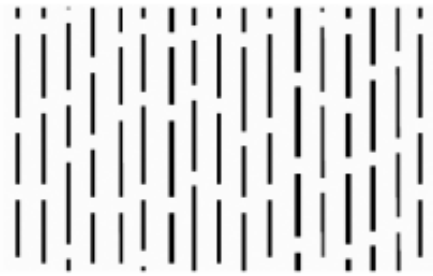


Figura 5.8: Patrones de rayas empleando segmentos

Patrones de rayas usando escala de grises repetidas

Si se usan mas de 2 niveles de intensidad, es posible organizar los niveles de intensidad de las franjas de modo que cualquier grupo de franjas tiene un patrón

de intensidad único dentro de un período. El proceso de coincidencia de patrones, comienza con una correlación de la intensidad de la imagen adquirida con la intensidad de la imagen proyectada. Una vez que se encuentra una coincidencia, se realiza una búsqueda adicional, buscando coincidencias en subgrupos de nivel de grises.

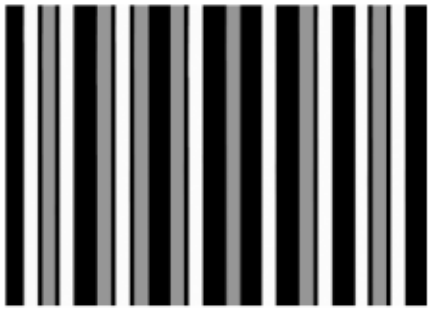


Figura 5.9: Patrones de rayas empleando escalas de grises repetidas

Patrones de rayas basados en la secuencia De Bruijn

Una secuencia "De Bruijn"de rango "n" en un alfabeto de tamaño k, es un ciclo de palabras en la que cada una de las palabras k_n, de longitud n, aparece exactamente una vez a medida que nos movemos por el ciclo. Esta característica única de la secuencia De Bruijn, hace que la decodificación del patrón sea una tarea más fácil.

A continuacion, en la Figura 5.10, se puede ver un ejemplo de generación de patrones empleando la secuencia De Bruijn.

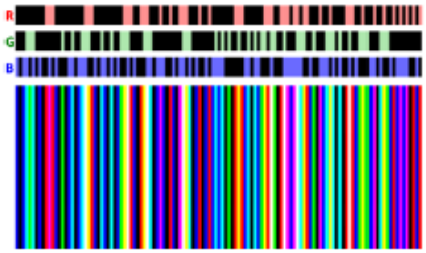


Figura 5.10: Patrones de rayas basados en la secuencia De Bruijn

- Patrones de cuadrículas (disparo único)

  El concepto en el que se basan estas técnicas de cuadrícula o red, es etiquetar únicamente cada subventana en el patrón 2D proyectado, de manera que el patrón en cualquier subventana es único y totalmente identificable.

Array binario pseudo-aleatorio

Una estrategia de indexación de cuadrícula es, usar una matriz binaria pseudoaleatoria, para producir ubicaciones de cuadrícula que se pueden marcar con puntos u otros patrones, de modo que, el patrón codificado para cualquier subventana, sea único. Esta matriz, de tamaño n_1 x n_2, se codifica utilizando una secuencia pseudoaleatoria, tal que, cualquier subventana perteneciente a la matriz es única y define completamente su coordenada absoluta (i,j) dentro de la matriz. El patrón de codificación de la matriz, se genera en base a una secuencia binaria pseudoaleatoria empleando el método del polinomio primitivo módulo 2^n.

Mini-patrones utilizados como palabras código

En lugar de utilizar una matriz binaria pseudoaleatoria, como en el caso anterior, se puede emplear una matriz de multivalores pseudoaleatorios. Se puede representar cada valor con un mini patrón, formando así un patrón de proyección en cuadrícula. Con los códigos definidos, la matriz de multivalores pseudoaleatorios se puede convertir en un patrón de proyección, con subventanas únicas. Se puede ver un ejemplo de uso de este tipo de técnica, en la figura 5.11.

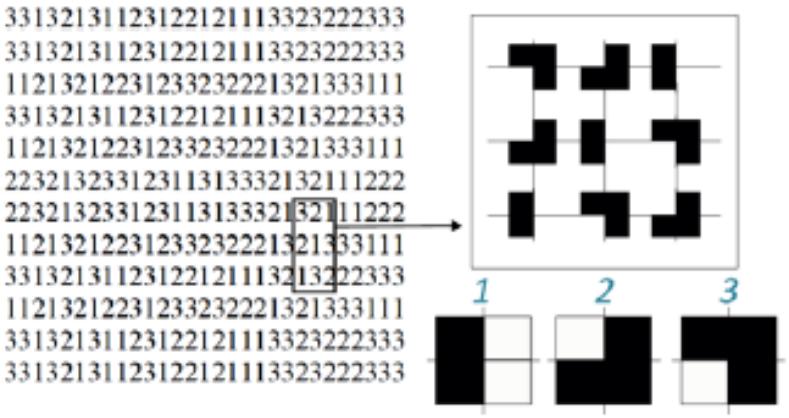


Figura 5.11: Uso de mini patrones como códigos para formar patrones de cuadrícula

Cuadrículas codificadas por colores

Otra estrategia es codificar por colores, tanto en franjas verticales como horizontales, para que se pueda lograr una indexación de cuadrícula en 2 dimensiones. Los esquemas de codificación de franja vertical y horizontal pueden ser iguales o diferentes, dependiendo de las aplicaciones. No hay garantía de la singularidad de las subventanas, pero las franjas de colores en ambas direcciones pueden ayudar a la decodificación del patrón, para establecer la correspondencia. En la figura 5.12, se puede ver los patrones formando una cuadrícula de colores.

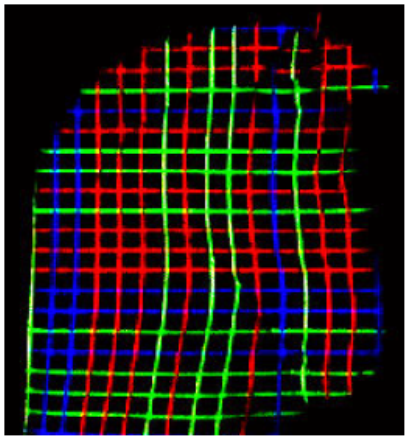


Figura 5.12: Uso de cuadrículas de colores

Array 2D de puntos codificados por color

Existen métodos alternativos para generar la matriz pseudoaleatoria. Un ejemplo se puede ver en la figura 5.13, en el que se representa, paso a paso, la generación de los patrones para esta técnica.

Se observa un array de dimensión 6x6 con tamaño de subventana de 3x3. El procedimiento de cálculo incluye los pasos que se comentan a continuación. En primer lugar, se llena la parte superior izquierda del array con un patrón elegido al azar. En segundo lugar, se añade una columna de tres elementos a la derecha con palabra de código aleatorio, comprobando previamente la unicidad de la subventana anterior. En tercer lugar, se deben seguir agregando columnas hasta que todas se llenen con código aleatorio, verificando a su vez la singularidad de las palabras y de las subventanas. Del mismo modo, se agregan al azar filas en dirección hacia abajo desde la posición inicial de la subventana. Por último, se agregan nuevas palabras de código aleatorio a lo largo de la dirección diagonal, repitiendo estos procedimientos hasta que todos los puntos estén llenos de colores.

Esta técnica ha resultado beneficiosa para muchos casos, pero no se garantiza para todos los tamaños posible de array.

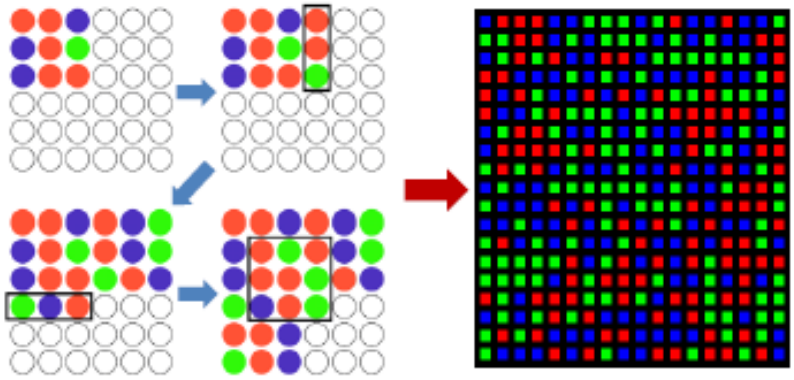


Figura 5.13: Array 2D de puntos codificados por color

- Métodos híbridos

  Existen formas de incrementar algunas de las características de las reconstrucciones 3D. Para ello, se pueden combinar algunas de las técnicas explicadas en este capítulo. En la figura 5.14, se muestra un ejemplo en el cual se combinan 2 códigos de tiras unidimensionales para formar un patrón de rejilla bidimensional.

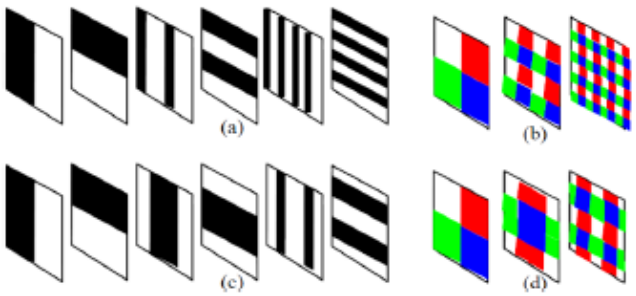


Figura 5.14: Combinaciones híbridas de diferentes técnicas de generación de patrones

CAPÍTULO 6

Sistema desarrollado para la creación de modelos 3D mediante luz estructurada

El sistema que se ha desarrollado emplea un proyector para emitir patrones de luz sobre el objeto. Estos patrones tienen una estructura conocida. Además, se hace uso de 2 cámaras que capturan las imágenes del objeto mientras se están proyectando los patrones sobre él. Cada patrón generado se distorsionará según la geometría del objeto que se está escaneando. Para poder identificar las correspondencias entre píxeles, el patrón proyectado se codifica de tal manera que cada píxel pueda ser identificado de forma única en las imágenes captadas por las cámaras. Esto hace que este sea un método eficiente y preciso para calcular la correlación entre los píxeles. Por último, utilizando esta relación entre píxeles, es posible calcular sus posiciones 3D, mediante unos algoritmos que serán explicados más adelante. En la Figura 6.1, se puede observar una representación del sistema empleado, con las 2 cámaras y el proyector emitiendo los patrones sobre la pieza que se quiere reconstruir.

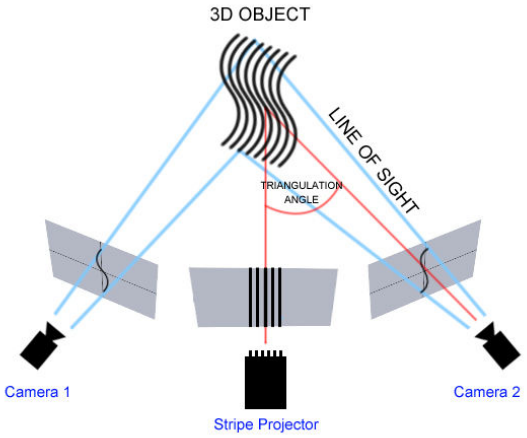


Figura 6.1: Sistema empleado para la reconstrucción 3D

A continuación, (Figura 6.2) se muestra un esquema donde se pueden observar los pasos que se siguen en el proyecto para llegar a tener la reconstrucción 3D del objeto. Estos pasos, serán explicados en detalle en las próximas secciones.

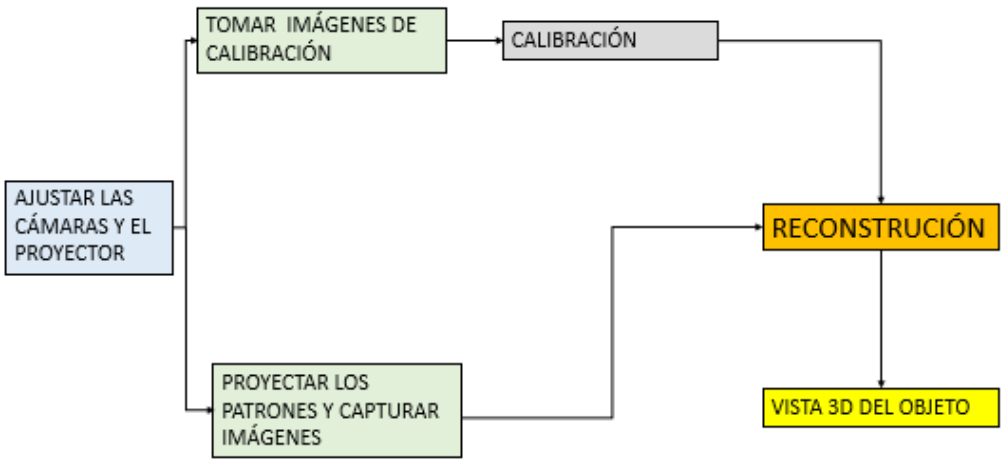


Figura 6.2: Esquema de los pasos seguidos para llevar a cabo la reconstrucción 3D

El sistema ha sido diseñado de acuerdo con los siguientes requisitos: precisión media-alta, fácil uso y coste asequible.

Al proyectar una banda estrecha de luz sobre una superficie con forma tridimensional, se produce una línea de iluminación. Esta línea aparece distorsionada desde otras perspectivas distintas a la del proyector. Esto puede utilizarse para reconstruir la forma geométrica de la superficie. Además, al tratarse de un proyector convencional, se amplía la flexibilidad al permitir diferentes patrones de luz. En la siguiente sección se explica la captura de las imágenes y se describen los patrones elegidos y las razones por las cuales han sido seleccionados para esta aplicación concreta.

6.1. Software

Lenguaje C++ haciendo uso de la biblioteca OpenCV

Cabe reseñar que la programación de todo el sistema se ha realizado empleando el lenguaje de programación C++, haciendo además uso de la biblioteca OpenCV.
OpenCV(Open Source Computer Vision Library) es una biblioteca libre desarrollada originalmente por Intel, que soporta diferentes lenguajes, como pueden ser C, C++ o python. Esta librería es empleada en el proyecto debido a su gran potencial y las

numerosas funciones que tiene definidas para el tratamiento de imágenes, extracción de parámetros de calibración de las cámaras, manejo fácil de las cámaras entre otras.

Todo el código se ejecuta en un portátil con procesador i7, conectando las cámaras a sendos puertos USB de este y el proyector a la salida VGA.

6.2. Hardware

Camara Logitech C270

Se emplean 2 cámaras iguales (Logitech C270), como la mostrada en la Figura 6.3. Estas camaras son de gama media, con muy bajo coste y son capaces de capturar imágenes de hasta 1280 x 720 píxeles.

Figura 6.3: Modelo de cámaras empleado para capturar las imágenes de los patrones sobre el objeto

Proyector Hitachi CP-X308

Este proyector se emplea para generar los patrones sobre el objeto. El proyector es ajustado antes de comenzar el proceso de captura, mediante un proceso de calibración. Este modelo tiene una resolución de 1024 puntos x 768 líneas. Este dato es importante ya que, la resolución de la cámara conviene que sea mayor que la del proyector, para así ser capaces de captar la pantalla completa, no perder patrones y poder reconstruir de forma correcta.

Figura 6.4: Proyector empleado para emitir los patrones

Camara Intel Real Sense ZR300

Esta camara se emplea para realizar un análisis comparativo de la reconstrucción obtenida, para hacernos una idea de la precisión y calidad del sistema que se ha diseñado.

Su funcionamiento también se basa en la emisión de una serie de patrones de luz infrarroja, para obtener una imágen de profundidad, que junto con la imagen de color, se emplean para hallar la digitalización 3D de la escena que se esté capturando.

En la figura 6.5, se aprecian todas las cámaras de las que consta el modelo usado.

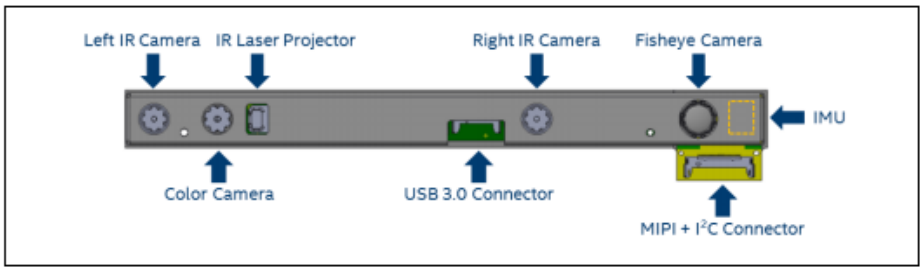


Figura 6.5: Estructura interna de la camara Real Sense ZR300

6.3. Captura de Imágenes

Para la adquisición de las imágenes es necesario ajustar tanto las cámaras como el proyector de forma que no exista movimiento alguno, ya que de esta forma se descalibraría el sistema, causando un error en la reconstrucción del objeto. Algo muy importante a tener en cuenta es la iluminación de la escena. Esta puede causar problemas adicionales, como la identificación de sombras que realmente no formarían parte de los patrones.

6.3.1. Captura de imágenes para la calibración

La calibración de las cámaras y el proyector es un paso crucial para poder obtener el modelo 3D correcto. Por este motivo, se toman una serie de imágenes para llevar a cabo la calibración. Teniendo todos los elementos fijados, se toman un número suficiente de imágenes (entre 10 y 20) a un tablero de ajedrez. Estas imágenes se toman moviendo el tablero y poniéndolo en diferentes orientaciones. Por otra parte, una última imagen es tomada con las 2 cámaras a la vez, con el tablero situado en una posición fija. Esto se hace para obtener los parámetros extrínsecos.

Para cada posición y orientación de la tabla de calibración, las imágenes se toman desde ambas cámaras, para poder tener un punto de referencia común.

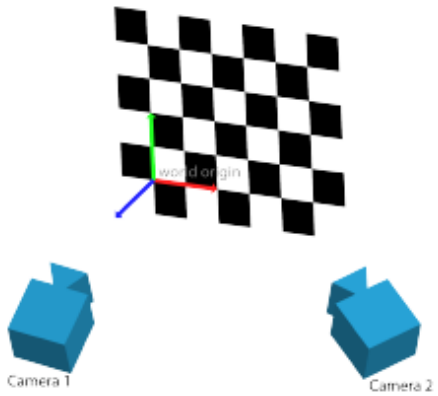


Figura 6.6: Calibración de las cámaras

6.3.2. Captura de las imágenes del objeto al proyectar patrones

Las imágenes se capturan de manera simultánea con ambas cámaras al proyectar cada patrón de imagen. En este paso es importante destacar que mediante programación ha sido necesario aplicar un pequeño delay extra, entre cada cambio de patrón, para que hubiera tiempo suficiente para procesar la captura de la imagen y así no se mezclasen 2 capturas con 2 patrones diferentes.

Tipo de patrones seleccionados para proyectarlos sobre el objeto

El primer paso del proceso es la codificación de la información en patrones en el dominio temporal. Esto implica la generación de patrones codificados que, cuando se proyectan en secuencia sobre el objeto, permitan la identificación única de cada uno de los píxeles del objeto. Hay varios esquemas para codificar información en secuencias de patrones pero los más populares son el código binario y el código de Gray.

En nuestro proyecto, los patrones empleados siguen un código de Gray. Este tipo de código asegura que solo haya un bit de diferencia entre patrones consecutivos. Estos patrones, se pueden calcular a partir de los datos binarios, simplemente copiando el bit más significativo y calculando para el resto de bits, la XOR del bit actual con el anterior de mayor peso. La codificación gray ha sido considerada la opción mejor.

6.4. Proceso de calibración

Una parte esencial de las tecnologías 3D es la calibración de las cámaras y del proyector, que es esencial para establecer la precisión del sistema 3D.
La imagen de una cámara se forma cuando los rayos de luz que se reflejan en la escena, pasan a través de la lente (a través de una apertura) e interactuan con el sensor de luz. Si no existiera ese sensor de luz, los rayos convergirían en un solo punto llamado centro de proyección, como se muestra en la Figura 6.7.

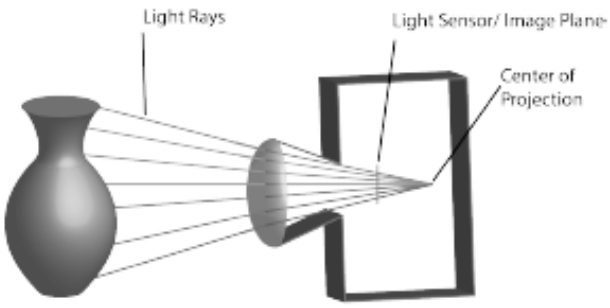


Figura 6.7: Formación de la imágen

Un proyector se puede considerar como una cámara invertida en la que los rayos de luz parten de un punto e interactuan posteriormente con la escena.

En ambos casos (cámara y proyector), podemos relacionar el rayo de luz con un píxel particular si se conoce la geometría del sistema en el momento en que se tomó la imagen. La geometría está definida por un conjunto de parámetros intrínsecos y extrínsecos que se obtienen mediante una calibración geométrica.

Dado que la mayoría de los sistemas de imágenes 3D utilizan sensores ópticos 2D, la calibración de la cámara establece los procedimientos para relacionar un píxel de una imagen 2D (en coordenadas de cámara) y una línea recta en el espacio 3D (coordenadas del mundo real). A lo largo de dicha línea se ubica el punto objeto.

Para calibrar las cámaras, se utiliza un objeto con características geométricas conocidas. En nuestro caso se ha utilizado un tablero de ajedrez, como el mostrado en la Imagen 6.8.

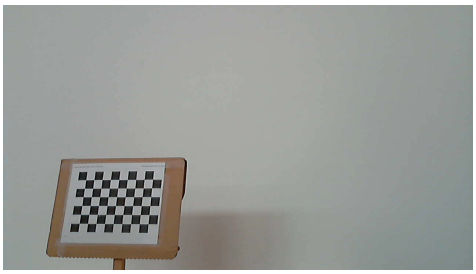


Figura 6.8: Tablero de ajedrez empleado para la calibración

Tomando imágenes del tablero en diferentes posiciones y orientaciones, podemos calcular tanto los parámetros intrínsecos como los extrínsecos. En la siguiente ecuación, se pueden observar la matriz de cámara C, que incluye todos los parámetros de calibración que afecten linealmente al sistema. Otros parámetros, como la distorsión de las lentes, no han sido introducidos en dicha ecuación.

$$C = \begin{bmatrix} \alpha & -\alpha\cot(\theta) & u_0 \\ 0 & \frac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

El primer término se refiere a los parámetros intrínsecos (parámetros de la cámara), mientras que el segundo se refiere a los parámetros extrínsecos (parámetros dependientes de la posición de la cámara).

En dicha matriz, $\alpha = kf_x$, $\beta = kf_y$, siendo f_x y f_y las distancias focales en el eje x e y respectivamente. θ es el ángulo de sesgo, u_0 y v_0 se refieren al punto principal en los ejes x e y respectivamente.Por otra parte, r y t, determinan la rotación y la traslación de la cámara en relación con el mundo.

Entre los parámetros intrínsecos se incluyen también los coeficientes de distorsión: tres coeficientes para la distorsión radial y dos para la distorsión tangencial. Aunque, en teoría, la matriz de cámara C se puede calcular a partir de una sola imagen, es recomendable tomar entre 10 y 20 imágenes de la tabla de calibración en diferentes orientaciones y posiciones, con objeto de minimizar el error en el cálculo.

Para calcular los parámetros extrínsecos de cada cámara, se deben haber calculado ya los intrínsecos. Empleando una imagen, donde se especifique el origen del sistema de referencia, se calculan la rotación y la traslación. Para lograr esto todas las cámaras en el sistema deben capturar la escena al mismo tiempo, para tomar la misma posición y orientación del patrón de calibración. De esta manera, se puede utilizar el mismo origen de coordenadas para relacionar el movimiento entre las cámaras, es decir, la rotación y la traslación.

Por otra parte, se va a emplear la Toolbox de Matlab, para realizar la misma calibración. Los resultados, tras el proceso de calibración, resultan muy similares a los obtenidos anteriormente. Pero la toolbox proporciona también un esquema para visualizar la posición tanto de las cámaras como del tablero de ajedrez en las diferentes posiciones, en las que ha sido capturado, como se visualiza en la figura 6.9.

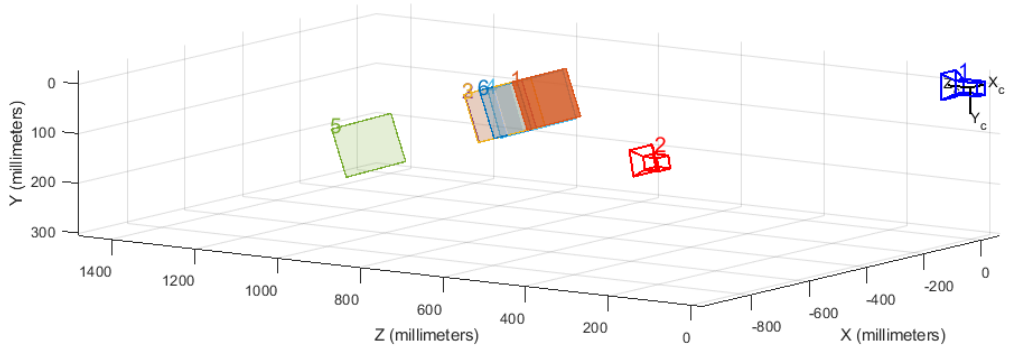


Figura 6.9: Posicionamiento de las cámaras a partir de la calibración con Matlab

6.5. Obtención del modelo 3D del objeto

6.5.1. Decodificación de las imágenes capturadas

Identificación de las regiones de sombra

Debido a que las cámaras observan al objeto desde diferentes ángulos que el proyector, habrá zonas de sombra que es conveniente no tener en cuenta a la hora de llevar a cabo la reconstrucción. Esto se puede conseguir proyectando una imagen blanca y otra negra sobre el objeto. Los píxeles sombreados son aquellos en que la diferencia de intensidades en las 2 imágenes capturadas sean menor que un determinado umbral, que se definirá en función de las condiciones de iluminación del ambiente donde se esté realizando la captura . Un ejemplo de la extracción de las sombras de un objeto se puede observar a continuación, en la Figure 6.10. Se puede observar como las zonas representadas en negro son las que pertenecen a una región de sombra y por tanto no se tienen en cuenta a la hora de realizar la reconstrucción.

Figura 6.10: Ejemplo de la extracción de las sombras de un objeto

Decodificación de los patrones

Existen varías estrategias para reconstruir la imagen a partir de la luz reflejada por el objeto. Un buen principio es añadir marcas de profundidad en los patrones de rayas adquiridos, teniendo en cuenta que el desplazamiento de cualquier raya se puede convertir directamente en una coordenada 3D. Para este propósito, la identificación del patrón debe hacerse individualmente, por ejemplo, empleando un conteo de las rayas (método por reconocimiento de patrones). Otro método se basa en proyectar la mitad del patrón en color blanco y la otra mitad en negro, resultando una secuencia binaria en código de Gray y asignar un valor a cada muestra codificada. De este modo se consigue depurar la información de profundidad relativa del píxel respecto sus vecinos. Una vez adquirida la profundidad de todos los píxeles se reconstruye el objeto tridimensional.

Este es el método empleado en nuestro proyecto.

El proceso de decodificación de los patrones proyectados es necesario para poder asociar los píxeles de las imágenes capturadas por las cámaras con los mismos píxeles del proyector. Tal y como se ha explicado anteriormente, los patrones emitidos van siguiendo una secuencia gray. El proceso de decodificación sigue los pasos que se comentarán a continuación. En primer lugar cabe destacar que para cada píxel se va a obtener un valor decimal con coordenada x (valor decimal de la decodificación de los patrones horizontales) y con coordenada y (valor decimal de la decodificación de los patrones verticales). Se debe comentar que se han generado 10 patrones horizontales y 10 patrones verticales, por tanto se van a hacer agrupaciones en parejas de patrones. Para cada píxel se va a comprobar si la diferencia de intensidad entre el mismo píxel de ambas imágenes se encuentra en un rango válido, caso en el que la diferencia es menor que un determinado umbral. El resultado para cada píxel serán 10 bits 0 ó 1, dependiendo si el píxel se encuentra en el rango correcto o no. Posteriormente lo único que se debe hacer es pasar esos 10 bits a su valor decimal. Este proceso se debe repetir para todos los píxeles de las 2 cámaras, tanto con los patrones horizontales (lo que proporciona Xdec) como con los verticales (valor Ydec). Por tanto, para cada píxel de cada cámara, se tendrá una decodificación Xdec,Ydec.

En la Figura 6.11 se representa 1 imagen en la que se puede apreciar el sistema empleado para decodificar cada punto de la imagen.

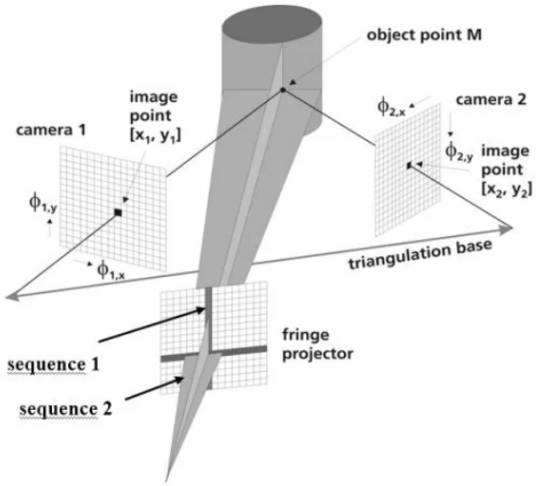


Figura 6.11: Mapeado entre píxeles de las cámaras y del proyector

Y a continuación, se adjunta el pseudo-código de la función decodePaterns(), para que de esta forma se pueda entender mejor su funcionamiento.

Pseudocódigo de la función decodePaterns()

```
for (int i=0; i<w; i++)
    for (int j=0; j<h; j++)
        if(pixel≠sombra)
            getProjPixel(i,j,projPixel)
            for(int count=0; count<numOfColBits; count++)
                if(abs(val1-val2)<whiteThreshold){
                    if(val1>val2)
                    grayCol←1
                    else
                    grayCol←0
                    end if
                end if
                xDec=grayToDec(grayCol)
            end
            for(int count=0;count<numOfRowBits;count++)
                if(abs(val1-val2)<whiteThreshold){
                    if(val1>val2)
                    grayRow←1
                    else
                    grayRow←0
                    end if
                end if
                yDec=grayToDec(grayRow)
            end
            projPixel.x=xDec;
            projPixel.y=yDec;
            camPixels[projPixel.x*proj_h+projPixel.y]
        end if
    end
end
```

Tras decodificar las imágenes, tenemos un conjunto de relaciones entre los píxeles de ambas cámaras. El siguiente paso es triangular los rayos correspondientes a cada par de imágenes, para obtener de esta forma el punto de intersección. La proyección de este punto cae sobre los píxeles mapeados en las 2 cámaras.

6.5.2. Reconstrucción

Transformación de píxel a rayo

Un rayo de una cámara, es una línea en el espacio que empieza en el centro de proyección de la cámara. Este rayo, se puede definir con un punto del rayo y un vector de dirección.
Para obtener el vector dirección, son necesarios 2 puntos: el primero, es el (0,0,0) ó centro de proyección de la cámara y el segundo punto es el correspondiente al píxel por el que pasa el rayo. Este punto se puede obtener mediante la siguiente expresión:
$q_{pixel}^{camara} = C^{-1} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, donde C es la matriz de la cámara.

Posteriormente se debe cambiar el sistema de referencia de los puntos(pasando de las coordenadas de las cámaras a coordenadas del mundo físico). Para ello se emplean los parámetros extrínsecos obtenidos mediante la calibración de las cámaras.

$q = R^{-1} \times q^{camara} - R^{-1} \times T$

siendo R la matriz de rotación y T el vector de transalación.

Por tanto, el rayo correspondiente al píxel p está definido de la siguiente forma:

$R_p =< p_{center}, v_{dir} >, conv_{dir} = p_{pixel} - p_{centro}$

Triangulación

El siguiente paso es la triangulación de los puntos. Dado un píxel en la imagen correspondiente a la cámara 1 y su píxel correspondiente en la imagen de la cámara 2, se forman los rayos para ambos y se debe calcular la intersección.
En ocasiones, los rayos no intersecan exactamente y pasan muy cerca. Para solventar esta situación, se traza el segmento perpendicular a ambos, cuya distancia sea la menor posible y el punto medio del segmento será el considerado como punto de intersección.
En las imágenes 6.12 y 6.13 se representan 2 esquemas del proceso de triangulación.

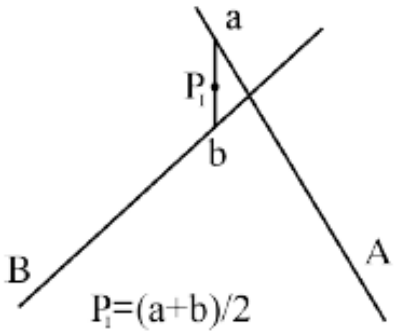


Figura 6.12: Triangulación de los rayos

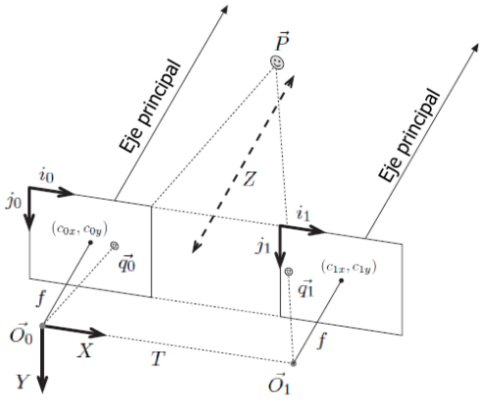


Figura 6.13: Triangulación de los rayos

A continuación, se detallará el cálculo del punto de intersección de los rayos utilizando ecuaciones.

Consideramos los rayos A y B pasando por los puntos p y q y cuyos vectores de dirección son: $\vec{u}$ y $\vec{v}$.

$a = p + s * \vec{u}$
$b = q + t * \vec{v}$,
s y t son valores escalares.

El segmento que conecta los puntos a y b es perpendicular a los rayos y por tanto el producto escalar de sus vectores es 0.

$(a - b).\vec{u} = 0$ $(a - b).\vec{v} = 0$

Juntamos las 4 ecuaciones anteriores, para así poder hallar s y t, se obtiene:

$\vec{w}.\vec{u} + s * \vec{u} * \vec{u} - t * \vec{v}.\vec{u} = 0$

$\vec{w}.\vec{v} + s * \vec{v} * \vec{u} - t * \vec{v}.\vec{v} = 0$
$\vec{w} = p - q$

$s = \frac{\vec{w}.\vec{u} * \vec{v}.\vec{v} - * \vec{v}.\vec{u} * \vec{w}.\vec{v}}{\vec{v}.\vec{u} * \vec{v}.\vec{u} - \vec{v}.\vec{v} * \vec{u}.\vec{u}}$

$t = \frac{\vec{v}.\vec{u} * \vec{w}.\vec{u} - * \vec{u}.\vec{u} * \vec{w}.\vec{v}}{\vec{v}.\vec{u} * \vec{v}.\vec{u} - \vec{v}.\vec{v} * \vec{u}.\vec{u}}$

Habiendo sacado los puntos s y t, el punto de intersección se calcula de manera sencilla, siendo este el punto medio del segmento que forma s y t:

$$P_i = \frac{(p + s * \vec{u}) + (q + t * \vec{v})}{2}$$

Para todo par de rayos, se calcula el punto de intersección de esta forma, evitándose cualquier problema que pueda surgir ante un par de rayos que no intersequen.

6.5.3. Conversión de nube de puntos a malla 3D

Una vez obtenidos todos los puntos, representando la escena escaneada, se debe generar una malla en 3 dimensiones donde los puntos (vértices) se interconectan para dar lugar a arcos y a caras.

Existen muchos métodos con diversas complejidades. En este caso se describirá un método de conversión rápida, que aprovecha la ventaja de la relación espacial entre los puntos 3D y los píxeles del proyector.

Se consideran vecinos de un píxel p a los 8 píxeles circundantes, donde los colocados en la misma fila o la columna con p son vecinos de primer nivel, mientras que las diagonales son vecinos de segundo nivel, como puede verse en la Figura 6.14.

2nd	1st	2nd
1st	p	1st
2nd	1st	2nd

Figura 6.14: Píxeles vecinos

Los puntos de los píxeles vecinos están conectados iterativamente entre sí formando caras, estas pueden ser triangulares o en forma de cuadriláteros. Para formar caras triangulares, los píxeles vecinos de 1er y 2do nivel están interconectados, en cambio las caras en forma de cuadrilátero solo requieren conexiones vecinas de primer nivel, como se aprecia en la figura 6.15. Las mallas con caras triangulares tienen mayor complejidad que las cuadriláteras.

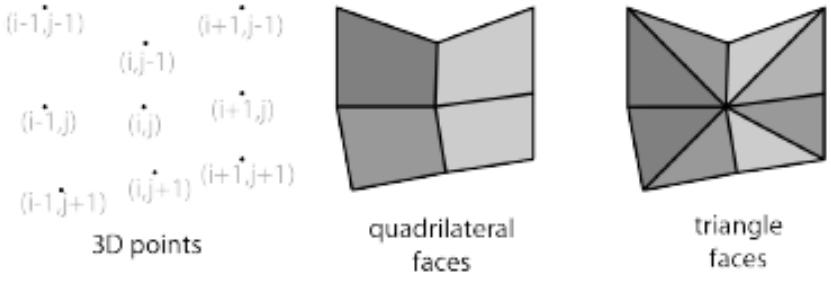


Figura 6.15: Formación de las caras de la malla a partir de los puntos de intersección

El método descrito puede introducir agujeros en la malla de salida en los casos en que algunos píxeles no sean visibles por ambas cámaras.Este método es genérico y bastante rápido, ya que, se realiza en el espacio de imagen del proyector y funciona muy bien para la mayor parte del área de la escena.

6.6. Explicación del programa completo y su estructura

A continuación se observa el esquema explicativo de todo el proceso que se lleva a cabo para realizar las digitalizaciones 3D. El programa consta de aproximadamente 2000 líneas de código, escrito en lenguaje C++ y en el que se emplean numerosas funciones ya definidas en la biblioteca de OpenCV, que facilitan mucho el desarrollo del programa.

Estructura completa del programa

1. Posicionar las cámaras y el proyector en una posición fija
2. Capturar las imágenes de calibración y las imágenes al proyectar los patrones **captureCalibrationImagesAndScan()**
3. Proceso de calibración **calibration()**
4. Reconstrucción **reconstruct()**

En caso de que no sea necesaria la calibración de las cámaras porque esta ya ha sido realizada anteriormente, únicamente se capturan las imágenes de los patrones y se reconstruye.

1. captureCalibrationImagesAndScan()

Esta función se va a emplear para la captura de todas las imágenes necesarias para llevar a cabo la digitalización del objeto. Este proceso de captura se divide en 2, el de captura de las imágenes para la calibración del sistema y el proceso de captura de los patrones. A continuación, se puede observar el pseudo-codigo de la función.

1.1. captura de las imágenes para la calibración
- **1.1.1.** captura de imágenes para hallar parámetros intrínsecos (para cada cámara)
  - **1.1.1.1.1.** 10 imágenes con el tablero en diferentes posiciones y orientaciones
- **1.1.2.** captura de imágenes para hallar parámetros extrínsecos
  - **1.1.2.1.1.** 1 imagen con las 2 cámaras simultáneamente, para tener el mismo punto de referencia y situar espacialmente ambas cámaras

1.2. captura de patrones
- **1.2.1.1.** generación de los patrones
  - **1.2.1.1.1.1.** 1 imagen blanca
  - **1.2.1.1.1.2.** 1 imagen negra
  - **1.2.1.1.1.3.** 20 patrones verticales
  - **1.2.1.1.1.4.** 20 patrones horizontales
- **1.2.1.2.** Proyección de cada una de las 42 imágenes de patrón generadas
  - **1.2.1.2.1.1.** Captura de la imagen del objeto al proyectar el patrón i con la cámara 1
  - **1.2.1.2.1.2.** Captura de la imagen del objeto al proyectar el patrón i con la cámara 2

2. calibration()

Esta función se va a encargar de obtener los parámetros de calibración del sistema, estos son necesarios para posteriormente llevar a cabo la reconstrucción. Este proceso es el empleado normalmente para calibración de sistemas estéreo. A continuación, se explica el proceso completo mediante pseudo-codigo, en el que por un lado se obtienen los parámetros intrínsecos y por otro, los extrínsecos.

2.1. Para cada cámara i
- **2.1.1.1.** Cargar las imágenes de calibración para la cámara i
- **2.1.1.2.** extractImageCorners()
  - **1.1.1.1.1.1.** Para cada imagen de los patrones con la cámara i
    - **1.1.1.1.1.1.1.1.** findCornersInCamImg()

    +mientras no se hayan encontrados las 4 esquinas{

    -marcar las 4 esquinas en el tablero

    -marcar un punto como origen, en el tablero

    -indicar numero de cuadrados en los ejes x e y

    -findChessboardCorners()→ *función OpenCV para hallar las esquinas*, te devuelve camCorners

-drawChessboardCorners()→ *función OpenCV para dibujar líneas sobre las esquinas del tablero*

}

+encontrar subpíxeles

-dar la anchura y altura de cada cuadrado del tablero, para establecer relación con la medida real

+objCorners

2.1.1.3. calibrateCamera, pasando camCorners y objCorners

1.1.1.1.1.1. calibrateCamera()→función OpenCV que nos devuelve camMatrix, distortion, camRotationVectors, camTranslationVectors

2.1.1.4. findCameraExtrisics

1.1.1.1.1.1. findCornersInCamImg()

misma función que en el caso anterior, con la salvedad de que en este caso se usa la imagen capturada para obtener los parámetros extrínsecos de calibración

+mientras no se hayan encontrados las 4 esquinas{
-marcar las 4 esquinas en el tablero
-marcar un punto como origen, en el tablero
-indicar número de cuadrados en los ejes x e y
-findChessboardCorners() función OpenCV para hallar las esquinas, te devuelve camCorners
-drawChessboardCorners() función OpenCV para dibujar líneas sobre las esquinas del tablero
}
+encontrar subpíxeles
-dar la anchura y altura de cada cuadrado del tablero, para establecer relación con la medida real

+objCorners

1.1.1.1.1.2. encontrar parámetros de rotación y translación con las siguientes funciones OpenCV

1.1.1.1.1.1.1.1.1. solvePnP(objPoints3D,imgPoints,camMatrix,distortion,rVec,translationVector);

1.1.1.1.1.1.1.1.2. Rodrigues(rVec,rotationMatrix);

2.1.1.5. Exportar los ficheros con los parámetros de calibración

2.1.1.5.1.1. cam_matrix.txt

2.1.1.5.1.2. cam_distortion.txt

2.1.1.5.1.3. cam_rotation_matrix.txt

2.1.1.5.1.4. cam_trans_vectror.txt

3. reconstruct()

Esta última función es la más importante porque es la que realmente lleva a cabo la reconstrucción. Esta parte se puede dividir en otras 3. En primer lugar, se van a decodificar los patrones capturados, quitando zonas sombreadas. En segundo lugar, se va a obtener la reconstrucción de los puntos de profundidad, mediante un algoritmo de triangulación. Por último, una vez obtenida la nube de puntos se va a construir la malla 3D formada por los vértices y las caras. A continuación, se adjunta un pseudo-codigo, que sirve para explicar la función.

1.1. loadCameras()

1.1.1.1. para cámara i, se cargan los 4 ficheros con los parámetros obtenidos en la calibración de dicha cámara i

1.2. se establecen los parámetros de la reconstrucción

1.2.1.1. BlackThreshold,WhiteThreshold

1.3. runReconstruction()

1.1.1.1. Para cada cámara i

1.1.1.1.1.1. loadCamImgs()

1.1.1.1.1.1.1.1. Se cargan las imágenes de los patrones para la cámara i

1.1.1.1.1.2. computeShadows()

1.1.1.1.1.1.1.2. Para cada píxel de las imágenes 1 y 2 de los patrones, correspondientes a los patrones blanco y negro

-if (whiteVal-BlackVal)>blackThreshold{

mask(i,j)=1→píxel no sombra

}

-else{

mask(i,j)=0→píxel de sombra

}

1.1.1.1.1.3. decodePaterns()

1.1.1.1.1.1.1.3. Para cada píxel de las 40 imágenes de los patrones verticales y horizontales, que no corresponda a un píxel de sombra

1.1.1.1.1.1.1.1. getProjPixel(i,j,projPixel)

1.1.1.1.1.1.1.1. Patrones verticales (para cada par de imágenes de patrones consecutivas, cogidas por parejas, se obtienen val1 y val 2

-if(abs(val1-val2)<whiteThreshold){

If(val1>val2){

grayCol←1

}

else{

grayCol←0

}

}

xDec=grayToDec(grayCol)

1.1.1.1.1.1.1.1.1.1. Patrones horizontales(para cada par de imágenes de patrones consecutivas, cogidas por parejas, se obtienen val1 y val 2

-if(abs(val1-val2)<whiteThreshold){

If(val1>val2){

grayRow←1

}

else{

grayRow←0

}

}

yDec=grayToDec(grayRow)

projPixel.x=xDec;

projPixel.y=yDec;

1.1.1.1.1.1.1.1.2.

1.1.1.1.1.1.1.1.2. camPixels[projPixel.x*proj_h+projPixel.y]

1.1.1.2. triangulation(camsPixels[0],cameras[0],camsPixels[1],cameras[1])

1.1.1.1.1. Para cada píxel del proyector

1.1.1.1.1.1.1.1. Obtener los rayos correspondientes a cada cámara, que pasan por el píxel actual

1.1.1.1.1.1.1.2. Calcular el punto de intersección de ambos rayos

1.4. computeMesh("projector_view.obj")

out1.open("projector_view.obj")

1.1.1.1. Para cada píxel del proyector

1.1.1.1.1.1. Escribir los puntos de intersección, anteriormente obtenidos, como vértices de la reconstrucción final

out1<<"v "<< point.x<< " "<< point.y<< " "<<point.z<< "\n";

1.1.1.1.1.2. Escribir caras de la reconstrucción

1.1.1.1.1.1.1.1.1. Cara triangular formada por vértices: píxel(i,j)→píxel((i+1),j)→píxel(i,(j+1)

1.1.1.1.1.1.1.1.2. Cara triangular formada por vértices: píxel(i,j)→píxel((i+1),j)→píxel((i+1),j)

CAPÍTULO 7

Resultados obtenidos

En este apartado, se presenta la reconstrucción final de la escena. Se adjuntan tanto la imagen de la escena, como su digitalización 3D, en la que se puede observar como el sistema reconstruye la escena perfectamente, siendo este ya un resultado consistente y el que se podría esperar para nuestro proyecto.

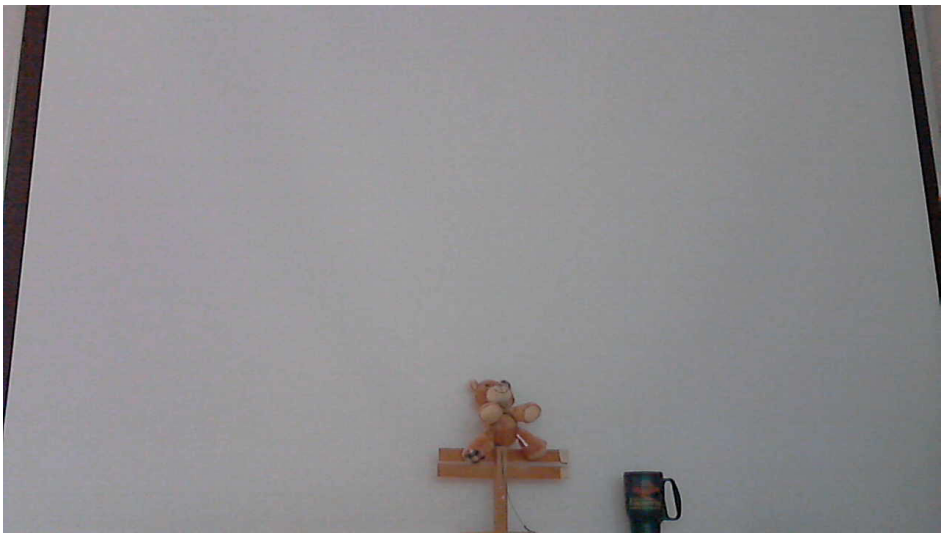


Figura 7.1: Fotografía de la escena a reconstruir

En las figuras 7.2 y 7.3, se aprecia perfectamente la forma del objeto. Cabe destacar que la escena está compuesta por un oso de peluche subido sobre una estructura de madera formada por 2 tablillas. Además, a un lado, se sitúa una taza, que se distingue perfectamente y se aprecian las dimensiones, correctamente escaladas.

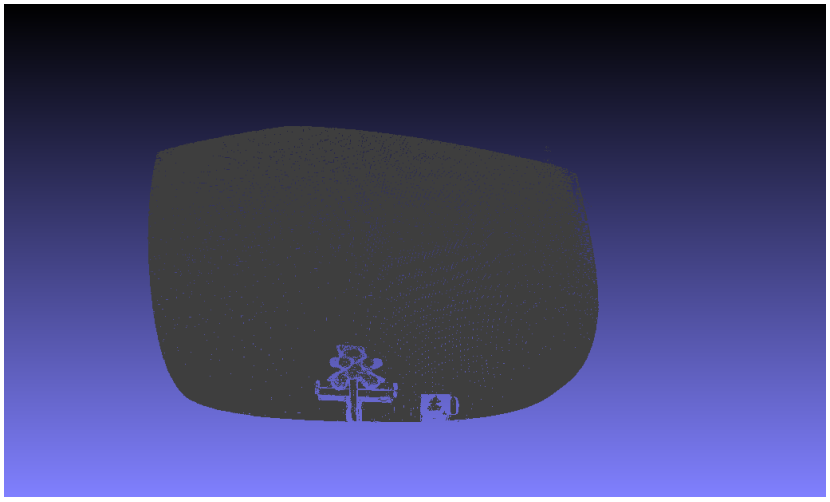


Figura 7.2: Reconstrucción 3D de la escena

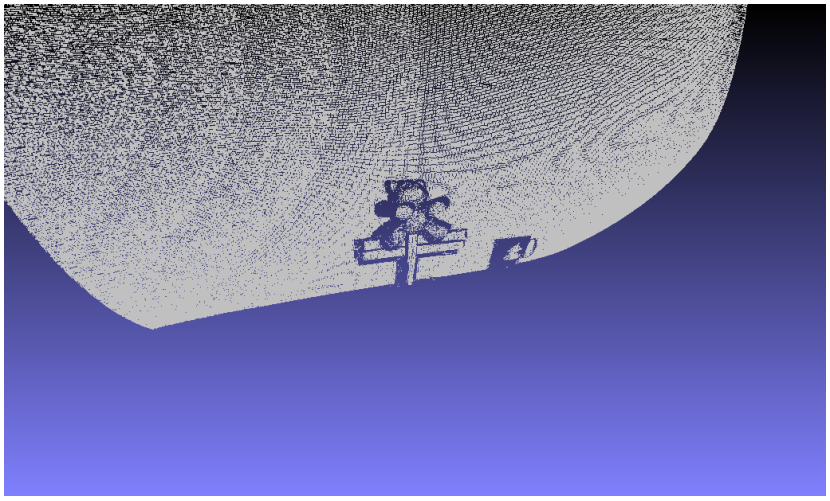


Figura 7.3: Reconstrucción 3D de la escena

Además en la figura 7.4 se observa un zoom sobre el objeto en el que se aprecia mejor la creación del modelo 3D y la sensación de profundidad.

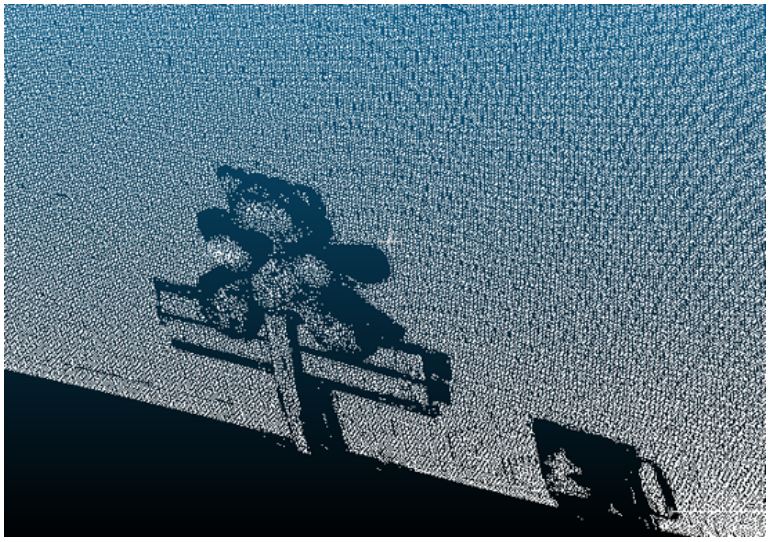


Figura 7.4: Zoom sobre los puntos de profundidad de la escena

En las siguientes 2 imágenes, se adjunta la escena al ser iluminada con los patrones. A partir de todas las imágenes capturadas al proyectar los patrones se ha obtenido el modelo 3D visto en la figura 7.2.

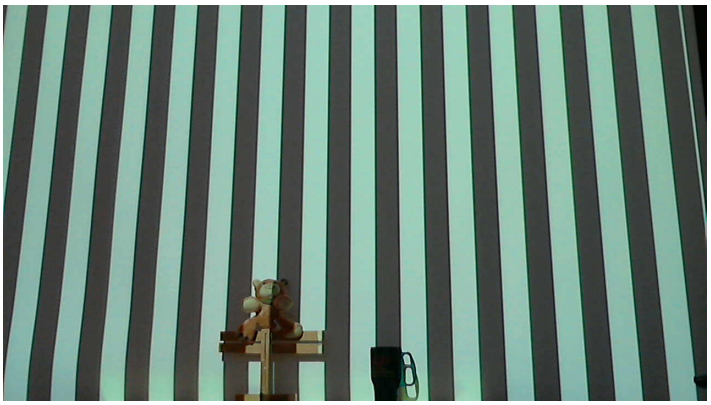


Figura 7.5: Emisión de patrones verticales

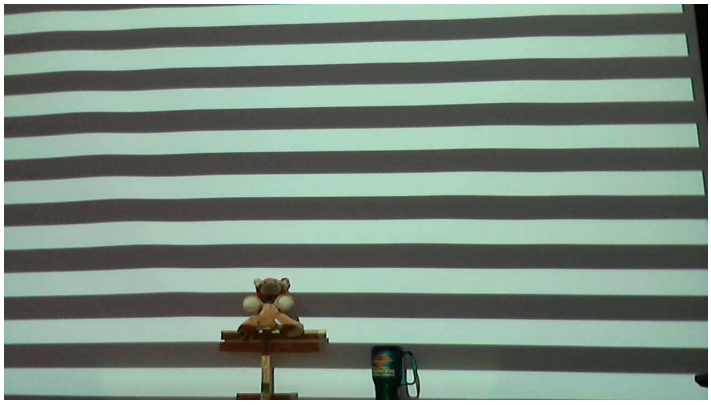


Figura 7.6: Emisión de patrones horizontales

7.1. Comparativa de resultados con respecto a la cámara Intel RealSense ZR300

Como último paso en el proyecto, se va a tratar de evaluar el rendimiento que nos puede llegar a dar el sistema que hemos diseñado. Para realizar este cálculo, se va a realizar una comparativa de error con respecto a la cámara Real Sense ZR300, diseñada por Intel, modelo que se puede observar en la figura 7.7.

Figura 7.7: Camara Real Sense ZR300

Para poder calcular el error en la reconstrucción, se va a reconstruir una superficie plana cuya profundidad es constante, respecto al plano en el que se sitúan las cámaras. Los 2 modelos 3D de dicha superficie, se pueden observar a continuación, en la figura 7.8. El modelo de la izquierda corresponde al obtenido con la cámara Real Sense y el de la derecha corresponde al obtenido con el sistema de luz estructurada, diseñado en el proyecto.

Para hallar el error en la reconstrucción, se va a calcular la medida de profundidad del plano, para ambos sistemas. Para poder realizar esta medida, es necesario tener una serie de cosas en cuenta. En primer lugar, de la reconstrucción se escoge únicamente aquellos puntos que pertenezcan al plano que interesa (este plano es el paralelo al plano en el que sitúan tanto las cámaras del sistema de luz estructurada como la Real Sense).

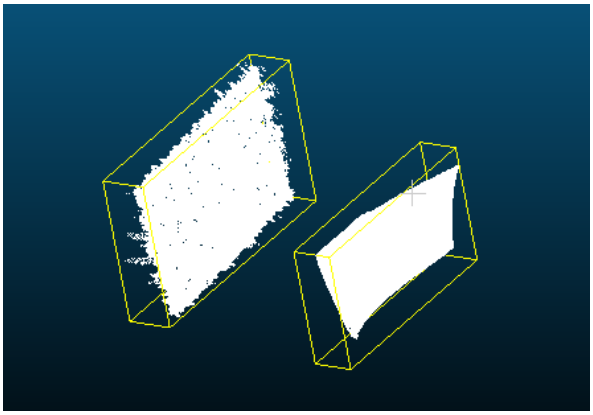


Figura 7.8: Reconstrucción empleada para realizar una comparativa entre los 2 sistemas

En segundo lugar, se van a desechar aquellos puntos que la medida de profundidad sea 0 y posteriormente se va a realizar una medida de la media de profundidad para ambas reconstrucciones. Tras esto se calcula el error de medida de nuestro sistema con respecto a la medida tomada con la cámara Real Sense empleada.

Los resultados, son los siguientes:

Medidas	Sistema diseñado con luz estructurada	Real Sense ZR300
Profundidad media	2.5627 m	2.6691 m

Cuadro 7.1: Resultados de la comparativa entre los 2 sistemas, para obtener modelos 3D

$$d = |r - s| = |2{,}6691 - 2{,}5627| = 0{,}1064 \text{ m},$$

r=profundidad media de los puntos tomados con la cámara Real Sense
s=profundidad media de los puntos tomados con el sistema diseñado
d=diferencia entre ambas medidas

$$error(\,\%) = d/r = 0{,}1064/2{,}6691 = 3{,}9854\,\%$$

Este resultado puede ser interpretado, de forma que la cámara diseñada por Intel nos proporciona unos resultados bastante exactos en comparación con nuestro sistema, principalmente porque nuestro sistema está formado por 2 cámaras que deben ser calibradas y en ese proceso se introduce un determinado error de posicionamiento.

Por otra parte, existen una serie de factores que también introducen errores, como la calidad de las cámaras, que hacen que a la hora de triangular los puntos, la medida no sea tan exacta, como lo puede ser las cámaras que lleva incorporadas el modelo

de Intel. Por otra parte, la iluminación hace que haya regiones de sombra, que no lo deberían ser y esto puede causar, que algunos puntos no se puedan reconstruir, al no ser correctamente reconocidos por ambas cámaras, esto hace que la medida de error también se vea incrementada.

7.2. Problemas a la hora de obtener resultados satisfactorios

En el apartado de resultados, se debe reseñar que antes de obtener una reconstrucción correcta se han realizado, numerosas pruebas. En el caso en que las cámaras no están correctamente colocadas, no son capaces de capturar el conjunto de todos los patrones emitidos por el proyector y entonces al tratar de decodificar los patrones, estos no son detectados. En la reconstrucción se observa que los bordes sufren dobles, debido a que se intentan recorrer todos los píxeles de las imágenes de los patrones, pero las correspondencias no son las correctas al no poder ser capturadas todas las líneas de los patrones .El resultado se ve reflejado en la Figura 7.9.

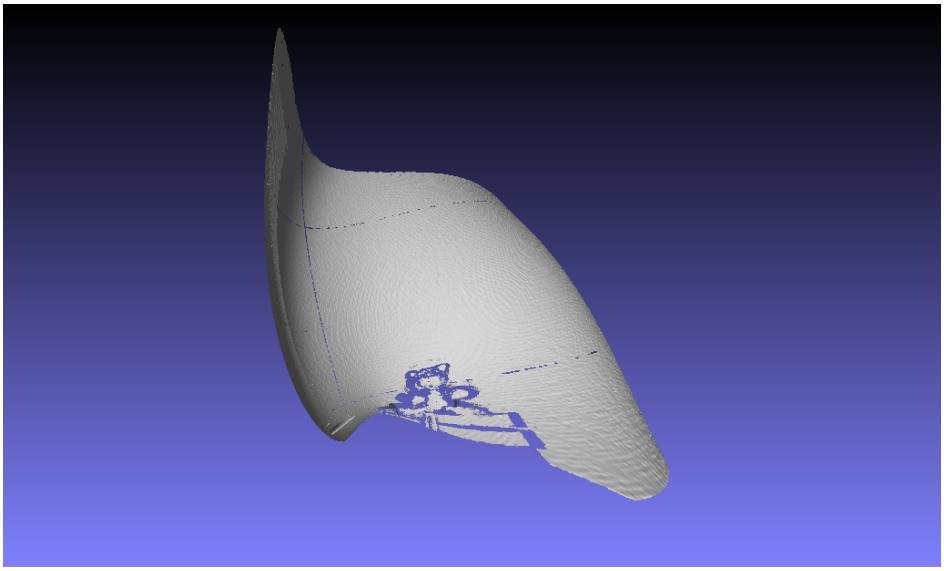
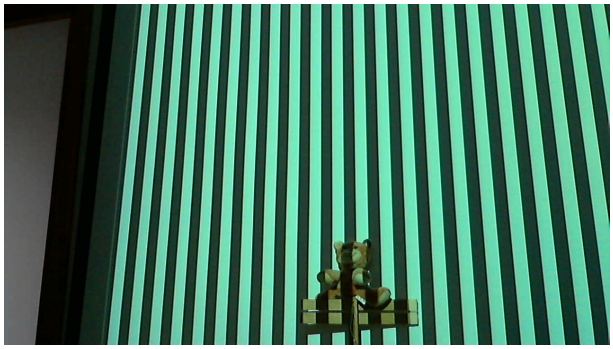
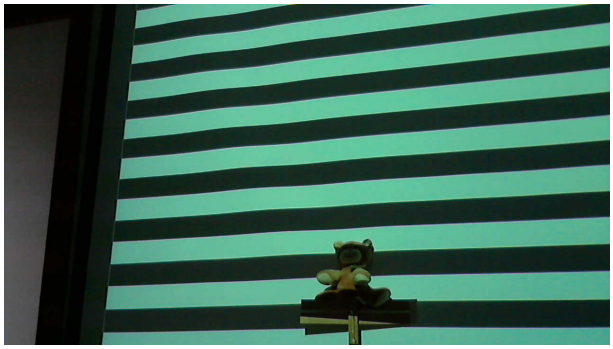


Figura 7.9: Reconstrucción de un objeto en el que las cámaras no captan todos los patrones

A continuación, se muestran un par de imágenes para que se vea como las capturas no cogen el total de las líneas de los patrones.



(a) Imágen en la que aparecen cortados patrones verticales por la derecha del objeto



(b) Imágen en la que aparecen cortados patrones horizontales por arriba y por abajo

Figura 7.10: Imágenes capturadas en las que no se aprecian el conjunto de todos los patrones

Ahora se van a mostrar los resultados obtenidos, tras una serie de mejoras en el código y con mejores ajustes del sistema, para obtener una mejor calibración. En la primera figura se representa la vista del objeto con ambas cámaras, mientras que en la figura 7.12 aparece la reconstrucción de dicho objeto. El objeto es un oso de peluche y como se ve, la forma exterior se reconstruye perfectamente. En cuanto a los detalles interiores (como las patas las orejas y la boca) como tienen una textura diferente se diferencian bien, mientras que el resto de partes, no se diferencian tan bien, debido a su textura. Cabe mencionar que en este caso la iluminación de la escena no era la adecuada y por tanto hay zonas del oso que han sido detectadas como sombra, cuando en realidad no lo son y por tanto dichos puntos no se han tenido en cuenta a la hora de reconstruir.

(a) Cámara 1 (b) Cámara 2

Figura 7.11: Vista del objeto por las cámara

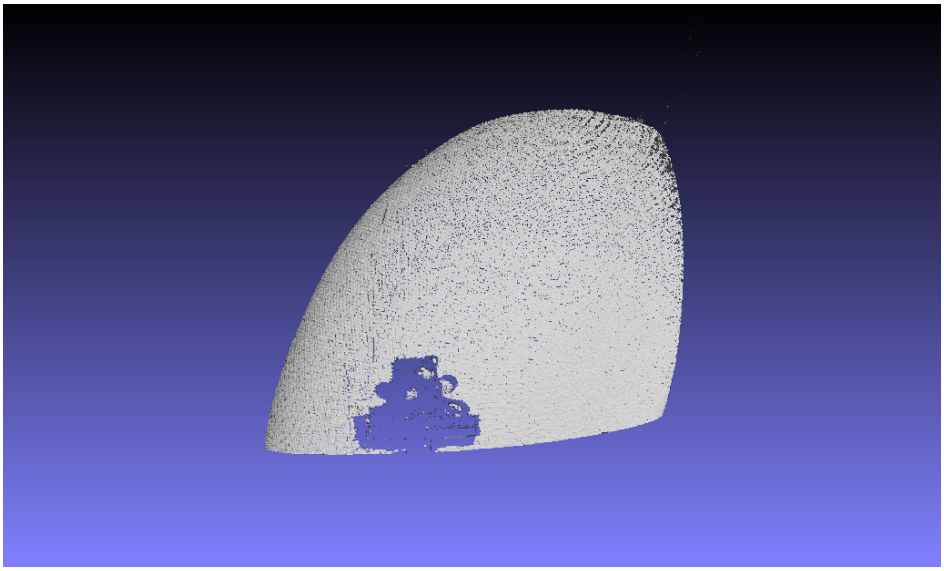


Figura 7.12: Vista 3D del objeto

CAPÍTULO 8

Conclusiones

Tras la realización del proyecto se han obtenido una serie de reconstrucciones, en las que la forma del objeto es distinguida con bastante precisión. Es por ello, que se ha llegado a la conclusión de que empleando una técnica conocida (luz estructurada) y un material bastante sencillo es posible obtener unos modelos bastante aceptables, para poder generar modelos 3D. Estos modelos posteriormente se pueden usar para diversas aplicaciones. Es por este motivo, que resulta tan útil un sistema que genere reconstrucciones 3D.

Por otra parte, se ha llegado a la conclusión de que para poder obtener unos resultados aceptables es necesario disponer de una serie de condiciones determinadas. Por ejemplo, la iluminación de la escena es completamente determinante, ya que, el sistema es muy sensible al exceso de luz y da lugar a reconstrucciones falsas. El motivo hallado para dar explicación a este hecho, es que a la hora de calcular los puntos de sombra, si tenemos demasiada luz en el ambiente, se identifican puntos que no deberán ser como sombras. Por tanto, se debe realizar la captura en un espacio poco iluminado.

Además en los algoritmos se tiene en cuenta otro umbral para comparar la intensidad de los patrones a la hora de realizar el proceso de decodificación, estos son de color negro y blanco, y en caso de existir mucha iluminación esto da lugar a una diferencia de intensidades distorsionada, provocando que se exceda el umbral y en consecuencia numerosos puntos se desechan, dando lugar a pérdida de información en el modelo final.

Además, la posición espacial de las cámaras también es determinante para poder obtener buenos resultados. Estas deben tener un ángulo de visión parecido, es decir, que estén capturando prácticamente la misma escena porque de lo contrario, habrá muchos puntos que no se puedan identificar por ambas. Esto daría como resultado huecos en la digitalización final, ya que, solo se reconstruyen aquellos puntos que sean identificados por las 2 cámaras.

8.1. Líneas de mejora

Algunas de las mejoras, que se podrían obtener, podrían ser:

- Inserción de una plataforma giratoria a motor, con el objetivo de poder tener una reconstrucción tridimensional completa del objeto, pudiendo solventar posibles oclusiones, que mediante el sistema usado, no pueden ser detectadas.

- Otra posible mejora podría ser emplear una Raspberry Pi, para correr el programa. Esto es posible, porque esta plataforma soporta OpenCV y además es muy sencillo de instalar. Se le podrían conectar las cámaras al puerto USB y mediante la salida VGA conectar el proyector. Por tanto, esto puedo ser una solución fácilmente implementable.

- Una línea de mejora, bastante interesante es la basada en un sistema de Moire (Figura 8.1), explicado en detalle en [18]. Este sistema diseñado, mejoraría la precisión de la reconstrucción. Para ello, emplea patrones no solo en el sistema proyectante sino que además, como novedad, introduce una serie de patrones en el receptor. Esto formaría los conocidos como patrones de Moire, que se pueden ver en la Figura 8.2.

  Este método, sin contacto y de alta resolución, se emplearía para medir desplazamientos fuera del plano, a través de una fórmula universal la cual expresa la relación existente entre la variación de fase y el desplazamiento fuera del plano.

  Para aumentar la precisión del sistema, es importante eliminar el error causado por la falta de coincidencia de los píxeles y el cambio del coeficiente de sensibilidad.

  El funcionamiento del algoritmo, se basa en el hecho de superponer dos rejillas de frecuencia espacial similar, de tal forma que, se proyectan unos patrones sobre el objeto, de la misma forma que se hace en nuestro proyecto. Pero además en esta aproximación, antes del receptor se coloca otra rejilla sobre la cuál se formaran los patrones de Moire (hecho observable en la figura 8.2), debidos a la deformación causada por el objeto en los patrones originales.

  La ventaja más notable de la topografía Moire es que puede magnificar la deformación sin distorsión, para lograr una alta resolución de medición.

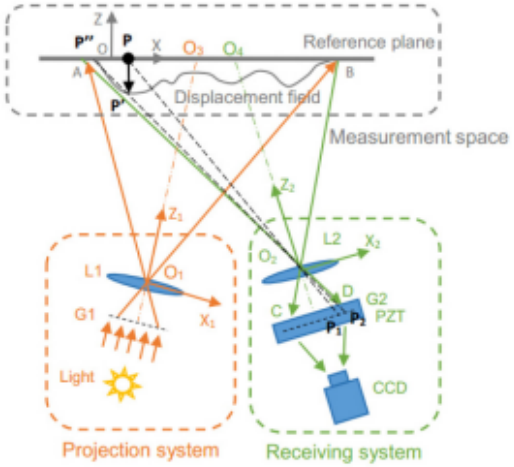


Figura 8.1: Esquemático general de un sistema Moire

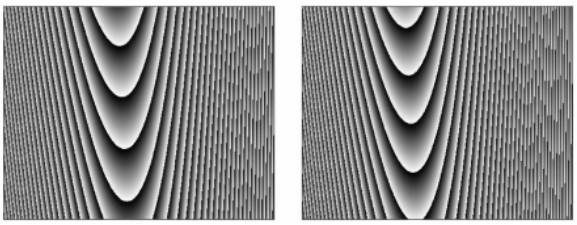


Figura 8.2: Patrones de Moire

Bibliografía

[1] Jason Geng, *Structured-light 3D surface imaging.* March 31, 2011

[2] https://www.opencv.org/

[3] Escáner 3D, https://es.wikipedia.org/wiki/Escáner_3D

[4] Prof. Didier Stricker, *3D Computer Vision.*

[5] Sven Jörissen, *3D Real-time Scanning Using a Projector-based Structured Light System.* September, 2017

[6] Rodrigues, Marcos, *Fast 3D Reconstruction using Structured Light Methods.* September, 2017

[7] Morillo Romero, Miguel Ángel , *Digitalización 3D con escáner de luz estructurada aplicada al área de la gestión de calidad y la conservación del patrimonio histórico-artístico.* 2015

[8] Structured light 3D scanning, https://www.instructables.com/id/Structured-Light-3D-Scanning/

[9] How to compare two 3D entities, http://www.cloudcompare.org/doc/wiki/index.php?title=How_to_compare_two_3D_entities

[10] Zhouyi Wu, Chao Han, Changuei Yang and Jiangtao Huangfu, *3D imaging scanner.*June 28, 2018

[11] Jean-Yves Bouguet,*Camera calibration toolbox for matlab.* 2004.

[12] Richard I. Hartley and Peter Sturm,*Triangulation.* August 24, 1996

[13] MeshLab, http://www.meshlab.net/

[14] Structured light, https://en.wikipedia.org/wiki/Structured_light

[15] Tyler Bell, Beiwen Li and Song Zhang,*Structured light techniques and applications.* August 24, 1996

[16] Intel RealSense 3D Camera ZR300,
https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/ZR300-Product-Datasheet-Public.pdf

[17] Ying Tang, Jun Yao and Jubing Chen, *Novel method for increasing accuracy of projection moiré contouring of large surfaces.* Septenber 5, 2016

[18] Structured-light 3D scanner,
http://www.opensourceimaging.org/project/structured-light-3d-scanner/