

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

REGISTRO DE UNA SECUENCIA TEMPORAL DE NUBES DE  
PUNTOS UTILIZANDO TECNOLOGIA KINECT PARA LA  
RECONSTRUCCION TRIDIMENSIONAL DE MATERIAL  
ARQUEOLOGICO

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

**Matias Alberto Quintana Rosales**

**ASESORES**

Benjamín Castañeda Aphán, Ph.D.

Ing. Fernando Zvietcovich Zegarra.

**Lima, junio del 2014**

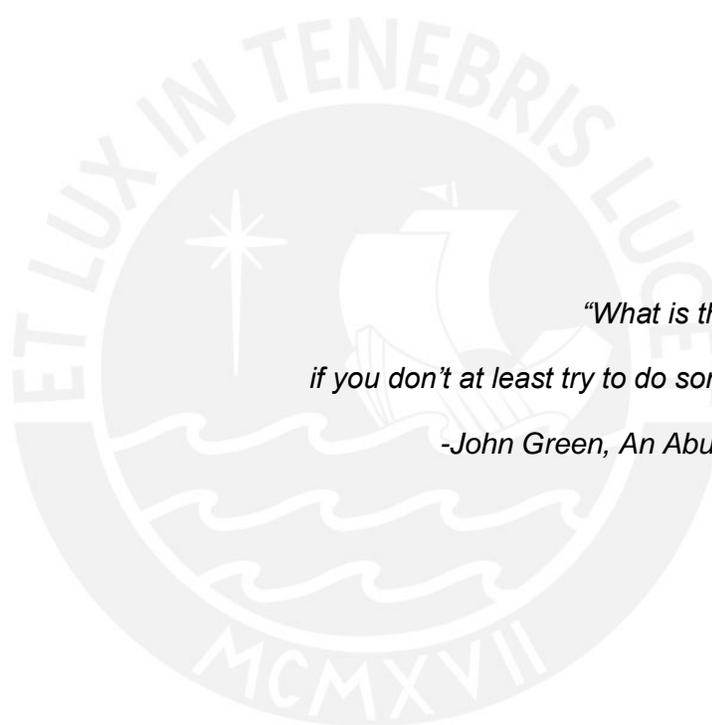
## Resumen

En la actualidad, las diversas técnicas y dispositivos de reconstrucción tridimensional [1][2] permiten obtener una representación fiel a la realidad del objeto de estudio. Si bien existen equipos que extraen de manera completa toda una escena, en muchas ocasiones el equipo tiene un rango de visión limitado y será necesario tomar varias muestras de distintos ángulos para luego alinearse en una imagen más grande.

Esta etapa conocida como registro es el primer paso para poder llevar a cabo la reconstrucción tridimensional por lo que es una parte crítica de todo el proceso. Si se tienen muestras alineadas correctamente se podrá reconstruir, fiel a la realidad el objeto, caso contrario se podrían tener dimensiones o relieves erróneos. En el campo de la arqueología, diversos dispositivos y técnicas se han venido aplicando dependiendo si se trabajará en un objeto como una cerámica o una estructura [2]. Debido a que el trabajo se desarrolla usualmente *in situ*, existe el problema de portabilidad y el requerimiento de personal capacitado para el manejo de los equipos [23], así como el elevado precio que los equipos usados mayormente poseen.

La presente investigación desarrolla un conjunto de herramientas computacionales para realizar el registro de nubes de puntos empleando un Kinect (Microsoft, Washington EE.UU.) como sensor, una alternativa contemporánea para medir profundidad. A partir de la toma de muestras consecutivas a lo largo de un muro arqueológico se realizará el registro y afinamiento de las muestras. Utilizando técnicas de triangulación (Delaunay [29]) y reconstrucción de Poisson [30] se comparará la reconstrucción obtenida a base de muestras del Kinect con técnicas comerciales empleadas en el ámbito arqueológico.

En este sentido, se caracterizó el Kinect con un error de precisión de 2.2 milímetros y 6.3 milímetros de exactitud en profundidad. Se elaboró una metodología para la adquisición y registro de muestras en base al Kinect. Finalmente, se comparan el modelo tridimensional reconstruido con modelos de la misma escena de estudio empleando técnicas comerciales en arqueología. Resultados preliminares indican un error de 4.74 centímetros frente a la reconstrucción por fotogrametría y 1.49 centímetros frente a la de un escáner láser (*NextEngine Laser Scanner HD*).



*“What is the point of being alive  
if you don’t at least try to do something remarkable?”*

*-John Green, An Abundance of Katherines*

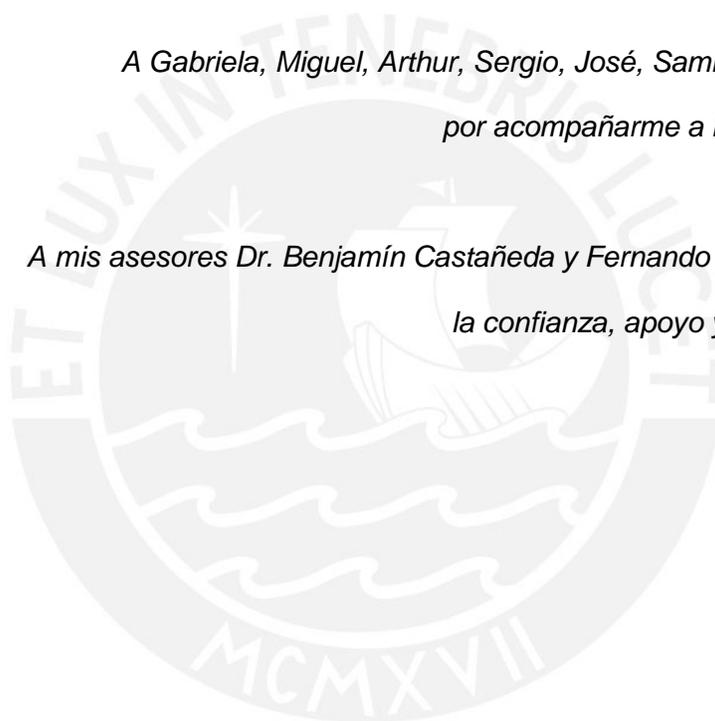
*A mis padres, por la educación.*

*A Adriana y Patricia, por todo el apoyo a lo largo de los años y  
siempre dibujarme una sonrisa en el rostro.*

*A Valeria, Mary, Irene, Johnny y los dos Diegos,  
mis incondicionales amigos y segunda familia.*

*A Gabriela, Miguel, Arthur, Sergio, José, Sammy, Humberto y Luis,  
por acompañarme a lo largo de la carrera.*

*A mis asesores Dr. Benjamín Castañeda y Fernando Zvietcovich, por toda  
la confianza, apoyo y paciencia mostrada*



## Índice

<u>Introducción</u> .....	1
<u>Capítulo 1: Reconstrucción de Modelos Superficiales Tridimensionales</u> .....	2
1.1 <u>Generalidades</u> .....	2
1.2 <u>Técnicas de reconstrucción tridimensional</u> .....	2
1.2.1 <u>Escáneres de rango</u> .....	2
1.2.2 <u>Escáneres de triangulación</u> .....	4
1.2.3 <u>Escáneres de luz estructurada</u> .....	5
1.2.4 <u>Fotogrametría</u> .....	5
1.3 <u>Procedimiento de la reconstrucción tridimensional</u> .....	7
1.4 <u>Consideraciones</u> .....	8
1.5 <u>Objetivos</u> .....	10
1.5.1 <u>Objetivo general</u> .....	10
1.5.2 <u>Objetivos específicos</u> .....	10
<u>Capítulo 2: Características del Kinect y determinación de errores</u> .....	11
2.1 <u>Historia</u> .....	11
2.2 <u>Características</u> .....	12
2.2.1 <u>Eléctricas</u> .....	12
2.2.2 <u>Componentes</u> .....	13
2.2.3 <u>Limitaciones</u> .....	14
2.2.4 <u>Programas de desarrollo</u> .....	15
2.3 <u>Caracterización del Kinect</u> .....	16
2.3.1 <u>Elaboración de maqueta de pruebas</u> .....	16
2.3.2 <u>Error de precisión</u> .....	16
2.3.3 <u>Error de exactitud</u> .....	19
2.3.4 <u>Análisis de resultados</u> .....	23
<u>Capítulo 3: Desarrollo de Algoritmo de Registro de Nubes de Puntos</u> .....	25
3.1 <u>Metodología del diseño</u> .....	25
3.2 <u>Adquisición de las muestras empleando el Kinect</u> .....	26
3.3 <u>Transformación de las muestras a nubes de puntos</u> .....	28
3.4 <u>Registro rígido de nubes de puntos</u> .....	30

3.4.1	<u>Arquitectura del método de registro</u> .....	30
3.4.1.1	<u>Localización de puntos característicos y determinación de puntos en común</u> 32	
3.4.1.2	<u>Determinación de alineación inicial</u> .....	34
3.4.1.3	<u>Refinado de Alineación</u> .....	36
3.5	<u>Creación de mallas partir de las nubes de puntos</u> .....	41
3.6	<u>Discusión y sugerencias</u> .....	41
<u>Capítulo 4: Análisis de Resultados</u> .....		42
4.1	<u>Visualización de registro y malla</u> .....	42
4.2	<u>Comparación con otras técnicas de reconstrucción</u> .....	43
4.2.1	<u>Fotogrametría</u> .....	43
4.2.2	<u>Escáner de triangulación</u> .....	45
4.2.3	<u>Análisis de resultados</u> .....	46
<u>Conclusiones</u> .....		50
<u>Recomendaciones</u> .....		51
<u>Bibliografía</u> .....		52

## Índice de figuras

Figura 1.1 Principio de tiempo de vuelo: Un pulso laser es emitido y reflejado por un espejo hacia el exterior, al rebotar en el objeto este será capturado por el receptor, en muchos casos el mismo emisor, logrando así poder determinar el tiempo que demoró el láser en regresar. Adaptado de [1].	3
Figura 1.2 El error tangencial se presenta debido a que, frente a cambios muy bruscos el láser es incapaz de obtener los valores correctos, puntos grises y negros, del objeto de estudio y en su lugar obtiene los puntos rojos. Adaptado de [2].	3
Figura 1.3 Principio de triangulación: El láser emitido rebota en el espejo e incide en el objeto. Esta posición es capturada por la cámara CDD. Para obtener la posición de distintos puntos, el espejo gira de tal manera que el ángulo en el cual incide el láser al objeto varia. Esta variación altera las dimensiones del triángulo formado por el emisor, punto de incidencia y la cámara logrando así una nueva medida de la distancia del objeto [1].	4
Figura 1.4 Varias muestras unidas procedentes del escáner de triangulación. El color representa un momento diferente de adquisición. Adaptado de [4].	5
Figura 1.5 En (a) se ve el principio de funcionamiento del escáner de luz estructurada, adaptado de [4]. Por otro lado, en (b) observamos el patrón de luz estructurada del Kinect de Microsoft. En él podemos ver rectángulos, cuando estos se deformen debido a la superficie del objeto, el sensor entenderá eso como un cambio de profundidad. Adaptado de [5].	6
Figura 1.6 Secuencia de muestras fotogramétricas: Para obtener información de la posición del objeto serán necesaria dos muestras para realizar la triangulación. No obstante, a cuantas más muestras se tomen, de distintos ángulos al mismo objeto de estudio, se obtendrá más información tridimensional [6].	6
Figura 1.7 Imagen de la reconstrucción de la excavación con las respectivas posiciones de las imágenes usadas el <i>stereo matching</i> . Adaptado de [7].	7
Figura 2.1 Sensor Kinect.	11
Figura 2.2 Sensor Kinect y sus componentes: El proyector infrarrojo, un indicador LED que nos indica el estado del dispositivo (se enciende de color amarillo al estar energizado), la cámara RGB, la cámara (receptor) infrarrojo, a los extremos se encuentran los micrófonos y en la base un servomotor.	13
Figura 2.3 Maqueta fabricada para la medición de errores.	16
Figura 2.4 Toma de muestras a diferentes distancias en el Laboratorio de Imágenes Médicas (LIM) de la PUCP.	17
Figura 2.5 En (a) el segmento plano de la maqueta tal cual se extrae del Kinect, las medidas están en milímetros. Luego, en (b) el mismo segmento transformado a nube de puntos cuya vista frontal se ve en (c), las medidas están expresadas en metros. ...	18
Figura 2.6 Gráficas de las desviaciones de las medidas reales versus las distancias a las que fueron tomadas.	18
Figura 2.7 Escáner NextEngine Laser Scanner HD.	19

Figura 2.8 En (a) podemos ver el resultado de la adquisición de datos y luego en (b) la misma imagen con la maqueta segmentada. Las unidades de esta última están en milímetros. .... 20

Figura 2.9 La transformación de la imagen de rango a nube de puntos se ve en (a). Por otro lado, (b) se realiza la identificación de puntos de control manual entre la nube de puntos procedente del Kinect (izquierda) y del *NextEngine Laser Scanner HD* (derecha)..... 21

Figura 2.10 En la primera columna, vemos el resultado de la diferencia de las nubes de puntos provenientes del Kinect y del *NextEngine Laser Scanner HD* representada por una nube colorizada. En la segunda columna vemos los histogramas respectivos de los errores con el ajuste Gaussiano. Los pares de figuras (a), (b) y (c) representan las medidas a las distancias de 0.85, 1.10 y 1.35 metros respectivamente ..... 22

Figura 2.11 Gráfica de La media de la diferencia de las nubes de puntos versus la distancia a la que el Kinect tomó las muestras en donde figuran también las desviaciones de cada media. .... 23

Figura 3.1 Esquema metodológico del algoritmo. .... 25

Figura 3.2 En (a) vemos el montaje del Kinect para iniciar la adquisición de datos. En (b) la dirección en la cual una persona moverá el Kinect durante la adquisición. Por otro lado, en (c) y (d) observamos una de las muestras obtenidas con el Kinect, de color y de profundidad respectivamente. .... 28

Figura 3.3 Se procede a formar un nuevo conjunto de muestras escogiendo la actual y descartando las dos siguientes se hace el mismo procedimiento con las imágenes de profundidad..... 29

Figura 3.4 Nubes de puntos sin color (a) y con color (b) en formato PLY visualizadas en CloudCompare. .... 29

Figura 3.5 Arquitectura del sistema en su totalidad, en azul el proceso de registro. .... 31

Figura 3.6 Puntos en común de dos muestras unidos por la línea celeste. En (a) existen 55 puntos en común mientras que en (b) solo 6. .... 33

Figura 3.7 Método con el cual se empleó el algoritmo SURF. Todas las muestras de cada grupo son comparadas con la primera del mismo grupo para no disminuir el número de puntos en común. El tamaño del grupo es de 4 para este ejemplo. .... 34

Figura 3.8 Las nubes de puntos de (a) serán alineadas mediante las matrices R y T obtenidas luego de aplicar SURF. El resultado se puede ver en (b) en donde la parte manchada (mezcla de colores) representa las nubes alineadas en esa región. .... 36

Figura 3.9 En (a) se ve como hay un desplazamiento de la nube roja mientras que en (b) se aprecia gran traslape de ambas. .... 38

Figura 3.10 Se aplica el algoritmo ICP de manera combinatoria para cada grupo mejorando así la alineación en cada grupo. Se toma como ejemplo un grupo de 4 muestras. .... 38

Figura 3.11 Proceso acumulativo de operar las nubes de puntos con las R y T involucradas. El método de operación a seguir es el descrito por la ecuación 3.15 y 3.16..... 40

Figura 3.12 Ilustración intuitiva de la reconstrucción de Poisson. Adaptado de [30]. ... 41

Figura 4.1 Nubes de puntos registradas: En (a) las nubes de puntos con color. Los colores varían debido a la luz solar en el momento de adquisición. En (b) las nubes registrados ya reconstruidas mediante el método de Poisson [30].	42
Figura 4.2 En (a) nube de puntos sin color procedente de fotogrametría del primer caso del Camino Inca y en (b) la reconstrucción del mismo luego de haber aplicado Poisson [30].	43
Figura 4.3 Alineaciones de reconstrucciones tridimensionales de fotogrametría a color y Kinect en verde. El primer caso del Camino Inca en (a), adquisición modo 1 del Kinect y (b) con modo 2. En (c) es el segundo caso del Camino Inca con adquisiciones modo 2.	44
Figura 4.4 En (a) adquisición de datos del primer caso del Camino Inca y en (b) la alineación de reconstrucción del <i>NextEngine Laser Scanner HD</i> y Kinect. El modelo a color es resultado del escáner y el blanco del Kinect.	45
Figura 4.5 Sustracción de modelos tridimensionales e histograma con ajuste Gaussiano para los modelos del <i>NextEngine Laser Scanner HD</i> y el Kinect.	46
Figura 4.6 Sustracción de modelos tridimensionales e histograma con ajuste Gaussiano para los modelos de Fotogrametría y Kinect. Primer caso del Camino Inca con muestras de adquisición modo 2 del Kinect en (a) y modo 1 en (b).	47
Figura 4.7 Sustracción de modelos tridimensionales e histograma con ajuste Gaussiano para los modelos de Fotogrametría y Kinect. Segundo caso del Camino Inca con muestras de adquisición modo 2 del Kinect.	48



## Introducción

La tecnología de escaneo en tres dimensiones ha evolucionado considerablemente durante las últimas décadas. El uso de esta radica principalmente en poder extraer un modelo computacional geométrico de un objeto físico. Dentro de todas las tecnologías disponibles para las diversas aplicaciones, se pueden encontrar productos de todo precio, exactitud, precisión y resolución. La importancia de un sistema de bajo costo que cumpla con las especificaciones requeridas para determinado proyecto es indispensable. Muchos sensores disponibles actualmente en el mercado pueden lograr costos accesibles pero aun así tienen inconvenientes en cuanto a portabilidad y modo de uso. Sin embargo, la aparición del sensor Kinect ha permitido lidiar con estos aspectos y equilibrar dichos requerimientos. Aunque su propósito principal está en el rubro del entretenimiento, las características técnicas del Kinect (Microsoft, Washington EE.UU.) hacen que este pueda ser empleado en otras áreas. El objetivo principal de esta tesis es desarrollar un programa capaz de registrar una secuencia de nubes 3D producidos por este sensor de bajo costo. Como resultado de este programa se obtiene un modelo geométrico y de color de un objeto, como por ejemplo un muro arqueológico, de manera que se aprecie qué tan buena es la reconstrucción comparando su exactitud con otros métodos de reconstrucción actualmente disponibles en el mercado, empleados en aplicaciones de arqueología.

En el primer capítulo se explicará en qué consiste la reconstrucción tridimensional, y el estado del arte de las técnicas y dispositivos que se emplean en la actualidad. Así mismo, se detallará la problemática existente en el uso de dichas técnicas y los objetivos de la presente tesis. En el segundo capítulo se explicarán las características del sensor Kinect así como los experimentos que se llevaron a cabo para determinar sus parámetros. Posteriormente, en el capítulo tres, se explicará el algoritmo desarrollado para llevar a cabo el registro automático 3D de nubes de puntos, detallando todos los pasos y procedimientos necesarios. Finalmente, en el capítulo cuatro se analizarán los resultados obtenidos y se compararán con resultados de reconstrucciones de otras las técnicas empleadas. Finalmente, se detallarán las conclusiones y recomendaciones de esta investigación.

# Capítulo 1: Reconstrucción de Modelos Superficiales

## Tridimensionales

### 1.1 Generalidades

La reconstrucción tridimensional consiste en tener una representación en tres dimensiones de un objeto o una región. Para poder realizarla, es necesario primero obtener información acerca del objeto o escena a trabajar mediante algún dispositivo, sea una cámara u otro tipo de sensor.

En este aspecto, la tecnología de escaneo en tres dimensiones ha venido evolucionando considerablemente durante las últimas décadas. Desarrollándose así sensores y escáneres tridimensionales (3D) con los que es posible extraer un modelo geométrico de la superficie de un objeto físico de distintas maneras.

### 1.2 Técnicas de reconstrucción tridimensional

En la actualidad, no existe una definición general aceptada para definir que dispositivos o no deberían considerarse escáneres 3D. Desde el punto de vista del usuario, un escáner 3D es cualquier dispositivo que recolecte coordenadas tridimensionales de una región de la superficie del objeto seleccionado [1].

Estos dispositivos se caracterizan por recolectar la información de manera automática, en un patrón sistemático, a una velocidad muy alta (miles de puntos por segundos) y obteniendo resultados en tiempo real. A continuación se detallarán las técnicas y sus respectivos dispositivos más usados.

#### 1.2.1 Escáneres de rango

Dentro de esta categoría se ubican los dispositivos que trabajan midiendo el tiempo de vuelo de un pulso láser, es decir, el tiempo que demora en ir y regresar el pulso que fue enviado y rebota en la superficie del objeto [1], tal como se aprecia en la Figura 1.1. Dado que la velocidad con la que viaja el láser es la velocidad de la luz (ver la

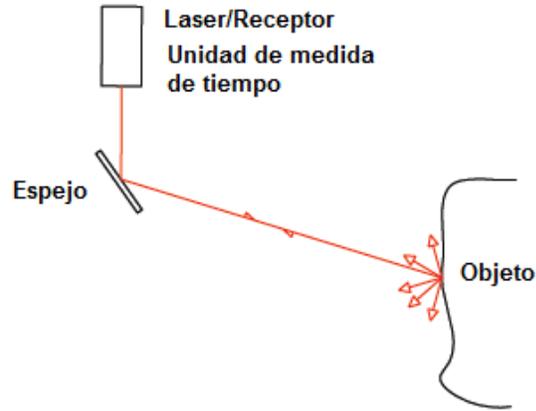


Figura 1.1 Principio de tiempo de vuelo: Un pulso láser es emitido y reflejado por un espejo hacia el exterior, al rebotar en el objeto este será capturado por el receptor, en muchos casos el mismo emisor, logrando así poder determinar el tiempo que demoró el láser en regresar. Adaptado de [1].

ecuación 1.1), al encontrar el tiempo que demora este en ir y regresar al receptor, se calcula fácilmente la distancia a la que se encuentra el punto usando la relación 1.2. Esto es posible debido a que  $t$  es el tiempo que tarda la onda en ir y regresar del objeto, además la velocidad del láser (luz) es constante, consecuencia de un Movimiento Rectilíneo Uniforme (MRU).

$$c = 3 \times 10^8 \text{ m/s} \quad (1.1)$$

$$d = v \times \frac{t}{2} \quad (1.2)$$

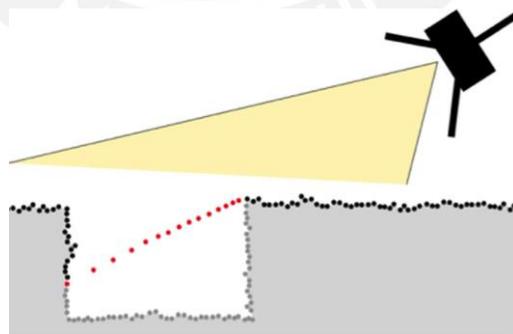


Figura 1.2 El error tangencial se presenta debido a que, frente a cambios muy bruscos el láser es incapaz de obtener los valores correctos, puntos grises y negros, del objeto de estudio y en su lugar obtiene los puntos rojos. Adaptado de [2].

En [2] vemos un caso de aplicación real en donde se emplean numerosos escáneres de este tipo para trabajo de campo arqueológico. Se evidencia su gran utilidad para objetos de dimensiones pequeñas y el problema que existe cuando el objeto posee cambios bruscos, lo que se suele denominar error tangencial como se aprecia en la Figura 1.2. Finalmente, existen escáneres de rango que permiten un mayor alcance debido a que la cámara que emite el láser puede girar 180 grados capaces de obtener información de estructuras enteras. Un ejemplo de estos dispositivos se puede apreciar en [3], en donde el objeto de estudios involucraba el monumento arqueológico Huaca de la Luna.

### 1.2.2 Escáneres de triangulación

En esta otra categoría, los equipos emiten un haz de láser que al rebotar en el espejo, incide en el objeto y mediante una cámara CDD (Dispositivo de Carga Acoplada) se captura la ubicación del láser. El emisor, el punto de incidencia del láser en el objeto y la cámara forman un triángulo. La longitud del lado del triángulo definido por la cámara y el emisor es conocida, además, el ángulo del vértice del emisor al láser también se conoce. Por otro lado, el ángulo del vértice de la cámara puede ser determinado mirando la ubicación del punto láser en la misma cámara.

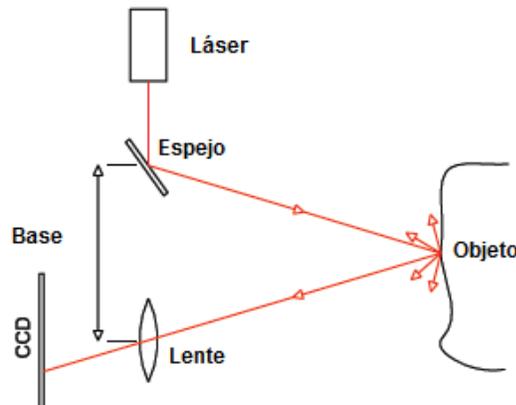


Figura 1.3 Principio de triangulación: El láser emitido rebota en el espejo e incide en el objeto. Esta posición es capturada por la cámara CDD. Para obtener la posición de distintos puntos, el espejo gira de tal manera que el ángulo en el cual incide el láser al objeto varía. Esta variación altera las dimensiones del triángulo formado por el emisor, punto de incidencia y la cámara logrando así una nueva medida de la distancia del objeto [1].



Figura 1.4 Varias muestras unidas procedentes del escáner de triangulación. El color representa un momento diferente de adquisición. Adaptado de [4].

Finalmente, estos tres valores permiten determinar las dimensiones restantes del triángulo, entre ellas la distancia del punto en el objeto [1]. Esta misma solución se puede llevar a cabo tanto con una cámara o con dos. La Figura 1.3 muestra la representación con una sola cámara. Un ejemplo del uso de escáneres de triangulación en material arqueológico se evidencia en el trabajo de fragmentos de la estatua de Terracotta, como se puede ver a continuación en la Figura 1.4. Para mayor información, consultar [4].

### 1.2.3 Escáneres de luz estructurada

El principio de funcionamiento de estos dispositivos es muy similar al de los escáneres de rango, como se puede apreciar en la Figura 1.5 (a). La diferencia radica en que la distancia del objeto no se determina a partir del tiempo que demora en retornar la señal. En este caso, se proyecta una señal conocida sobre el objeto o escena deseada. Este patrón puede ser bandas de frecuencia, luz codificada o un patrón de formas y al observar cómo este patrón es deformado conforme incide en las superficies es que se calcula la distancia [4]. El patrón de luz estructurada del Kinect se puede apreciar en la Figura 1.5 (b).

### 1.2.4 Fotogrametría

Otra técnica que existe para poder realizar reconstrucciones tridimensionales es la fotogrametría digital de corto o largo alcance [6]. La fotogrametría consiste en determinar las propiedades de un objeto mediante imágenes en dos dimensiones, es

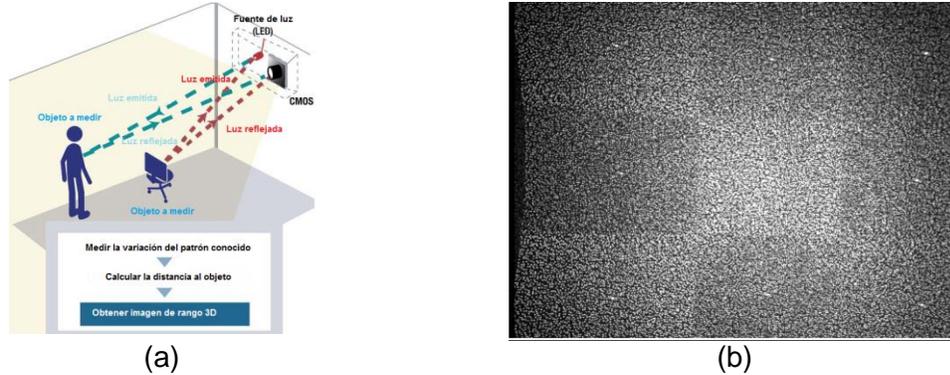


Figura 1.5 En (a) se ve el principio de funcionamiento del escáner de luz estructurada, adaptado de [4]. Por otro lado, en (b) observamos el patrón de luz estructurada del Kinect de Microsoft. En él podemos ver rectángulos, cuando estos se deformen debido a la superficie del objeto, el sensor entenderá eso como un cambio de profundidad. Adaptado de [5].

decir, fotografías. Esta técnica no necesita de equipos sofisticados y puede llevarse a cabo con cualquier dispositivo que nos permita tomar fotografías. La resolución y precisión dependerá estrictamente de la calidad del dispositivo a usar. Para obtener la posición del objeto, se cruzarán matemáticamente las líneas de vista de ambas tomas para producir coordenadas tridimensionales, como podemos ver en la Figura 1.6. Como ya se mencionó, se tendrán varias fotos de distintas vistas de objeto o escena a trabajar, será necesario encontrar zonas en común entre las mismas a fin de recrear tridimensionalmente todo el objeto, este proceso se suele llamar unión estéreo (*stereo matching*). Sin embargo, serán necesarias ciertas mediciones físicas del objeto pues no se tendrá información de la escala.

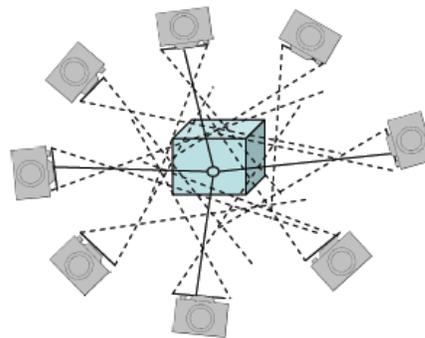


Figura 1.6 Secuencia de muestras fotogramétricas: Para obtener información de la posición del objeto serán necesaria dos muestras para realizar la triangulación. No obstante, a cuantas más muestras se tomen, de distintos ángulos al mismo objeto de estudio, se obtendrá más información tridimensional [6].

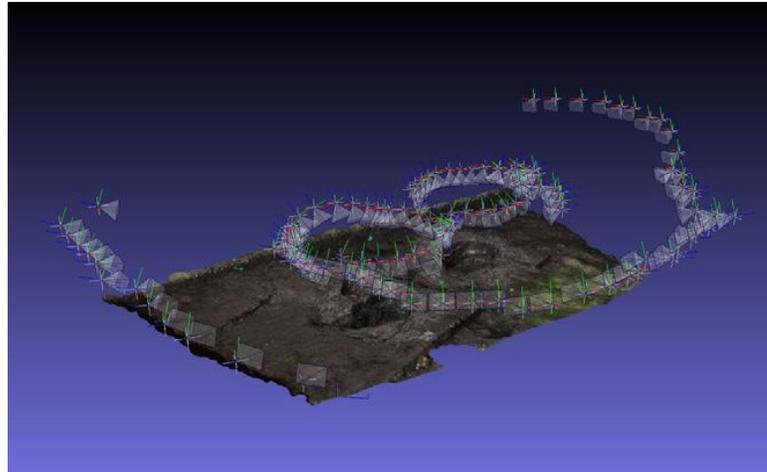


Figura 1.7 Imagen de la reconstrucción de la excavación con las respectivas posiciones de las imágenes usadas el *stereo matching*. Adaptado de [7].

Esta técnica permite abarcar grandes porciones de área, como se ve en la Figura 1.7, al usar diversas fotos de distintas vistas, es posible recrear la escena. Este caso particular es una excavación arqueológica desarrollada en [7].

### 1.3 Procedimiento de la reconstrucción tridimensional

La información procedente de los escáneres o sensores son, en su mayoría, nubes de puntos. Esta nube de puntos describe las coordenadas Cartesianas de cada uno de los puntos, no unidos entre sí, ubicados en un espacio tridimensional. Es llamado “nube” debido a que al ser observados, la falta de conectividad entre ellos hace parecer que flotan en el espacio [5].

Una sola muestra no es suficiente para realizar la reconstrucción de una estructura en su totalidad. Es por ello que varias muestras serán necesarias para tener el modelo tridimensional completo. El proceso para alinear estas muestras se denomina registro.

El concepto de registro, en este caso, se refiere obtener la información necesaria que permita relacionar y alinear varias nubes de puntos (muestras), que describen diferentes partes de un objeto, en un mismo espacio 3D. De esta manera, se podrá construir un modelo que contenga información de varias vistas y describa de forma unificada un objeto que es, en principio, mucho más grande de lo que el sensor permite obtener en una sola toma [8].

Este proceso, por lo general, consta de tres pasos fundamentales para llevarse a cabo. El primero de ellos consiste en determinar puntos descriptores de cada imagen (nube), es decir, puntos que son únicos para la imagen debido a características como por ejemplo el color. Estos se pueden encontrar de manera visual, seleccionándolos manualmente o empleando algún algoritmo que los relacione como SIFT, SURF, entre otros [9]. El segundo paso es determinar la similitud entre los puntos descriptores de dos imágenes, también llamado *matching*, de manera que exista una correspondencia entre ambas imágenes. Estos se encontrarán en la zona en común entre ambas imágenes y a cuantos más puntos en común se tengan, será más preciso el registro. Finalmente, será necesario tomar una de las nubes de puntos como referencia, de modo que las otras serán transformadas para que se sitúen en un mismo plano coordinado determinado por la referencia, teniendo como objetivo unir sus puntos en común. Al tratar las imágenes como nubes de puntos y estas como matrices tridimensionales, es necesario trasladar y rotar la nube de puntos para que se alinee con otra. Para encontrar las matrices de rotación y el vector de traslación se emplean diversos algoritmos como SVD, ICP y sus variantes [10].

Una vez concluido estos tres pasos se tendrán varias nubes de puntos (procedentes de las imágenes) en un único sistema coordinado que representarán tridimensionalmente la superficie del objeto en estudio, con lo que es posible crear una malla 3D a partir de estos puntos.

Una malla 3D consiste en una superficie formada por polígonos, generalmente triángulos, los cuales a su vez están formados por los puntos de la nube que representan los vértices. Además es posible indexar la información de textura a cada uno de los vértices que conforman la malla lo que permite que los modelos sean útiles para aplicaciones de realidad virtual.

#### **1.4 Consideraciones**

En todo tipo de mediciones, el factor determinante es la precisión y esta escala con el costo comercial de los equipos. Escáneres con una precisión de 0.1 mm pueden llegar a costar \$50,000 [6]. Debido a que la reconstrucción se hace para objetos de todo tamaño o forma, no es posible recomendar un solo escáner para todas las tareas o usos. Mientras los escáneres de tiempo de vuelo permiten reconstruir grandes

territorios y edificaciones, los escáneres de triangulación están diseñados, mayormente, para objetos pequeños. En la práctica, no existen parámetros mínimos de precisión para arqueología. Considerando la magnitud del objeto de estudio se tiende a seleccionar el escáner más adecuado.

Por otro lado, existen otros aspectos también importantes como son las limitaciones del lugar donde se pretende realizar un levantamiento 3D. Por lo general las mediciones se realizan *in situ* y el polvo juega en contra de muchos escáneres que no están hechos para trabajar en ambientes así. Además, la mayoría de equipos suelen ser pesados y requieren de mucho cuidado al ser transportados y su tiempo de adquisición de datos es muy elevado, lo cual los hace más propensos al deterioro debido al ambiente agresivo. Por último, en la mayoría de los casos existe la necesidad de una persona capacitada para el uso del escáner y del software del mismo, así como de una computadora con ciertas características específicas.

Por ejemplo, en el Laboratorio de Imágenes Médicas (LIM) de la Pontificia Universidad Católica del Perú, se cuenta con el equipo *NextEngine Laser Scanner HD*. Este se emplea en diversas aplicaciones, tanto médicas [11] como arqueológicas [12]. Pese a tener un performance muy bueno a un precio relativamente bajo para un escáner láser de triangulación (\$2,995 sin contar los software) presenta inconvenientes como los mencionados anteriormente.

Todo esto pone en evidencia que es necesario un equipo y un software de procesamiento de imágenes que permita la reconstrucción de material arqueológico, que sea económico, fácil de transportar y pueda ser utilizado por usuarios no expertos.

La alternativa de solución que se propone es el sensor Kinect de Microsoft. Este dispositivo, inicialmente concebido como una interfaz entre el usuario y la consola de videojuegos, está siendo empleado en diversas áreas de investigación. Este equipo ya satisface los requerimientos de precio y portabilidad, además de tener compatibilidad con distintos programas de desarrollo como Microsoft Visual Studio, MATLAB, OpenCV, entre otros.

## 1.5 Objetivos

### 1.5.1 Objetivo general

La presente tesis tiene como objetivos diseñar e implementar un programa capaz de registrar una secuencia de nubes de puntos producidos por el Kinect de Microsoft. Como resultado de este programa se obtendrá un modelo geométrico y de color de un objeto, como por ejemplo de un muro arqueológico, de manera que se apreciará qué tan buena es la reconstrucción comparando su exactitud con otros métodos de reconstrucción actualmente disponibles en el mercado.

### 1.5.2 Objetivos específicos

1. Caracterización del equipo Kinect for XBOX de Microsoft en exactitud y precisión.
2. Corroborar la utilidad del equipo para la representación de material arqueológico.
3. Elaborar comparaciones con otras técnicas de reconstrucción como fotogrametría y escáneres de triangulación.

## Capítulo 2: Características del Kinect y determinación de errores

### 2.1 Historia

El Kinect (ver la Figura 2.1), es un sensor de movimiento desarrollado por Microsoft para su consola de videojuegos XBOX 360 y Windows PC. Este dispositivo fue creado bajo la premisa que no se necesitarían controles para interactuar con los videojuegos y aplicaciones. A un precio de S/.299 en el país [13] este dispositivo se encargaría de leer los movimientos del usuario y estos representarían acciones en los distintos videojuegos y aplicaciones. Pese a que fue desarrollado por Microsoft, los componentes y circuitos del sensor de profundidad fue diseñado por la empresa israelí PrimeSense.

Casi en paralelo al lanzamiento mundial del dispositivo, PrimeSense, distribuyó sus propios drivers y framework de programación para el Kinect llamado OpenNI. Este entorno nos permite manejar cualquier dispositivo de interacción natural, es decir que no necesite de ningún tipo de contacto físico entre el usuario y el dispositivo. En el caso del Kinect, debido a que PrimeSense solo desarrolló el hardware del sensor de profundidad, solo era posible obtener información de este y se perdían todas las demás características del dispositivo.



Figura 2.1 Sensor Kinect.

A los meses de este acontecimiento, se anunció una alianza entre PrimeSense y su segundo mayor concesionario de hardware, la compañía de computadoras taiwanesa ASUS. Juntos anunciaron que producirían el primer kit profesional de desarrollo de software para sensores de movimiento diseñado para computadoras personales, el WAVI Xtion PRO [14].

La diferencia con el Kinect radica en que este producto fue concebido con la intención de conectarse a las computadoras personales y que contenga herramientas incluidas para facilitar su programación, no obstante, presenta características físicas y dimensionales similares al Kinect. Además, este dispositivo no está actualmente disponible en el país y tiene un precio de \$179.99 [15] en el exterior y cuenta con una tienda online para sus propias aplicaciones en donde los desarrolladores podrían vender sus programas.

## **2.2 Características**

### **2.2.1 Eléctricas**

Como en un primer momento se pensó que el Kinect for XBOX 360 solo funcionaría conectado a la consola XBOX 360, este dispositivo fue diseñado con un cable USB que se conecta directamente a la consola de videojuegos. Este no es un cable USB estándar debido a que este dispositivo consume una potencia de 12 Watts.

Posteriormente, al enfatizarse el uso del Kinect con una computadora personal se vio la necesidad de diseñar un adaptador, para que el nuevo conector del Kinect iguale las características eléctricas del puerto USB estándar y además pueda energizarse de alguna fuente externa. Es por esto que comenzaron a comercializar adaptadores AC (100 ~ 240 V y 47 – 63 HZ), permitiendo así la conexión a la red eléctrica como fuente de alimentación, una regulación a 12VDC y 1A que son los requisitos del Kinect y una salida USB estándar.

Una vez que se conecta el sensor a este accesorio, es posible trabajar con el mediante cualquier puerto USB. A pesar de esto, su consumo sigue siendo elevado y nos limita a usarlo con pocos periféricos conectados a la computadora.

### 2.2.2 Componentes

Este sensor cuenta con una cámara RGB con resolución de 640x480 píxeles, un sensor de profundidad que consiste en un proyector de láser infrarrojo y un sensor infrarrojo CMOS, ambos trabajan a 30 cuadros por segundo. Además, posee un conjunto de micrófonos que permiten la localización por fuente acústica y supresión de ruido del ambiente. Además, posee un indicador LED de encendido, un acelerómetro tri-axial y un pequeño servo motor [16]. Como se puede observar a continuación en la Figura 2.2.

Analizando con mayor detenimiento cómo funciona el sensor de profundidad del Kinect, podemos identificar dos partes principales: el emisor infrarrojo (IR) y la cámara infrarroja. El emisor IR crea un patrón de luz infrarroja estructurada a 830nm. La obtención de la información trabaja bajo un principio de luz estructurada. Existe un patrón pseudoaleatorio y conocido de puntos que es enviado desde el emisor. Estos puntos son grabados por la cámara IR receptora y luego son comparados con un patrón conocido (ver Figuras 1.5 (a) y (b)). Cualquier disturbación se reconoce como una variación de la superficie y puede ser detectada como cercanía o lejanía. Esta información luego es procesada y se transmite de manera serial en 11 bits, lo cual nos da hasta 2048 niveles de profundidad.

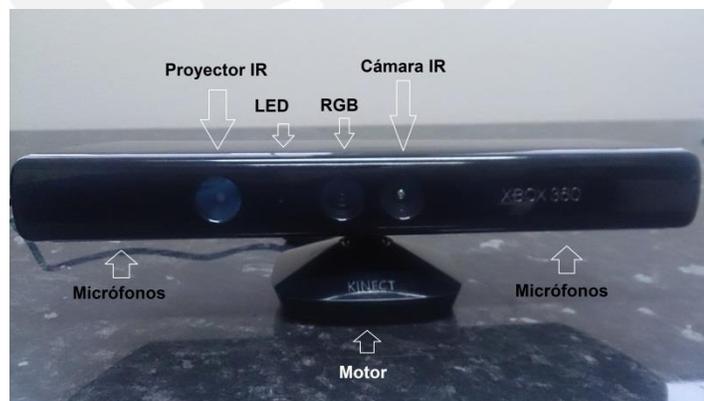


Figura 2.2 Sensor Kinect y sus componentes: El proyector infrarrojo, un indicador LED que nos indica el estado del dispositivo (se enciende de color amarillo al estar energizado), la cámara RGB, la cámara (receptor) infrarrojo, a los extremos se encuentran los micrófonos y en la base un servomotor.

El rango de operación del Kinect for XBOX 360 para que reconozca el entorno es de 0.8 a 4 metros, según el fabricante. Mediante los diversos programas que existen para trabajar con el Kinect, estos datos pueden ser representados en una escala de grises, una imagen térmica o si se procesan posteriormente, una nube de puntos.

### 2.2.3 Limitaciones

Debido a que el principio de funcionamiento del Kinect parte de la emisión de una luz estructurada, existen tres problemas clave.

- La longitud de onda debe ser constante.
- La luz del ambiente dificulta la toma de datos.
- La distancia es limitada por la fuerza del emisor infrarrojo.

La constancia de la longitud de onda es mayormente controlada por el dispositivo. Junto con el sensor, existe un pequeño enfriador/calentador Peltier (circuito que permite el efecto Peltier, el cual consiste en la presencia de calentamiento y enfriamiento en una unión eléctrica de dos conductores diferentes) que mantiene al diodo láser a una temperatura constante. Esto asegura que la longitud de onda permanezca lo más constante posible, incluso si hay variaciones en la temperatura o en la potencia disipada.

La luz del ambiente es el punto débil de los sensores de luz estructurada, aunque existen medidas para mitigar este efecto. Uno de ellos es el filtro pasa bajos infrarrojo a 830 nm antes de la cámara. Esto previene que ondas en otros rangos, como por ejemplo las del control remoto de un televisor (940 nm), cieguen al sensor o provean de resultados incorrectos. A pesar de esto, el Kinect no funciona bien en lugares a plena luz del sol. La banda del brillo solar tiene suficiente potencia a 830 nm para cegar por completo al sensor [9]. Por último, como bien ya se especificó, el Kinect no fue diseñado como un instrumento de medición, no obstante, consigue una precisión de 4mm si trabaja en el rango de distancia de 0.4 a 1.5 metros [11]. Esto se da en el Kinect for Windows, que fue desarrollado y comercializado luego de la acogida del Kinect para propósitos de investigación. Es muy parecido al Kinect for XBOX 360, siendo la única diferencia que el Kinect for XBOX 360 trabaja en el rango de 0.8 a 4

metros. Este último será con el que se trabajará dado que es el que posee el Laboratorio de Imágenes Médicas, laboratorio en el cual se realizaron la gran mayoría de pruebas. Finalmente, aún con estas limitaciones presentes, el Kinect puede ser comparado con diversos sensores que existen en el mercado y mostrar buenos resultados frente a ellos. A diferencia de los escáneres láser, el tiempo que requiere para obtener muestras es mucho menor aunque los resultados son menos precisos [17]. Es por esta ventaja de velocidad y su bajo costo que lo posicionan como un sensor importante en robótica y reconocimiento de objetos [18].

#### **2.2.4 Programas de desarrollo**

Existen diversos programas con los que se puede trabajar el Kinect. Dentro de las alternativas open source, figuran OpenCV (OpenCV v2.4.9, 2014, Itseez, Nizhny Novgorod, Rusia) y Point Cloud Library (Point Cloud Library, v1.6.0, 2010, Willow Garage, Menlo Park, California). Cabe resaltar que este último es un conjunto de librerías gráficas que facilitan la generación de nubes de puntos y en general el procesamiento de imágenes en dos y tres dimensiones. Si bien Microsoft desarrolló un programa llamado Kinect SDK (Starter Development Kit), este está más orientado a explotar las habilidades del Kinect como seguimiento de movimiento. Además, también es compatible con Microsoft Visual Studio, Por otro lado, es posible descargar toolboxes para poder trabajar desde MATLAB y aprovechar las funciones de toolboxes existentes como el de Image Processing y Statistics

Todos estos programas nos facilitan el manejo de la información que proviene del sensor. Como nuestro objetivo es obtener nubes de puntos, será necesario escoger un programa adecuado que nos facilite el manejo de las imágenes de profundidad y RGB para así obtener lo deseado. Luego, será necesario realizar el registro de imágenes de todas las muestras y finalmente obtener una representación en tres dimensiones, lo cual indica que el entorno a elegir debe soportar un alto nivel de procesamiento matemático y de imágenes, así como un buen manejo de gráficas 3D.

Se eligió MATLAB [19] debido a la previa familiarización que se tiene con este programa y para aprovechar la variedad de funciones que existen para el manejo de nubes de puntos, registro de imágenes y manejo de matrices. Para la representación

visual de las nubes de puntos, se empleó MeshLab [20] y CloudCompare [21], ambos software open source que facilitan la visualización de nubes de puntos y mallas 3D.

## 2.3 Caracterización del Kinect

### 2.3.1 Elaboración de maqueta de pruebas

Para llevar a cabo la determinación de errores, se procedió a fabricar una maqueta de madera balsa de 33 centímetros de ancho y 45 de largo. Esta consiste en una plataforma plana con figuras geométricas distintas y de diferente tamaño sobresalientes. Como se puede ver en la Figura 2.3, la maqueta consta de 4 medias esferas, un paraboloides sólido y una escalinata, cuyas medidas están representadas en la Tabla 1. Además, se puede ver la posición de los ejes coordenados que se tomaron como referencia.

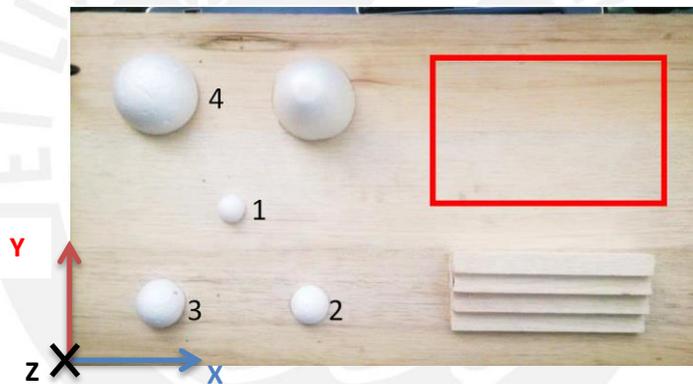


Figura 2.3 Maqueta fabricada para la medición de errores.

### 2.3.2 Error de precisión

Como parte de esta tesis, se procedió a caracterizar el Kinect y de esta manera hallar sus errores. Se procedió a tomar 300 muestras consecutivas a determinadas distancias, en el eje Z, del Kinect como se aprecia en la Figura 2.4. Se comenzó con 0.85 metros hasta llegar a 1.35 metros con intervalos de medio metro. Además se consideraron 3 condiciones de luz en el laboratorio, en donde la menor (Luz 0) consistía en las luces artificiales apagadas y cortinas negras cerradas (luz solar suprimida), entre 30 y 60 lux; la intermedia (Luz1) tenía las luces artificiales prendidas pero con luz solar suprimida, entre 60 y 100 lux y finalmente la mayor (Luz 2) incluía

Tabla 1 Dimensiones de los elementos de la maqueta. Las medidas están en milímetros

Maqueta		Semiesfera				Paraboloide	Escalinata		
Largo	Ancho	Radio1	Radio2	Radio3	Radio4	Altura	Largo	Alto	Altura escalón
450	330	11	15	18.5	31.5	46	150	30	7.5

todas las luces prendidas y las cortinas abiertas donde la intensidad luminosa estaba dentro del rango de 100 y 150 lux.

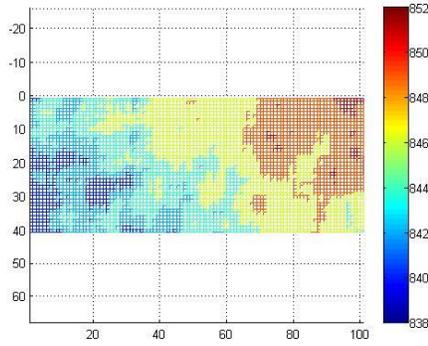
Se seleccionó la parte señalada mediante un recuadro rojo en la Figura 2.3, puesto que la sección es plana y la nube de puntos de ese sector puede ser comparada con la ecuación de un plano perfecto.

Una vez que se adquirieron todas las muestras, se concatenaron con las que compartían la misma distancia. El resultado de esta operación fue una imagen de rango, como se ve en la Figura 2.5 (a), en la cual los colores están asociados a lejanía o proximidad al plano ideal. Para poder realizar medidas más concretas, se procedió a transformar esta imagen a una nube de puntos, ambos resultados se pueden apreciar en la Figura 2.5 (b) y (c).

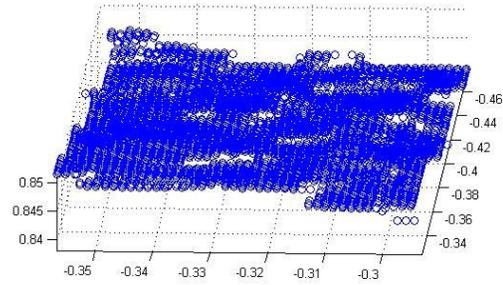
Posteriormente, se elaboró un ajuste un plano a la nube de puntos resultante y este fue rotado hacia un plano  $Z=0$ . De esta manera se pudieron encontrar todas las distancias reales y comprobar si coinciden con la distancia a la que fueron tomadas las muestras. La Figura 2.8 muestra la comparación de la desviación de las mediciones de las muestras así como su respectiva media a su determinada condición de luz.



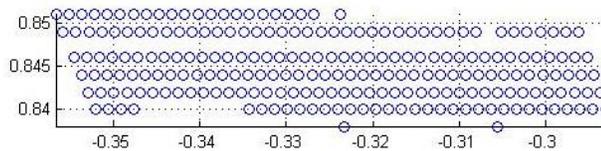
Figura 2.4 Toma de muestras a diferentes distancias en el Laboratorio de Imágenes Médicas (LIM) de la PUCP.



(a) Imagen de rango.



(b) Nube de puntos procedente de la imagen de rango



(c) Vista frontal de la nube de puntos.

Figura 2.5 En (a) el segmento plano de la maqueta tal cual se extrae del Kinect, las medidas están en milímetros. Luego, en (b) el mismo segmento transformado a nube de puntos cuya vista frontal se ve en (c), las medidas están expresadas en metros.

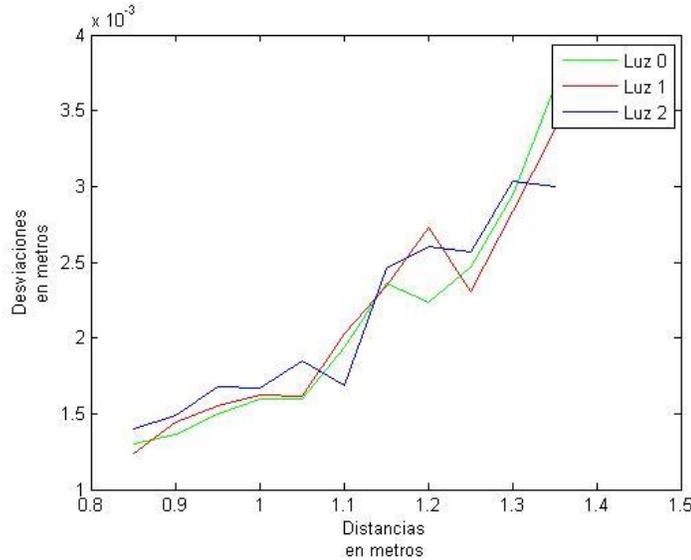


Figura 2.6 Gráficas de las desviaciones de las medidas reales versus las distancias a las que fueron tomadas.

Podemos ver como el error de precisión crece de manera exponencial conforme nos alejamos del sensor Kinect. Analizando estos datos podemos concluir que el Kinect for XBOX360 tiene un error de precisión de 2.2 milímetros en un rango de 0.85 a 1.35 metros a las condiciones de luz de interiores (mediana luz artificial). Estos resultados son comparables con los obtenidos en [17] y [22] en donde especifican un valor de precisión de 3 a 4 milímetros pero para condiciones de luz distintas y con mayores distancias medidas.

### 2.3.3 Error de exactitud

En esta ocasión se buscó averiguar qué tanto difieren las medidas obtenidas por el Kinect con las dimensiones reales, se tomó como referencia la luz 1 para las experiencias. El objeto de prueba fue la maqueta completa (Figura 2.3) y las mediciones extraídas fueron comparadas con las de un sensor de escáner laser *NextEngine Laser Scanner HD*, ver Figura 2.7, del Laboratorio de Imágenes Médicas (LIM) de la PUCP.

Durante este experimento se evidenció la ventaja del Kinect al momento de adquirir los datos. Para las 300 muestras el Kinect demora aproximadamente 1 minuto, es decir un total de 11 minutos para todas las distancias ya mencionadas previamente. Por otro lado, el *NextEngine Laser Scanner HD* (NextEngine HD, NextEngine Inc., CA, USA), el cual presenta un ruido del tipo acumulativo, toma alrededor de 20 minutos debido a que es necesario mover el objeto a medir para que el láser pueda abarcar toda la superficie. Cabe resaltar que solo en el Kinect se adquirieron muestras a distintas distancias, pues se quiere ver cómo influye esta variación.



Figura 2.7 Escáner NextEngine Laser Scanner HD.

Como en el caso anterior, se convierte a nube de puntos la imagen de rango de la maqueta en su totalidad, fue necesario segmentar la maqueta de todo lo demás, como se puede ver en la Figura 2.8 (a) y (b) respectivamente.

Para comparar las medidas de ambos sensores, se restaron las nubes de puntos. Antes de hacer esta operación, es imprescindible que las nubes de puntos estén alineadas, es decir, tengan el mismo sistema de coordenadas. Se emplearon dos softwares libres para realizar esta tarea, el primero de ellos MeshLab [20] sirve para visualizar mejor las nubes de puntos, como se puede ver en la Figura 2.9 (a), y estimar la alineación de ambas nubes seleccionando manualmente los puntos de control (ver Figura 2.9 (b)). Además, se utilizó el algoritmo iterativo de puntos cercanos (ICP por sus siglas en inglés) que se encuentra implementado en el software, con el fin de optimizar la alineación manual inicial. Este programa reconoce archivos de nubes de puntos que estén en el formato Polygon File Format (PLY) para lo cual se procedió a almacenar las nubes de puntos con esta extensión.

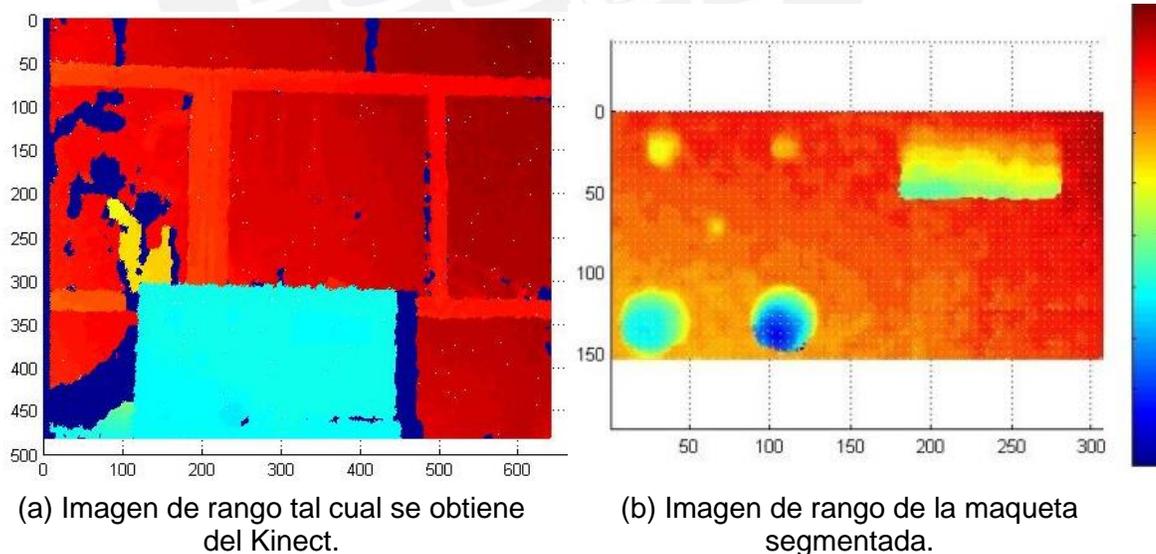
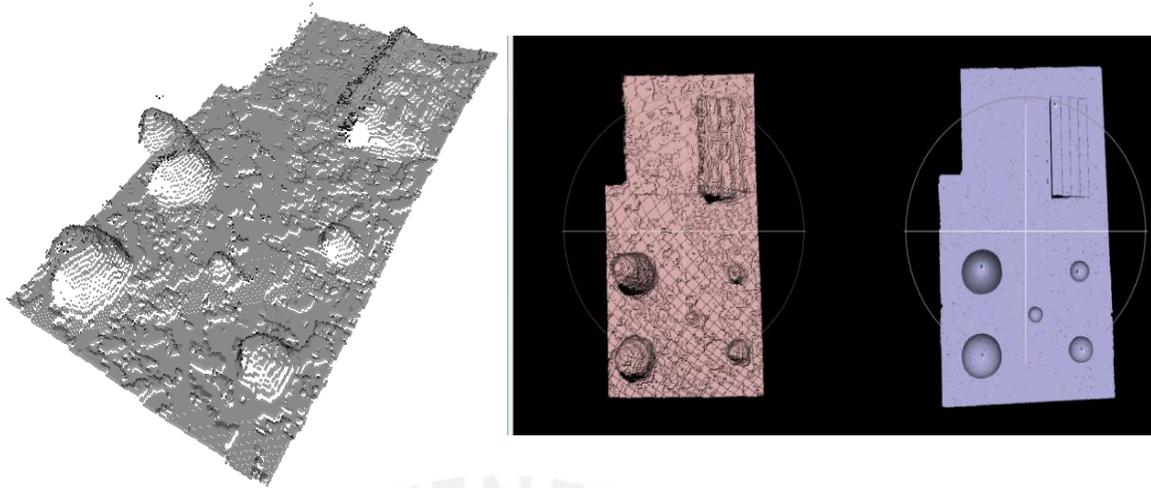


Figura 2.8 En (a) podemos ver el resultado de la adquisición de datos y luego en (b) la misma imagen con la maqueta segmentada. Las unidades de esta última están en milímetros.



(a) Nube de puntos de la maqueta. (b) Interfaz para la selección de puntos de control entre la nube de puntos del Kinect y del NextEngine Laser Scanner HD.

Figura 2.9 La transformación de la imagen de rango a nube de puntos se ve en (a). Por otro lado, (b) se realiza la identificación de puntos de control manual entre la nube de puntos procedente del Kinect (izquierda) y del *NextEngine Laser Scanner HD* (derecha).

Posteriormente, se empleó el programa CloudCompare en el cual existe un comando directo para restar dos nubes de puntos. De esta manera, se obtuvo una nube de puntos colorizada donde el color representa el error. Además, se generó un histograma de dicho error. Tomando en cuenta el teorema de límite central, los valores de profundidad de cada punto son independientes entre sí y provienen de una misma fuente de ruido, de manera que es posible aplicar un ajuste Gaussiano. Los resultados se detallan a continuación en la Figura 2.10, en donde se aprecian tres casos posibles, la distancia más cercana, intermedia y lejana de todo el rango de medición (0.85m a 1.35m).

Finalmente, en la Figura 2.11 podemos ver la media de cada error para cada distancia, así como su desviación. Notamos que el error de exactitud tiene un comportamiento constante conforme se aleja el dispositivo Kinect del objeto a medir. De estos datos podemos concluir que bajo las condiciones ya señaladas, el error de exactitud es de 6.3 milímetros.

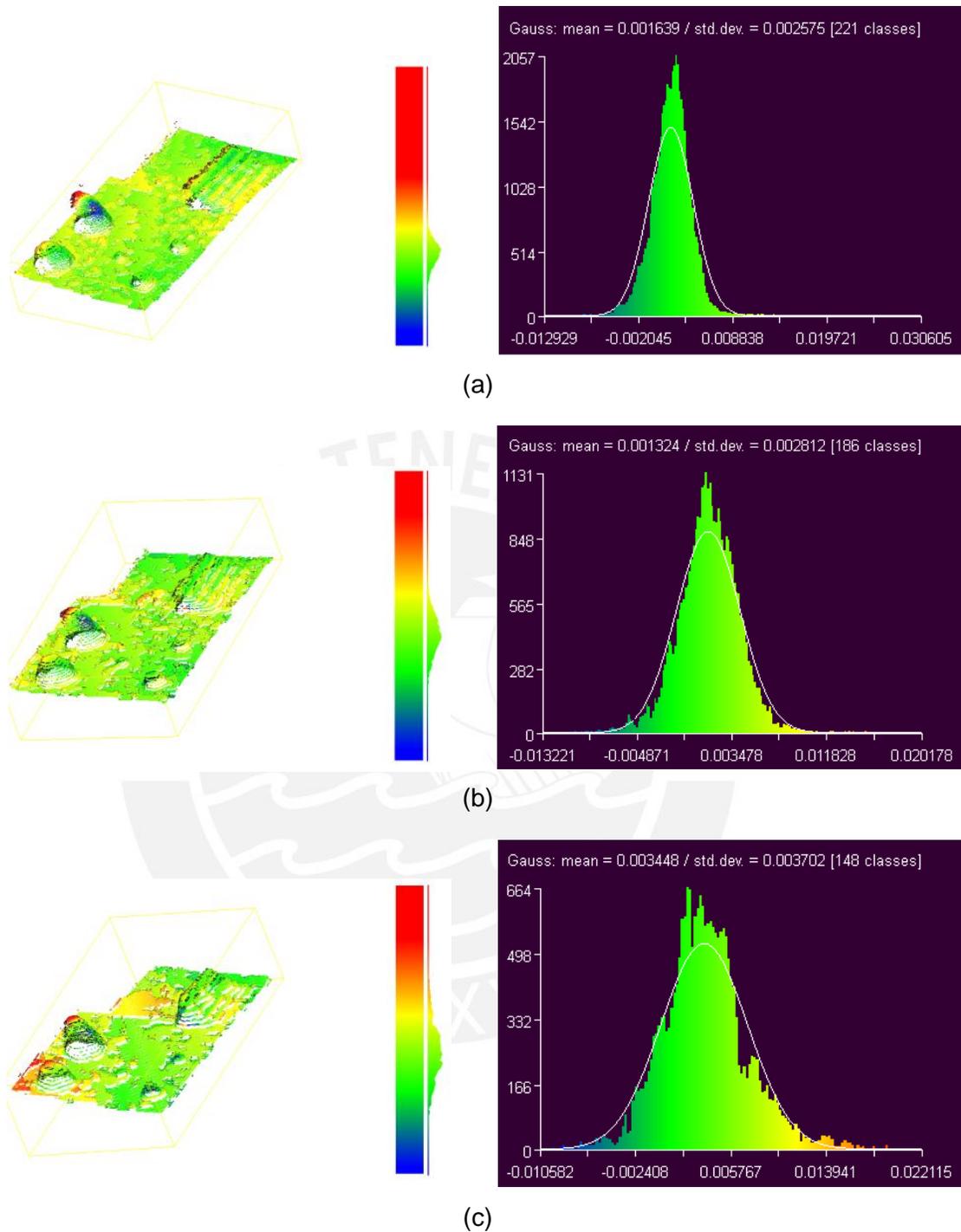


Figura 2.10 En la primera columna, vemos el resultado de la diferencia de las nubes de puntos provenientes del Kinect y del *NextEngine Laser Scanner HD* representada por una nube colorizada. En la segunda columna vemos los histogramas respectivos de los errores con el ajuste Gaussiano. Los pares de figuras (a), (b) y (c) representan las medidas a las distancias de 0.85, 1.10 y 1.35 metros respectivamente

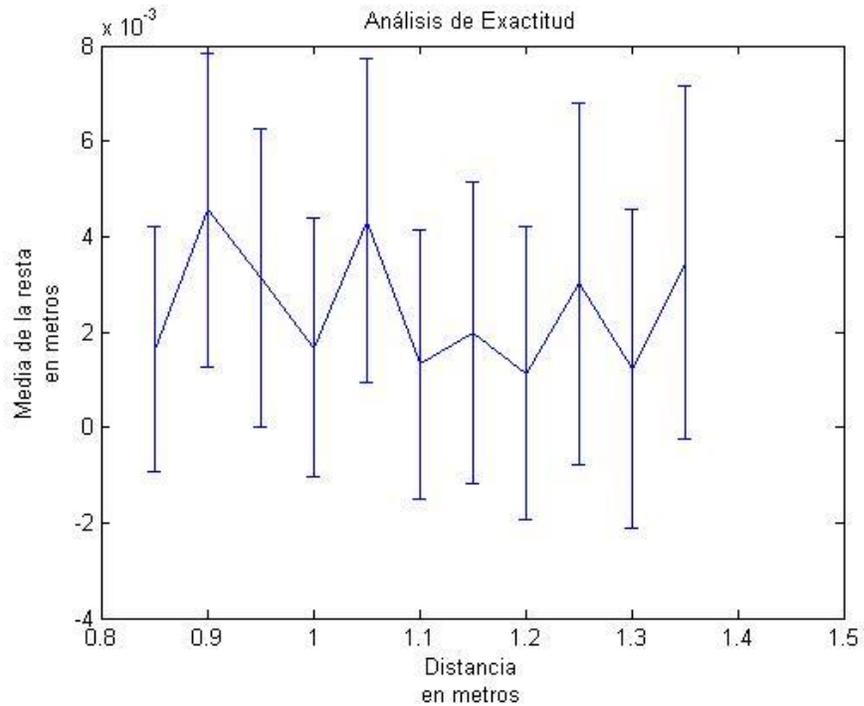


Figura 2.11 Gráfica de La media de la diferencia de las nubes de puntos versus la distancia a la que el Kinect tomó las muestras en donde figuran también las desviaciones de cada media.

### 2.3.4 Análisis de resultados

La comparación de los parámetros obtenidos del Kinect for XBOX 360, los del *NextEngine Laser Scanner HD* obtenidos en [11] y los resultados de un análisis de fotogrametría en una estructura arqueológica evaluado en [23] se pueden ver a continuación en la Tabla 2.

Tabla 2 Comparación de los parámetros de precisión y exactitud de tres técnicas: Kinect for XBOX, *NextEngine Laser Scanner HD* [11] y fotogrametría [23].

	Kinect for XBOX	NextEngine Laser Scanner HD*	Fotogrametría
<b>Error de precisión (mm)</b>	2.2	0.107	16
<b>Error de exactitud (mm)</b>	6.3	0.127	4

Dentro de los experimentos realizados en [11] se caracteriza al escáner láser de triangulación y se obtienen los parámetros ya señalados. Por otro lado, en [23] se establecen como parámetros aceptables, para el ámbito de arqueología, los valores señalados en su columna respectiva de la tabla. Estos valores son encontrados luego de elaborar diversas reconstrucciones con la técnica de fotogrametría.

En base a estos resultados podemos concluir que el Kinect for XBOX 360 es apto para aplicaciones en arqueología a un rango de distancia de 0.85 a 1.35 metros a condiciones de luz media, es decir con poca presencia de luz solar y ligera presencia de luz artificial.



## Capítulo 3: Desarrollo de Algoritmo de Registro de Nubes de Puntos

### 3.1 Metodología del diseño

El presente algoritmo se desarrolla de acuerdo a la metodología planteada en la Figura 3.1. En la primera etapa se realiza la adquisición de muestras del monumento arqueológico Camino Inca situado dentro de la Pontificia Universidad Católica del Perú. Estas muestras consistirán en las imágenes de profundidad y las imágenes a color que el Kinect brinda.

En la segunda etapa se comienza a trabajar con las muestras como nubes de puntos. Se evaluará la calidad de las mismas como la cantidad de puntos que contiene y se verá la variación de las mismas a lo largo de todas las muestras que se tomen. También se analizará la relación entre las imágenes de profundidad y color con la intención de verificar la utilidad de las dos para el registro y se eliminarán los puntos que contienen información incorrecta debido a las limitaciones de distancia del Kinect.

Finalmente, en la última etapa se definirá la manera en que se trabajaran todas las muestras así como los algoritmos indicados para encontrar los puntos de control entre estas. Se verá la factibilidad de emplear las imágenes de color en conjunto con las de profundidad para elaborar un mejor registro. Luego de tener un registro completo con todas las muestras en nube de puntos, se procederá a elaborar una malla del resultado y finalmente se exportará el modelo tridimensional con color para poder ser comparado con otros modelos.



Figura 3.1 Esquema metodológico del algoritmo.

### 3.2 Adquisición de las muestras empleando el Kinect

La adquisición de las muestras fue posible gracias a la colaboración con el Laboratorio de Imágenes Médicas de la Pontificia Universidad Católica del Perú quienes facilitaron el uso del Kinect for XBOX y su trípode para todo el desarrollo de la investigación.

#### Materiales

- Equipo Kinect for XBOX.
- Software MATLAB[19] versión 2013a con la extensión *Microsoft Kinect for Windows Support from Image Acquisition Toolbox™* [24].
- Computadora Core(TM) i5 CPU M 460 2.53GHz, 4GB de Memoria RAM, USB 2.0, SO Windows 7 Home Premium, 64 bits, con tarjeta gráfica NVIDIA GeForce GT 420M, 1GHz.
- Flexómetro de 3 metros.
- Trípode RocketFish Gaming® para Kinect.
- Extensión eléctrica de 15 metros.
- Medidor digital de Luz, modelo SLM – 110 A.W. Sperry Instruments, INC.

#### Parámetros y consideraciones

Como se indicó en el inciso 2.2 del Capítulo 2, se seleccionó un día con poco brillo solar, totalmente nublado, para que la adquisición de datos no se vea afectada por la luz natural. No obstante, las muestras se tomaron pocas horas después de mediodía dado que es necesaria cierta luz para que las imágenes de color tengan buena calidad, entre 500 lux y 580 lux. Adicionalmente, como se vio en el inciso 2.3 del mismo capítulo, el Kinect es apto para mediciones en el rubro de muestras arqueológicas en el rango de 0.85 metros a 1.35 metros. Por lo tanto, se ubicó el Kinect a una distancia de un metro de las estructuras a medir.

### Montaje del sistema de adquisición

El montaje del sistema consiste en la habilitación del Kinect controlado por la computadora a través del software MATLAB. Así mismo, se coloca el Kinect a la distancia señalada previamente.

### Procedimientos

- Se escoge una porción del Camino Inca para la adquisición de datos para lugar ubicar al Kinect a 1 metro de distancia del mismo (ver Figura 3.2 (a)) y se sitúa el Kinect en el trípode extendido al máximo, lo cual lleva al Kinect a una altura de un metro sobre el suelo.
- Se enciende el Kinect y se inicia el software MATLAB en la computadora. Se reconoce el Kinect como una cámara RGB y un sensor de profundidad dentro del software. El software desarrollado permite adquirir las muestras de manera automática o manual. La primera de estas consiste en la captura de una ráfaga determinada de muestras mientras que en la otra es necesario presionar una tecla para capturar la siguiente muestra. Para el modo automático de adquisición, que desde ahora llamaremos modo de adquisición 1, se tomaron 200 muestras y para el manual, modo de adquisición 2, 45 muestras.
- Antes de inicializar la adquisición, una persona coge el Kinect y será quien se trasladé con él a lo largo de la porción del objeto de estudios seleccionado. Debido a que el Kinect ya se encuentra en el trípode, solo será necesario elevarlo lo suficiente para que las patas del mismo no golpeen el suelo. Una vez posicionado, otra persona da inicio a la captura de datos y la otra se desplaza horizontalmente. Para el modo de adquisición 1 se desplazará a una velocidad promedio de un metro por segundo, tratando de no mover al Kinect verticalmente, capturando una muestra cada 0.1 segundo mientras que para el modo de adquisición 2 se moverá dando pasos de aproximadamente medio metro (ver Figura 3.2 (b)). De esta manera, al tomar los datos de manera manual se terminaría la adquisición en tres minutos, tiempo máximo para las adquisiciones manuales y automáticas.
- Terminada la toma de datos, las muestras tanto de color como de profundidad se verificarán visualmente la calidad de las tomas y corroborar que no hubo



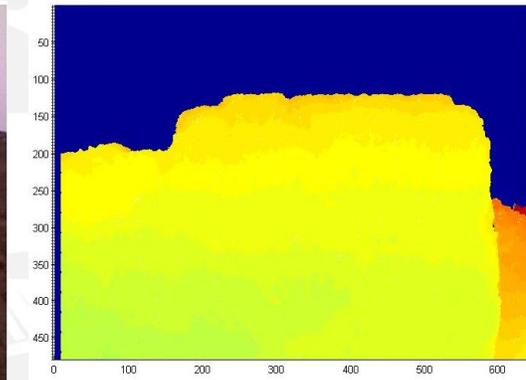
(a) Kinect posicionado en el trípode.



(b) La flecha roja indica la dirección en la cual se trasladará el Kinect.



(c) Imagen de color capturada con el Kinect.



(d) Imagen de profundidad capturada con el Kinect.

Figura 3.2 En (a) vemos el montaje del Kinect para iniciar la adquisición de datos. En (b) la dirección en la cual una persona moverá el Kinect durante la adquisición. Por otro lado, en (c) y (d) observamos una de las muestras obtenidas con el Kinect, de color y de profundidad respectivamente.

movimientos verticales bruscos (ver Figura 3.3 (c) y (d) respectivamente). Si este no es el caso, se procederá a guardar la data de las imágenes de color y de profundidad, las cuales se encuentran en variables distintas.

### 3.3 Transformación de las muestras a nubes de puntos

Para las muestras obtenidas del modo de adquisición 1, la velocidad de adquisición del Kinect es tan alta que algunas muestras consecutivas no evidencian variación a simple vista. Es por esto que se disminuirá el número de muestras escogiendo una

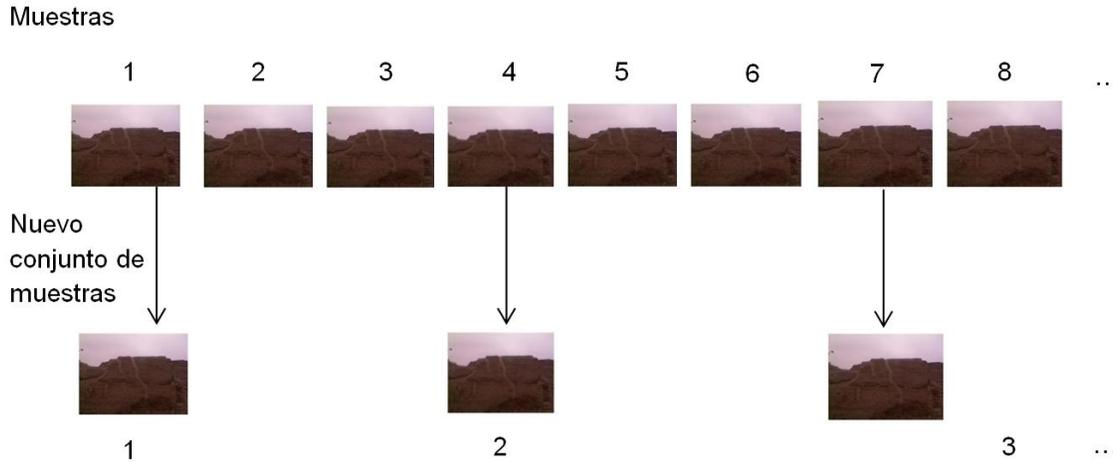


Figura 3.3 Se procede a formar un nuevo conjunto de muestras escogiendo la actual y descartando las dos siguientes se hace el mismo procedimiento con las imágenes de profundidad.

muestra y descartando las dos siguientes (ver Figura 3.3), este valor fue escogido para debido a que a partir de la cuarta muestra se aprecia un cambio entre muestras, de manera visual. Cabe resaltar que esto se hizo para las imágenes de color y de profundidad.



(a) Nube de puntos exportada sin color. (b) Nube de puntos exportada con color.

Figura 3.4 Nubes de puntos sin color (a) y con color (b) en formato PLY visualizadas en CloudCompare.

Las imágenes de profundidad obtenidas del Kinect vienen en formato de imagen de rango y no se es posible trabajar con ellas correctamente. Es por esto que se procede a realizar la transformación a nubes de puntos en formato XYZ. Este formato consiste en una lista de coordenadas cartesianas de los puntos que conforman la nube de puntos del modelo 3D. Consta de tres columnas (X, Y y Z) y el número de filas será la cantidad de puntos en la nube (640x480x3). Cabe resaltar que a la imagen de color se le aplica el mismo formato (RGB), para que siga existiendo una relación directa entre el punto y el color que lleva. En este caso las columnas indican la cantidad de cada color que tendrá cada punto.

El siguiente paso será eliminar los puntos que no contienen información. Como se ve en la Figura 3.3 (d), hay puntos que escapan del rango del Kinect (representados por el color azul) y no será posible hacer operaciones con la nube de puntos dado que tiene elementos de valor *NaN* (valor que toman las tres coordenadas de la nube de puntos al ser una medida que escapa del rango del dispositivo). Se encuentran los índices en los cuales existe este valor y se eliminan, no obstante para no dejar un vacío en la nube, automáticamente los puntos sucesivos suben su índice, disminuyendo así el tamaño de la nube de puntos. Los índices encontrados que poseen un valor *NaN* también son eliminados de la imagen de color para que siga existiendo una relación directa entre la posición del punto y su color.

Finalmente, para tener una visualización más adecuada del modelo, se procede a exportar la nube en formato PLY para visualizarlas con el software CloudCompare, tal como se hizo en los incisos 2.3.1 y 2.3.2 del Capítulo 2, el resultado de se puede ver a continuación en la Figura 3.4(a) y (b).

### **3.4 Registro rígido de nubes de puntos**

#### **3.4.1 Arquitectura del método de registro**

A continuación se presenta, en la Figura 3.5, un diagrama más detallado que el de la Figura 3.1, enfatizando el proceso de registro. Los procesos de color verde son los ya descritos hasta el inciso anterior.

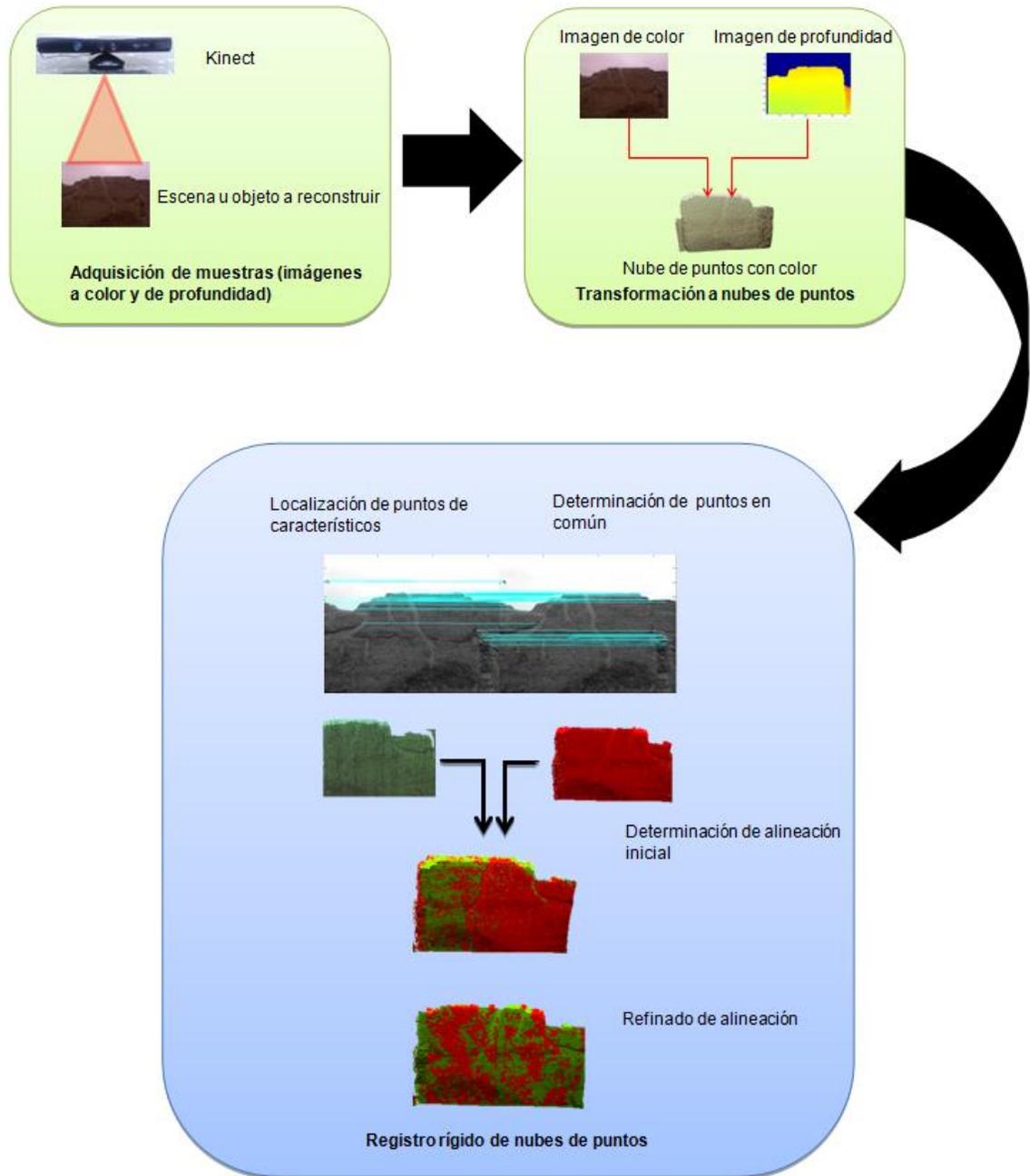


Figura 3.5 Arquitectura del sistema en su totalidad, en azul el proceso de registro.

### **3.4.1.1 Localización de puntos característicos y determinación de puntos en común**

En esta primera etapa del registro, se encontrarán los puntos característicos de cada muestra o de puntos de interés. Estos son únicos de manera global, es decir, aunque la imagen sea trasladada, rotada o sufra un cambio de escala, estos se podrán seguir identificando. Estos puntos son necesarios para poder realizar una primera alineación de las muestras, de manera que el problema recaería en trasladar una muestra hacia otra, logrando que sus puntos en común coincidan en ubicación. A cuantos más puntos en común se tengan, la alineación será más precisa.

El algoritmo más utilizado para este labor es SIFT (Scalar Invariant Feature Transform) [25]. Este se encarga de determinar los puntos característicos o de interés de dos imágenes basándose en las imágenes de color en escala de grises. Las tres partes principales de este algoritmo son la detección, descripción y relación.

En la detección, se buscan identificar todos pixeles o puntos candidatos que satisfagan lo ya mencionado líneas arriba. Estos son identificados los máximos y mínimos del resultado de una función de diferencia gaussiana (ecuación 3.1) aplicada a la misma imagen a distintas escalas y suavizada [25].

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \quad (3.1)$$

En la descripción, se busca asignar las orientaciones dominantes a los puntos hallados previamente, de modo que sean más estables su reconocimiento y su relación con los puntos de otra imagen.

Finalmente, en la etapa de relación, mediante diversos algoritmos, intenta encontrar puntos de características similares, en la otra imagen, en un vecindario cercano alrededor de la posición del punto en la imagen original.

A pesar de su amplio uso en el ámbito de visión por computadora y reconstrucción tridimensional y un tiempo de procesamiento relativamente rápido [26], se optó por una variante de este algoritmo.

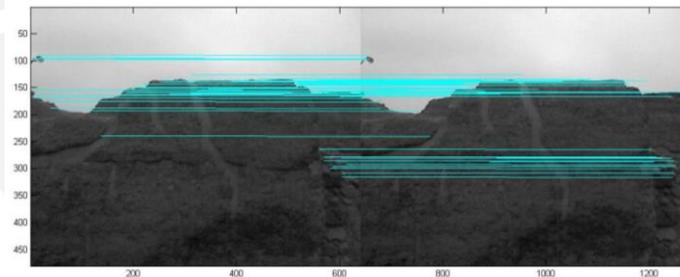
Esta modificación de SIFT denominada SURF (Speeded-Up Robust Features) se diferencia básicamente en que optimiza el último proceso de relación de puntos con la

intención de mejorar el tiempo de procesamiento total sin sacrificar significativamente la eficiencia [26], logrando así en muchas ocasiones reducir dicho parámetro de tiempo a la mitad [27].

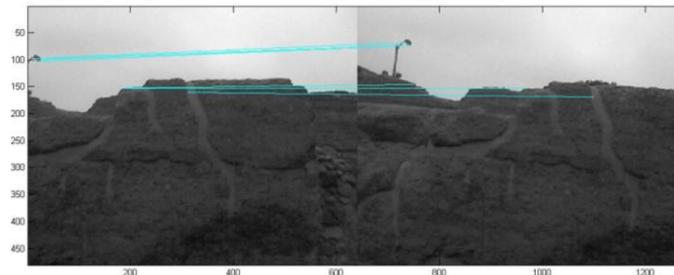
Antes de aplicar el algoritmo SURF [26], definiremos como referencia la primera muestra, es decir, todas las demás serán comparadas con esta y al final, será respecto a ella con quien se alineen. Cabe resaltar que se trabajaron con las imágenes en escala de grises dado que es así que trabaja el algoritmo.

Al ejecutar el algoritmo, se obtienen, en promedio, 175 puntos característicos por cada muestra. No obstante, al determinar la relación entre los puntos de dos muestras consecutivas, por ejemplo la primera y la segunda, se obtienen 55 puntos en común (ver Figura 3.6 (a)).

Por otro lado, al tener como referencia la primera muestra, conforme nos alejamos estos puntos en común van disminuyendo dado que cada vez tienen menos zonas comunes en las fotos, de manera que entre la primera y la última muestra existen solo 6 puntos en común (ver Figura 3.6 (b)).



(a) Visualización de puntos en común entre dos muestras consecutivas.



(b) Visualización de puntos en común entre la primera y la última muestra.

Figura 3.6 Puntos en común de dos muestras unidos por la línea celeste. En (a) existen 55 puntos en común mientras que en (b) solo 6.

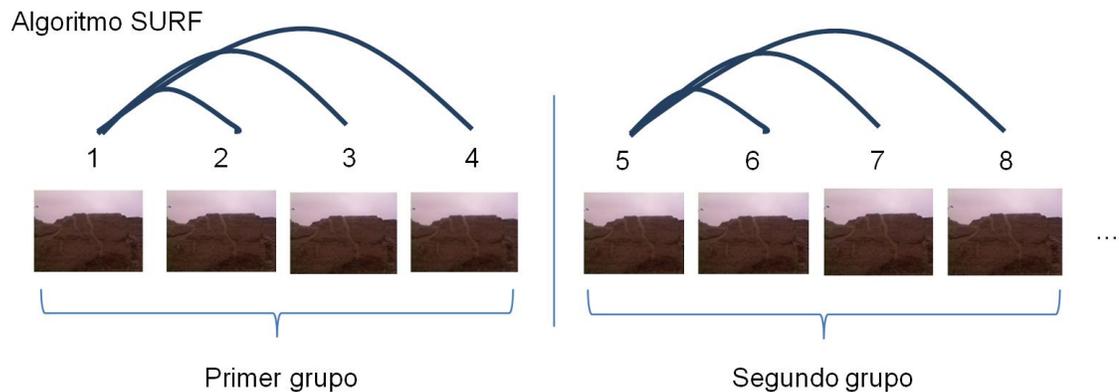


Figura 3.7 Método con el cual se empleó el algoritmo SURF. Todas las muestras de cada grupo son comparadas con la primera del mismo grupo para no disminuir el número de puntos en común. El tamaño del grupo es de 4 para este ejemplo.

Para solucionar esta baja cantidad de puntos en común, la cual haría muy complicada una buena alineación, se planteó dividir las muestras en pequeños grupos. De esta manera, se tomará como referencia la primera muestra de cada grupo y las restantes del mismo serán comparadas con esta.

Finalmente, se determinó que el tamaño del grupo sería de cinco muestras. Al correr el algoritmo SURF entre la primera y la última muestra de cada grupo, obtenemos en promedio 45 puntos en común, evidenciando que será posible una buena alineación. El proceso se detalla a continuación en la Figura 3.7 para un grupo de cuatro muestras, la metodología se mantiene.

#### **3.4.1.2 Determinación de alineación inicial**

El segundo paso en el proceso de registro consiste en usar la información obtenida hasta ahora (puntos en común entre las muestras) para poder alinearlas. Debido a que se compararon las muestras en cada grupo, la alineación será solo entre los elementos de cada uno de ellos. Esto quiere decir que las muestras del primero y segundo grupo estarán alineadas respecto a las muestras de cada una de ellas pero no necesariamente respecto a las de otro grupo.

Sabemos que los puntos en común fueron hallados en base a las muestras de color y que estos tienen una correspondencia directa con la información de profundidad ya transformada a nubes de puntos.

Sabemos que la información de las nubes de puntos está almacenada en forma matricial y que, al estar ubicada en el espacio, para que un elemento de una nube de puntos coincida con otro elemento de otra nube de puntos, bastará con que se rote y/o traslade, hasta que tengan las mismas coordenadas cartesianas. A esto se le denomina registro rígido, pues bastará de un solo parámetro común para que todo un grupo de elementos (todos los puntos de la nube) se ubique en otra posición deseada.

Conforme a lo explicado y de acuerdo con [9], será necesario encontrar una matriz de rotación  $R$  y un vector de traslación  $T$  para llevar un elemento de su posición original a la posición deseada. Esto quiere decir que, teniendo dos puntos  $p_i'$  y  $p_i$  comunes entre dos muestras donde  $i$  representa el índice del punto en la nube,  $p_j$  deberá ser rotado y trasladado para que tenga la misma posición espacial que  $p_i'$ , siendo este la referencia (ecuación 3.2).

$$p_i' = Rp_i + T \quad (3.2)$$

Para llevar a cabo esta operación, los puntos  $p$  deberán tener el formato de matriz columna, tres filas (X, Y y Z) y una columna (índice del punto dentro de la matriz). Para esto será necesario tener una nueva matriz  $p$  a la matriz traspuesta de la nube de puntos expresada en el inciso 3.3 de este capítulo. De esta manera,  $R$  es una matriz 3x3 y  $T$  una matriz de tres filas y una columna.

Para encontrar los parámetros  $R$  y  $T$  nos basamos en el trabajo desarrollado en [9] en donde se busca la matriz de rotación y vector de traslación minimizando una diferencia cuadrada (ecuación 3.3).

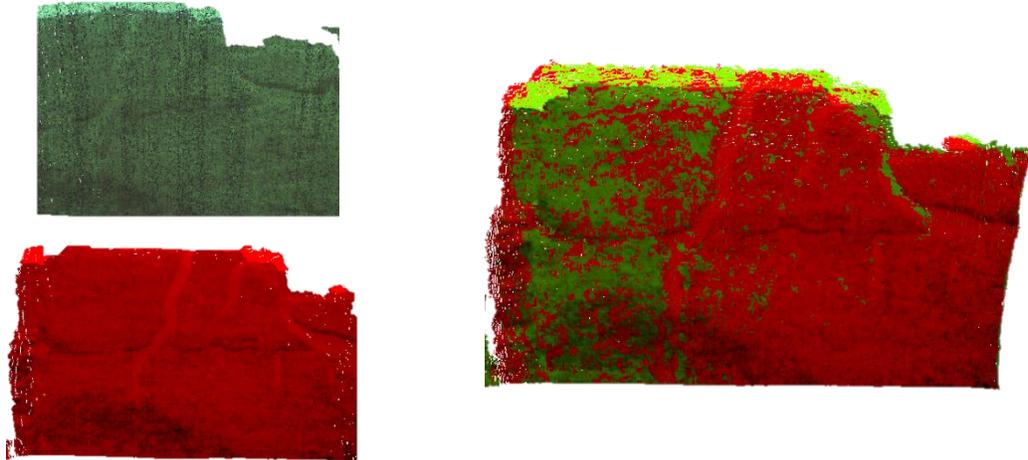
$$\Sigma^2 = \sum_{i=1}^N \|q_i' - Rq_i\|^2 \quad (3.3)$$

$$q_i \triangleq p_i - p \quad (3.4)$$

$$q_i' \triangleq p_i' - p' \quad (3.5)$$

Luego, el trabajo reincide en encontrar  $R$  que minimice la ecuación 3.3 y luego hallar  $T$  de manera directa de la relación 3.6.

$$T = p' - Rp \quad (3.6)$$



(a) La nube de puntos verde es la referencia y la roja se traslada y rota.

(b) Nubes de puntos ya alineadas.

Figura 3.8 Las nubes de puntos de (a) serán alineadas mediante las matrices  $R$  y  $T$  obtenidas luego de aplicar SURF. El resultado se puede ver en (b) en donde la parte manchada (mezcla de colores) representa las nubes alineadas en esa región.

Una manera de hallar  $R$  consiste en basarse en el algoritmo de descomposición de valor singular (SVD por sus siglas en inglés) el cual consiste en descomponer una matriz en dos factores que satisfacen ciertas características. Para más información en el desarrollo de este algoritmo ver el artículo [9].

Una vez hallado estos dos parámetros en base a los puntos en común entre muestras, se procede a ejecutar la ecuación 3.2 pero con toda la nube de puntos. En la Figura 3.8 (a) podemos ver dos nubes de puntos que serán alineadas donde la verde es la referencia y en (b) la nube de puntos roja rotada y trasladada con respecto a la verde.

### 3.4.1.3 Refinado de Alineación

Hasta este momento se tienen las muestras de cada grupo alineadas. Debido a que el la matriz  $p$  de la nube de puntos ya ha sido modificada al haber sido alineada, ya no existe una relación directa entre la imagen de color y la información de profundidad. Esto quiere decir que si intentamos alinear la primera muestra del segundo grupo con la última muestra del primer grupo mediante SURF, se podrán encontrar puntos en común y se hallará una matriz de rotación y vector de traslación. No obstante, esta alineación será con la posición inicial que tuvo la matriz  $p_{fin}$  que ahora se encuentra en una posición  $p_{fin}'$  al haber sido alineada con la primera muestra de su grupo. Para

resolver este problema, se almacenó el valor de  $R$  y  $T$  encontrados en el inciso anterior, pero solo de la última muestra de cada grupo. De esta manera se tendrán  $R_{fin1}$  y  $T_{fin1}$ ,  $R_{fin2}$  y  $T_{fin2}$  hasta la penúltima muestra del último grupo, dado que no será necesario almacenar datos de la última muestra pues no será alineada con ninguna otra más.

De este modo, si se alineasen los límites de dos grupos, la nueva alineación de la muestra inicial de un grupo se debería alinear aplicando las matrices  $R_{fin}$  y  $T_{fin}$ . Solo así esta muestra se alineará con la posición actual a la que se movió la muestra final del grupo anterior. Las ecuaciones 3.7 y 3.8 detallan lo mencionado en el anterior párrafo, donde  $R_{inicio-fin}$  y  $T_{inicio-fin}$  son los parámetros obtenidos tras aplicar SURF entre la muestra final de un grupo con la inicial del siguiente.

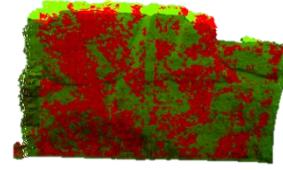
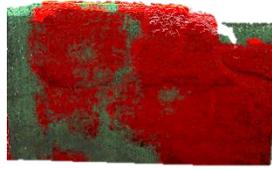
$$p_{inicial}' = R_{inicio-fin}p_{inicial} + T_{inicio-fin} \quad (3.7)$$

$$p_{inicial}'' = R_{fin}p_{inicial}' + T_{fin} \quad (3.8)$$

Por otro lado, la alineación obtenida hasta el momento ha intentado trasladar una muestra a la posición de la otra de manera que sus puntos en común coincidan. No obstante las nubes de puntos todavía pueden acercarse más entre sí para mejorar el registro. Para lo siguiente, se decidió implementar el algoritmo ICP (Iterative Closest Point) el cual consiste en minimizar la distancia entre dos nubes de puntos para mejorar el registro.

Este algoritmo se encarga de alinear iterativamente dos nubes de puntos reduciendo la distancia que separa dos puntos muy próximos, reduciendo la distancia euclidiana que las separa en cada iteración [28]. Debido a que es un algoritmo iterativo. Se puede ejecutar un número determinado de veces o hasta que se encuentre una separación mínima deseada. Para este caso se decidió ejecutar el algoritmo 100 veces pues coincidía con un error mínimo de 0.5 milímetros.

Si las muestras no se encuentran alineadas, este algoritmo solo acercará dos nubes en la posición que estén. A continuación se aprecia en la Figura 3.9 (a) el algoritmo usado entre dos nubes no alineadas y en (b) entre nubes alineadas previamente por SURF.



(a) Nubes luego de aplicar ICP sin previa alineación. (b) Nubes luego de aplicar ICP con previa alineación.

Figura 3.9 En (a) se ve como hay un desplazamiento de la nube roja mientras que en (b) se aprecia gran traslape de ambas.

El algoritmo de ICP es óptimo cuando las nubes de puntos ya tienen una alineación previa pues todos los puntos de la región en común que comparten ya se encuentran relativamente cerca. Es por esto que aplicamos el algoritmo entre las nubes de puntos de cada grupo previamente alineados tras usar SURF. Para mejorar el registro, ICP se aplica de manera combinatoria entre todas las muestras del mismo grupo para cada grupo (ver Figura 3.10). Como resultado, tendremos grupos de cinco muestras muy bien alineadas entre sí lo cual nos lleva al último paso, alinear los grupos.

Hasta este punto sabemos que cada matriz  $p$  que representa una muestra ha sufrido dos modificaciones: La primera al ser alineada mediante el algoritmo SURF con la primera muestra de su respectivo grupo y la segunda luego de ser alineada de manera combinatoria empleando el algoritmo ICP con todas las muestras de su respectivo grupo.

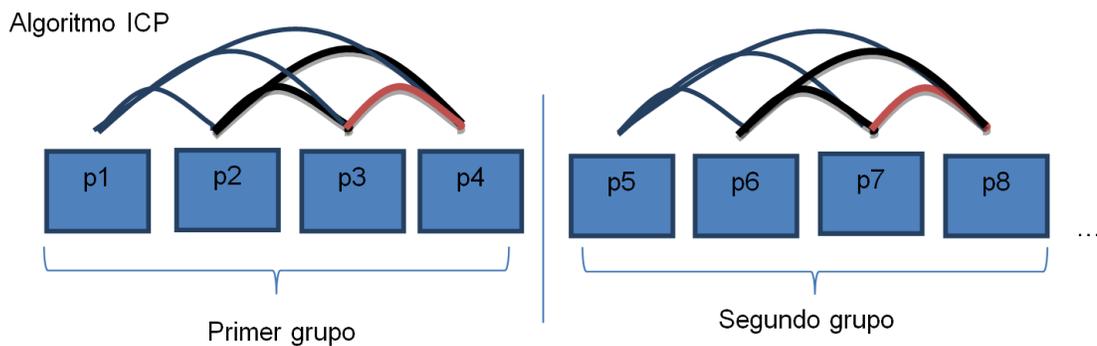


Figura 3.10 Se aplica el algoritmo ICP de manera combinatoria para cada grupo mejorando así la alineación en cada grupo. Se toma como ejemplo un grupo de 4 muestras.

El siguiente paso consiste en encontrar las matrices  $R$  y vectores  $T$  que determinan la conexión entre cada grupo. Como se mencionó al inicio de este inciso, SURF solo relaciona las muestras de color y como hasta el momento las nubes de puntos se han modificado en tres ocasiones, llevar a cabo la alineación previa puede parecer poco útil. No obstante, aplicar ICP directamente no conllevará a un buen registro dado que los grupos nunca fueron alineados entre sí. Por consecuencia, ICP solo podrá ser aplicado una vez que la muestra inicial de un grupo haya sido alineada mediante SURF con la muestra final del grupo anterior y además multiplicada por todas las matrices  $R$  y  $T$  que han afectado dicha muestra. Esto involucra  $R$  y  $T$  producto del SURF con la primera muestra del grupo y  $R$  y  $T$  producto del ICP combinatorio. A continuación se detalla el flujo que experimentan las matrices, incluidas las nubes de puntos hasta llevar a cabo el último ICP. Las matrices  $p_{final}$  y  $p_{inicial}$  representan las nubes de puntos iniciales con relación directa a sus muestras de color de la última muestra de un grupo con la primera del segundo grupo respectivamente.

$$p_{final} = R_{surf}p_{inicial} + T_{surf} \quad (3.9)$$

$$p_{final}' = R_{ICPcombinatorio}p_{final} + T_{ICPcombinatorio} \quad (3.10)$$

$$p_{final}'' = R_{SURFinicio-fin}p_{final}' + T_{SURFinicio-fin} \quad (3.11)$$

$$p_{final}''' = R_{ICPinicio-fin}p_{final}'' + T_{ICPinicio-fin} \quad (3.11)$$

Como en realidad el valor de la nube de puntos se va actualizando en cada cálculo, las nubes de puntos luego de la igualdad pueden pasar a la izquierda y tendremos una expresión general para  $R$  y  $T$  que conectan los grupos (ecuación 3.12 y 3.13)

$$R_{conexión} = R_{ICPinicio-fin}\{[R_{ICPcombinatorio}(R_{surf} + T_{surf}) + T_{ICPcombinatorio}] + T_{SURFinicio-fin}\} + T_{ICPinicio-fin} \quad (3.12)$$

$$T_{conexión} = T_{surf} + T_{ICPcombinatorio} + T_{SURFinicio-fin} + T_{ICPinicio-fin} \quad (3.13)$$

De estas ecuaciones podemos concluir una fórmula general para los  $R$  y  $T$  que involucran diferentes  $R$  y  $T$  previas, (ecuación 3.14, 3.15 y 3.16).

$$R_n = R_n(R_{n-1} + T_{n-1}) + T_n \quad (3.14)$$

$$R_n = R_n R_{n-1} + R_n T_{n-1} + T_n \quad (3.15)$$

$$T_n = T_n + T_{n-1} \quad (3.16)$$

Cabe resaltar que el  $R$  y  $T$  producto del ICP combinatorio se debe calcular conforme a las ecuaciones 3.15 y 3.16 ya que la muestra se modifica en más de una ocasión.

Finalmente, el orden en cómo se aplicarán las matrices es el siguiente:

- Se actualizan los valores de las nubes de puntos con las matrices  $R$  y  $T$  de SURFT encontradas entre la primera muestra del grupo  $R$  y  $T$  y el del ICP combinatorio.
- A excepción de la primera muestra de cada grupo, las muestras de un grupo serán multiplicada por el  $R$  y  $T$  ( $n$ ) de un nuevo ICP con la muestra anterior, dado que ha sido modificada, y el  $R$  y  $T$  ( $n-1$ ) que se ha venido acumulando. La primera muestra solo será multiplicada por el  $R$  y  $T$  de conexión al grupo anterior.
- Este proceso se repite para cada grupo. En cada uno de ellos existirá un  $R$  y  $T$  de conexión distintos al grupo anterior y la acumulación de  $R$  y  $T$  sólo se da dentro de cada grupo.

Cabe resaltar que para el primer grupo, las muestras solo se verán afectadas por el  $R$  y  $T$  de SURF y su ICP combinatorio. En la Figura 3.11 se detalla el proceso definitivo.

Previamente modificadas por SURF e ICP

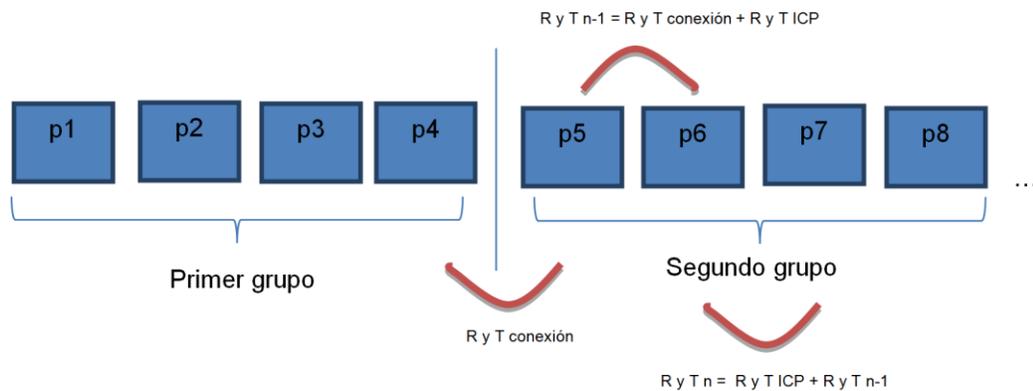


Figura 3.11 Proceso acumulativo de operar las nubes de puntos con las  $R$  y  $T$  involucradas. El método de operación a seguir es el descrito por la ecuación 3.15 y 3.16

### 3.5 Creación de mallas partir de las nubes de puntos

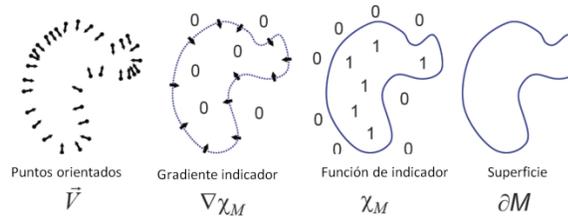


Figura 3.12 Ilustración intuitiva de la reconstrucción de Poisson. Adaptado de [30].

Como paso adicional, para poder hacer un análisis cuantitativo y cualitativo del registro, se procedió a exportar las nubes de puntos ya registradas, es decir las matrices  $p$  ya modificadas exportando también sus vectores normales. Estos se calcularon como vectores normales a los polígonos formados producto de la triangulación de Delaunay [29].

Una vez exportadas se empleó dentro del software MeshLab el algoritmo de reconstrucción de Poisson que viene incorporado. Este algoritmo se basa en resolver la ecuación 3.17, en la cual al tener un campo vectorial  $V$ , se debe hallar una función  $X$  tal que la gradiente de esta función sea  $V$  [30].

$$\Delta \tilde{X} = \nabla \cdot \vec{V} \quad (3.17)$$

$$\Delta X \equiv \nabla \cdot \Delta X = \nabla \cdot \vec{V} \quad (3.18)$$

Este campo vectorial estaría conformado por los vectores normales de las facetas de la nube de puntos. En la Figura 3.12 podemos ver los significados de las variables de las ecuaciones de Poisson.

### 3.6 Discusión y sugerencias

A lo largo del capítulo se presentó el algoritmo elaborado para la reconstrucción. Se realizaron registros de dos casos del Camino Inca, en el primero de ellos las muestras fueron adquiridas mediante los dos modos de adquisición, mientras que en el segundo caso solo se empleó el modo de adquisición 2. Se recomienda trabajar con una computadora de mejores características pues tanto el registro como la reconstrucción requieren de un alto costo computacional.

## Capítulo 4: Análisis de Resultados

### 4.1 Visualización de registro y malla

A continuación se ven las imágenes de los resultados del registro, tanto en nube de puntos, de la primera y segunda parte del Camino Inca, (Figura 4.1 (a) y (c) respectivamente) y en modelo malla aplicando la reconstrucción de Poisson [30] del mismo software MeshLab (Figura 4.1 (b) y (d)). Ambos resultados provienen del modo de adquisición 2 de muestras con el Kinect.

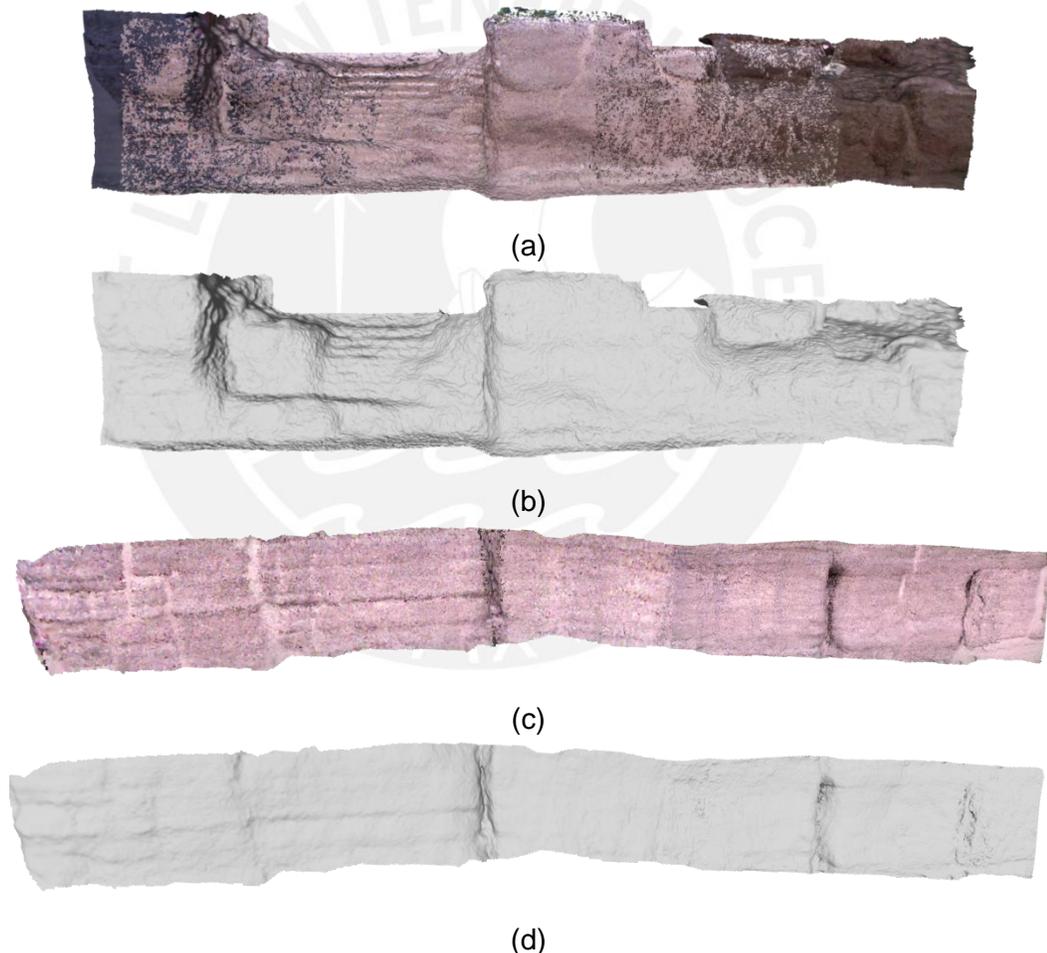


Figura 4.1 Nubes de puntos registradas: En (a) las nubes de puntos con color. Los colores varían debido a la luz solar en el momento de adquisición. En (b) las nubes registradas ya reconstruidas mediante el método de Poisson [30].

Ciertas zonas en la Figura 4.1 (a) tienen un color más claro o más oscuro. Esto se debe a la luz natural presente al tomar las muestras, a mayor luz las imágenes de color serán más claras y viceversa. Por otro lado, vemos que al reconstruir el modelo muchas de las características como las formas de las piedras se conservan, evidenciando así un buen registro. También se evidencia poco los cambios suaves, lo cual se debe a que son de un orden menor al error de precisión del Kinect.

## 4.2 Comparación con otras técnicas de reconstrucción

### 4.2.1 Fotogrametría

Para tener un análisis cuantitativo se decidió comparar la reconstrucción del Kinect con una basada en uno de los métodos más usados en reconstrucción de estructuras arqueológicas, fotogrametría [7]. La cámara usada fue una SONY NEX-7 de 24.3 MP con lente de 16 milímetros, prestada por el Grupo de Sistemas de Aeronaves no Tripuladas de la PUCP.

Se tomaron 30 fotos de las dos zonas analizadas por el Kinect del Camino Inca. Al momento de tomar las fotos hubo un desplazamiento horizontal de aproximadamente medio metro, sumando un tiempo total de adquisición de 5 minutos. Las muestras se introdujeron al software Agisoft [31] para llevar a cabo la reconstrucción tridimensional del área de trabajo. Como se mencionó en el inciso 1.2.4 del primer capítulo, el modelo resultante de la reconstrucción por fotogrametría necesita ser escalado con medidas reales dado que la nube de puntos resultante (ver Figura 4.2 (a) y (b) para el primer caso) tendrá valores adimensionales. Se determinó la escala encontrando la relación entre las medidas de ciertas porciones de la escena versus las dimensiones reales medidas con un flexómetro.

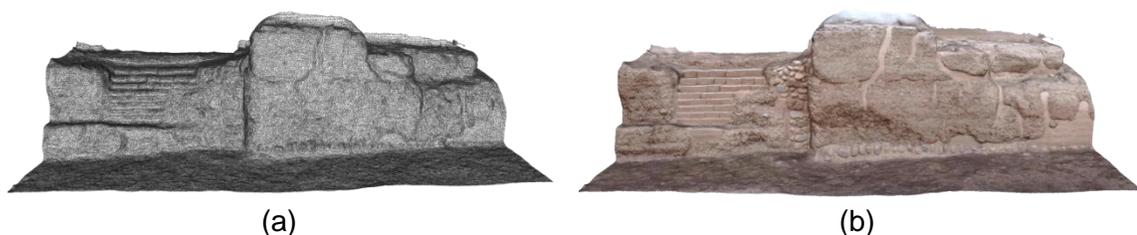


Figura 4.2 En (a) nube de puntos sin color procedente de fotogrametría del primer caso del Camino Inca y en (b) la reconstrucción del mismo luego de haber aplicado Poisson [30].

Una vez que se tienen ambos modelos (el obtenido por fotogrametría y el obtenido por el Kinect con el software propuesto), se procede a alinearlos manualmente en el software MeshLab con el objetivo de compararlos. Como se mencionó en el inciso 3.6 del Capítulo 3 se adquirieron las muestras de dos modos para el primer y segundo caso del Camino Inca: modo 1, automático (Figura 4.3 (a)) y modo 2, manual (Figura 4.3 (b) y (c)). Podemos ver que el registro con muestras obtenidas con el modo 2 cubre mayor área debido a que se controla el tiempo en el que se adquieren los datos, permitiendo que haya un mayor desplazamiento entre muestras de la persona sosteniendo el Kinect.

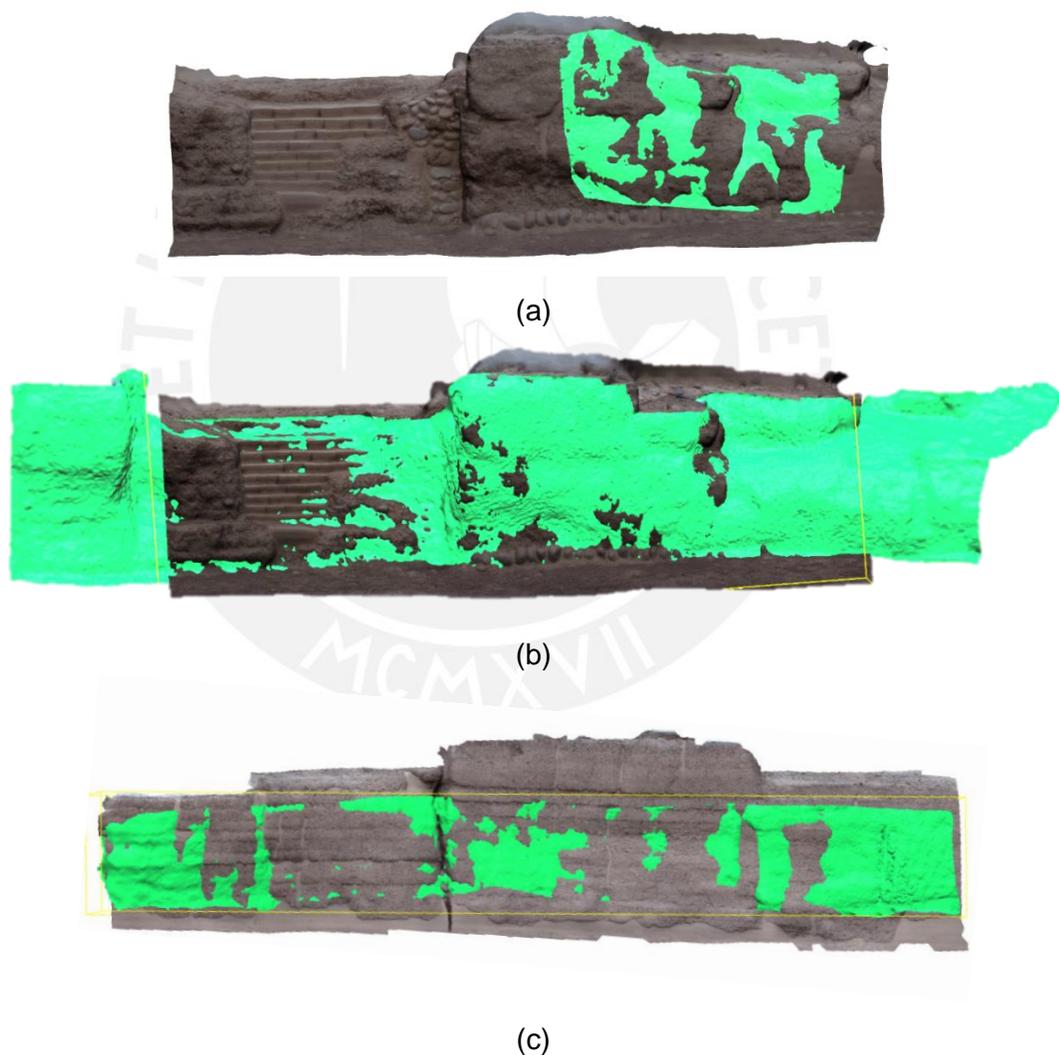


Figura 4.3 Alineaciones de reconstrucciones tridimensionales de fotogrametría a color y Kinect en verde. El primer caso del Camino Inca en (a), adquisición modo 1 del Kinect y (b) con modo 2. En (c) es el segundo caso del Camino Inca con adquisiciones modo 2.

#### 4.2.2 Escáner de triangulación

Debido a que el *NextEngine Laser Scanner HD* es un escáner de triangulación, solo se adquirieron muestras de un pequeño segmento del primer caso del Camino Inca. Se tuvieron que realizar 20 lecturas de datos con el escáner, a una distancia de 40 centímetros, según lo recomendado en [11], de la estructura, por lo cual fue necesaria una cantidad elevada de muestras, elevando así el tiempo total de adquisición a dos horas. Este tiempo de adquisición es alto debido a que los tiempos que toma situar el escáner y adquirir la información con el software propietario son muy extensos.

En la Figura 4.4 (a) se muestra el escáner laser obteniendo los datos y en (b) la alineación de las reconstrucciones tridimensionales entre los modelos del escáner laser y el Kinect, cuyas muestras provienen usando el modo de adquisición 1. Se eligió compararlas con el modelo obtenido por el escáner de triangulación debido a que se tiene una representación más fiel de los detalles dada la alta densidad de puntos, consecuente de tener nubes no tan distanciadas entre sí. Al igual que en el inciso anterior, ambos modelos se alinearon en MeshLab para su posterior comparación cuantitativa.



Figura 4.4 En (a) adquisición de datos del primer caso del Camino Inca y en (b) la alineación de reconstrucción del *NextEngine Laser Scanner HD* y Kinect. El modelo a color es resultado del escáner y el blanco del Kinect.

### 4.2.3 Análisis de resultados

Finalmente, se procede a realizar la diferencia euclidiana de los modelos en el software CloudCompare en donde vemos los resultados en la Figura 4.5, 4.6 y 4.7 La primera columna y la imagen superior de las figuras representa la nube de puntos de Kinect coloreada, donde cada color representa la distancia del punto con respecto al modelo de referencia. Por otro lado, la segunda columna e imagen inferior muestra el histograma de los resultados de dicha sustracción con un ajuste Gaussiano para poder estimar la media y desviación estándar de los errores.

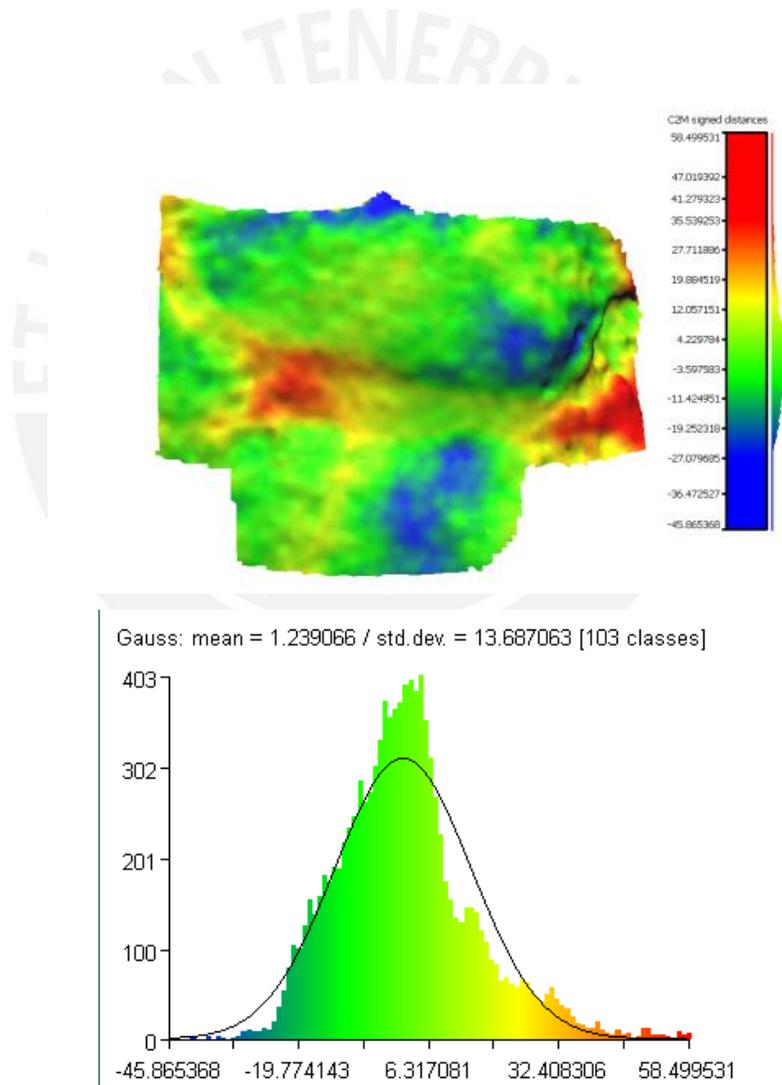


Figura 4.5 Sustracción de modelos tridimensionales e histograma con ajuste Gaussiano para los modelos del *NextEngine Laser Scanner HD* y el Kinect.

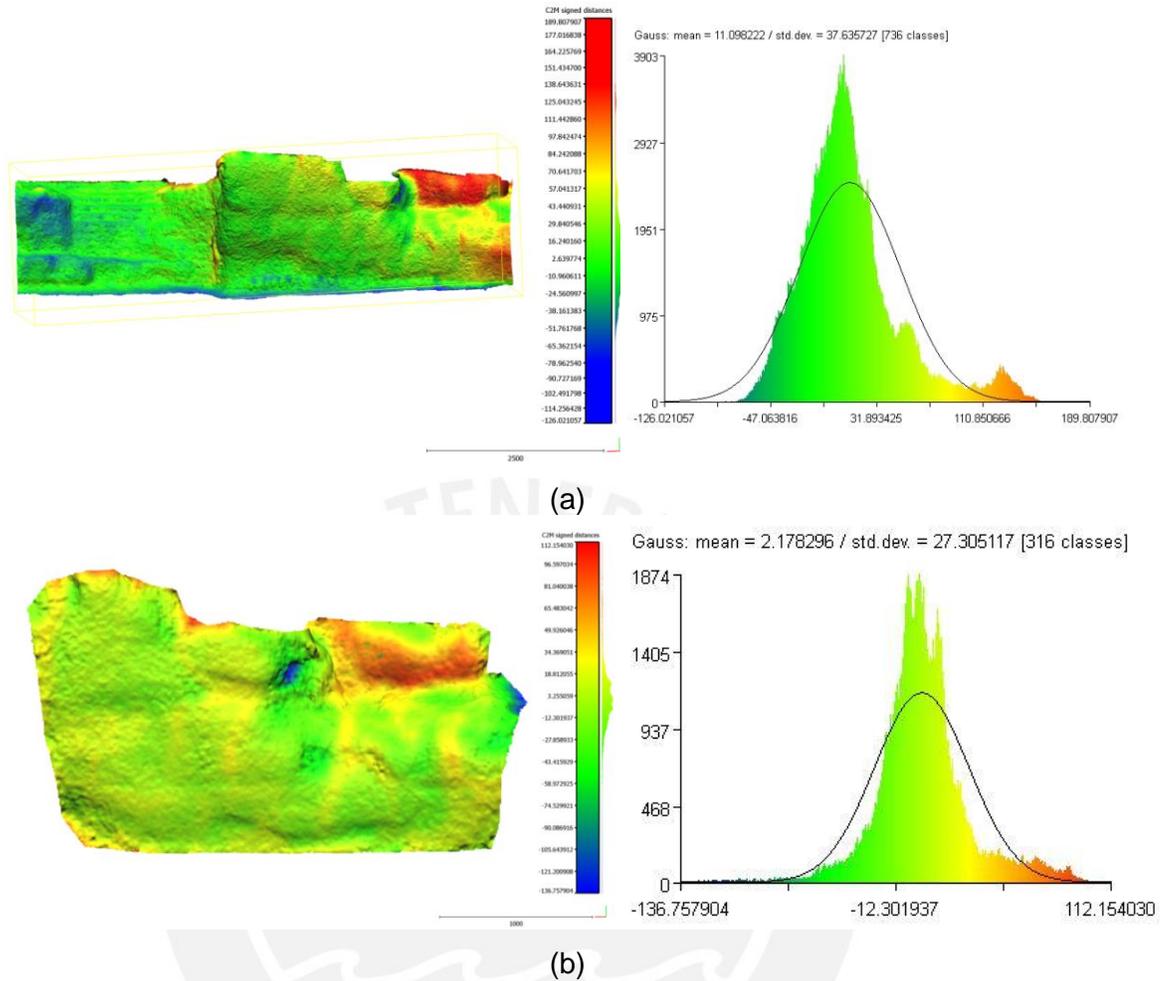


Figura 4.6 Sustracción de modelos tridimensionales e histograma con ajuste Gaussiano para los modelos de Fotogrametría y Kinect. Primer caso del Camino Inca con muestras de adquisición modo 2 del Kinect en (a) y modo 1 en (b).

Al observar los resultados resumidos en la Tabla 3, podemos ver que la exactitud de la reconstrucción del Kinect con la del escáner láser de triangulación es más baja que con la de fotogrametría. Tomando en cuenta los parámetros de la Tabla 2, el error del *NextEngine Laser Scanner HD* es de tan solo 0.127 milímetros, de manera que la reconstrucción basada en tecnología Kinect obtiene valores muy cercanos a la realidad, con 1.49 centímetros de error. No obstante, este escáner laser no se emplea en estructuras arqueológicas debido a su elevado tiempo de adquisición y poca portabilidad. Es por esto que al hacer la comparación con la técnica más usada, fotogrametría, notamos una diferencia clara entre las reconstrucciones basadas en adquisiciones de datos mediante el modo 1 y modo 2.

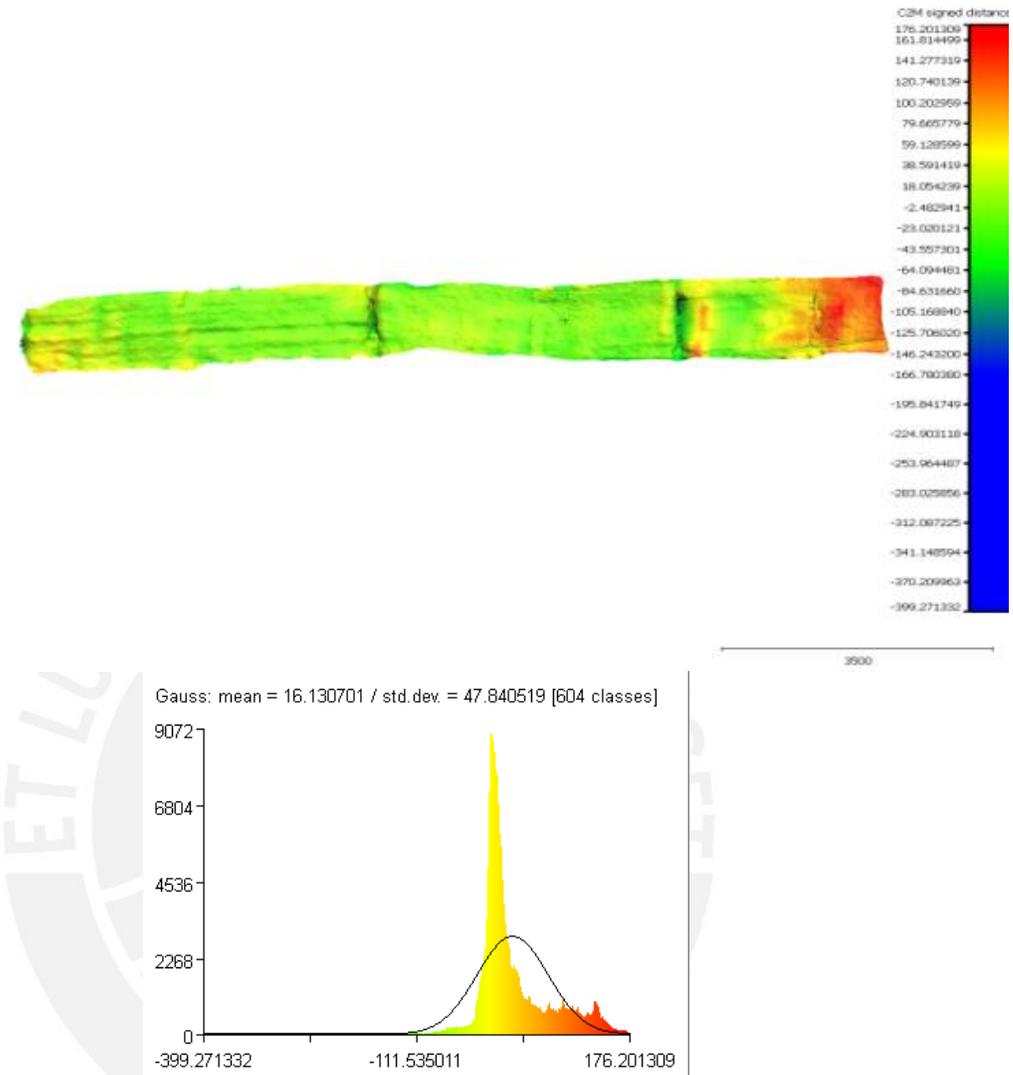


Figura 4.7 Sustracción de modelos tridimensionales e histograma con ajuste Gaussiano para los modelos de Fotogrametría y Kinect. Segundo caso del Camino Inca con muestras de adquisición modo 2 del Kinect.

Tabla 3 Resumen de los valores de error obtenidos luego del ajuste Gaussiano al histograma de la sustracción de modelos.

	Comparación con fotogrametría de Agisoft			Comparación con NextEngine Laser Scanner HD
<b>Error (cm)</b>	4.87*	2.95**	6.39*	1.49

Nota: Los valores (\*) provienen de reconstrucciones con muestras manuales mientras que los (\*\*) con muestras automáticas.

Tabla 4 Comparación de precio, número de muestras y tiempo de adquisición entre las técnicas implementadas.

	Kinect for XBOX	Fotogrametría (SONY NEX-7)	NextEngine Laser Scanner HD
<b>Precio (USD)</b>	108 [13]	1,099.99 [32]	2,995
<b>Cantidad de muestras</b>	200	30	20
<b>Tiempo de adquisición (min)</b>	3	5	120
<b>Software Necesario (USD)</b>	2,000 (Programa desarrollado)	3,499 (Agisoft PhotoScan Professional Edition)	Incluido en el precio del hardware.

Cuando se obtienen muestras de manera automática (modo 1), la separación entre nubes de puntos es mínima y habrá más densidad de puntos en la zona en común con la cual se pueden representar más detalles de la escena. Cabe resaltar que dicha densidad de puntos que se repiten puede aprovecharse para mejorar la calidad de la reconstrucción, se recomienda realizar las investigaciones necesarias en este aspecto. Promediando los tres valores obtenidos de error, se tiene un error de 4.74 centímetros aproximadamente, el cual se encuentra dentro de los límites señalados en la Tabla 2 y [23] como parámetros recomendados para trabajos en arqueología. A continuación se presenta una tabla en donde se comparan las tres técnicas en cuanto a precio, número de muestras y tiempo de procesamiento.

Analizando la Tabla 4, vemos que el Kinect es una alternativa de menor costo que no requiere de mucho tiempo para adquirir la data con un error de 4.74 centímetros. Este error puede parecer significativo pero hay que tomar en cuenta las dimensiones del objeto de estudio, específicamente de la estructura arqueológica, la cual tiene dimensiones en el orden de metros, de manera que un error de esta magnitud no es un problema significativo.

Para un país como el Perú, poseedor de un gran bagaje cultural, un dispositivo como el Kinect satisface las necesidades fundamentales de precio, portabilidad y precisión para poder digitalizar y preservar material arqueológico.

## Conclusiones

1. Se logró caracterizar el Kinect for XBOX y determinar un error de precisión de 2.2 milímetros y de 6.3 milímetros de exactitud en profundidad, en un rango de 0.85 a 1.35 metros con una intensidad de luz entre 60 lux y 100 lux.
2. Se logró diseñar e implementar un algoritmo basado en registro de nubes de puntos capaz de alinear las muestras adquiridas y obtener un modelo geométrico con color para luego ser reconstruido mediante la triangulación de Delaunay y reconstrucción de Poisson.
3. Se logró comparar la reconstrucción tridimensional del muro arqueológico con la técnica de fotogrametría y un escáner láser de triangulación obteniendo un error de 4.74 y 1.49 centímetros. Dada las dimensiones del objeto de estudio, el error no presenta ningún inconveniente y está dentro los parámetros recomendados para estudios en arqueología [23] citados en la Tabla 2.
4. La presente investigación brinda información sobre el uso del Kinect for XBOX como escáner tridimensional y el uso de sus muestras para la reconstrucción de material arqueológico. Otras técnicas como fotogrametría obtienen mejor error de exactitud con un tiempo de adquisición de muestras de casi igual duración (5 minutos versus 3 minutos) pero requiere de equipos muy costosos y el error de precisión es mucho más elevado. Los escáneres de triangulación como el *NextEngine Laser Scanner HD* obtiene mejores errores de precisión y de exactitud pero su poca portabilidad, precio, área y tiempo de adquisición lo descarta como instrumento para este tipo de aplicaciones en estructuras.

## Recomendaciones

1. En el momento de la adquisición de datos se presentaron inconvenientes debido a la aparición de agujeros en las imágenes de rango. Esto se debe a que el Kinect es un dispositivo para uso en interiores y es indispensable realizar muestras en un día muy nublado, pero con suficiente luz para que las imágenes de color no salgan oscuras.
2. En el momento de la adquisición de datos, al desplazar el Kinect horizontalmente mientras se capturan muestras, se deben evitar movimientos bruscos de manera vertical pues estos dificultan considerablemente la alineación.
3. Se recomienda profundizar en los algoritmos de detección de puntos característicos y la relación de los mismos entre muestras para agilizar el tiempo de procesamiento y obtener aún mejores alineaciones iniciales.
4. Debido a las limitaciones de hardware de la computadora empleada no se pudo trabajar con una mayor cantidad de muestras ni incrementar el número de iteraciones con el algoritmo de ICP. Se debe realizar esta etapa de procesamiento en una computadora de mejores características.
5. Se recomienda realizar nuevas investigaciones conforme a mejorar la resolución de la reconstrucción basada en el registro de las nubes de puntos dado que al final se tienen millones de puntos con información relevante y redundante que puede ser trabajada para mejorar la calidad de la reconstrucción. Una investigación tentativa puede ser la aplicación de super-resolución.

## Bibliografía

- [1] Böhler, W., & Marbs, A. (2002, September). 3D scanning instruments. En *Proceedings of the CIPA WG 6 International Workshop on Scanning for Cultural Heritage Recording, Ziti, Thessaloniki* (pp. 9-18).
- [2] Guidi, G., Russo, M., & Angheluddu, D. (2014). 3D survey and virtual reconstruction of archeological sites. *Digital Applications in Archaeology and Cultural Heritage*, 1(2), 55-69.
- [3] Zvietcovich, F., Castaneda, B., Perucchio, R., Morales, R., & Uceda, S. (2013). A Methodology for Updating 3D Solid Models of Complex Monumental Structures Based on Local Point-based Meshes. In G. Guidi, R. Scopigno, & H. S. Yang (Eds.), *Digital Heritage International Congress* (pp. 7-14). The Eurographics Association.
- [4] Arbace, L., Sonnino, E., Callieri, M., Dellepiane, M., Fabbri, M., Iaccarino Idelson, A., & Scopigno, R. (2013). Innovative uses of 3D digital technologies to assist the restoration of a fragmented terracotta statue. *Journal of Cultural Heritage*, 14(4), 332-345
- [5] Parker, M., Daniel, H. C., Echtler, F., & Burrus, N. (2012). *Hacking the Kinect*. New York, USA: Apress
- [6] Li, G. K., Gao, F., & Wang, Z. G. (2011, August). A photogrammetry-based system for 3D surface reconstruction of prosthetics and orthotics. En *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE* (pp. 8459-8462). IEEE.
- [7] Dellepiane, M., Dell'Unto, N., Callieri, M., Lindgren, S., & Scopigno, R. (2013). Archeological excavation monitoring using dense stereo matching techniques. *Journal of Cultural Heritage*, 14(3), 201-210.

- [8] Rojas Gómez, R. A. (2011). Diseño de un sistema de registro de imágenes orientado a la agricultura de precisión. Tesis de Pregrado. Pontificia Universidad Católica del Perú: Perú.
- [9] Arun, K. S., Huang, T. S., & Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5), 698-700.
- [10] Delponte, E., Isgrò, F., Odone, F., & Verri, A. (2006). SVD-matching using SIFT features. *Graphical models*, 68(5), 415-431.
- [11] Zvietcovich Zegarra, J. F. (2014). Estimación del volumen de lesiones producidas por Leishmaniasis cutánea utilizando un escáner láser de triangulación 3D. Tesis de Pregrado. Pontificia Universidad Católica del Perú: Perú.
- [12] Zvietcovich, J. F.; Castañeda, B, Castillo, L. J. y Saldaña, J. M. (2013). A 3D assessment tool for precise recording of ceramic fragments using image processing and computational geometry tools. En *Computer Applications and Quantitative Methods in Archaeology Annual Meeting 2013* (pp. 1-8). Computer Applications and Quantitative Methods in Archaeology.
- [13] SAGA Falabella. "Tienda Online". Consulta 19 de Junio del 2014. <<http://www.falabella.com.pe/falabella-pe/product/13486355/Sensor-de-Movimiento-Kinect-?passedNavAction=>>>
- [14] Kean, S., Hall, J., & Perry, P. (2011). *Meet the Kinect: An Introduction to Programming Natural User Interfaces*. Apress.
- [15] Newegg™. "Online Store". Consulta 19 de Junio del 2014. <<http://www.newegg.com/Product/Product.aspx?Item=N82E16826785030>>
- [16] Minds, B. T. (2012). Arduino and Kinect Projects.
- [17] Park, C. S., Kim, S. W., Kim, D., & Oh, S. R. (2011, November). Comparison of plane extraction performance using laser scanner and Kinect. En *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*(pp. 153-155). IEEE.

- [18] Smisek, J., Jancosek, M., & Pajdla, T. (2013). 3D with Kinect. In *Consumer Depth Cameras for Computer Vision* (pp. 3-25). Springer London.
- [19] MathWorks, 2011. Matlab [software]. Natick: MathWorks. Recuperado de: <http://www.mathworks.com/products/matlab/>
- [20] Visual Computing Lab – ISTI – CNR, 2012. Meshlab [computer software]. Pisa: Visual Computing Lab. Retrieved from: <http://meshlab.sourceforge.net/>
- [21] CloudCompare (version 2.5.3) [GPL software]. EDF R&D, Telecom ParisTech (2012). Recuperado de: <http://www.cloudcompare.org/>.
- [22] Wan, Y., Wang, J., Hu, J., Song, T., Bai, Y., & Ji, Z. (2012, September). A Study in 3D-Reconstruction Using Kinect Sensor. En *Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on* (pp. 1-7). IEEE.
- [23] Koutsoudis, A., Vidmar, B., Ioannakis, G., Arnaoutoglou, F., Pavlidis, G., & Chamzas, C. (2014). Multi-image 3D reconstruction data evaluation. *Journal of Cultural Heritage*, 15(1), 73-79.
- [24] Microsoft Kinect for Windows Support from Image Acquisition Toolbox™. MathWorks. Retrieved from: <http://www.mathworks.com/hardware-support/kinect-windows.html>
- [25] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). IEEE.
- [26] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3), 346-359.
- [27] Juan, L., & Gwon, O. (2009). A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4), 143-152.
- [28] Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2), 119-152.

[29] Delaunay, B. (1934). Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800), 1-2.

[30] Kazhdan, M., Bolitho, M., & Hoppe, H. (2006, June). Poisson surface reconstruction. En *Proceedings of the fourth Eurographics symposium on Geometry processing*.

[31] Agisoft PhotoScan [GPL software]. Agisoft LLC (2006). Retrieved from: <http://www.agisoft.ru/>

[32] © Sony Electronics Inc. "Sony Online Store". Consulta 19 de Julio 2014. <[http://store.sony.com/alpha-nex-7-with-18-55mm-lens-zid27-NEX7K/B/cat-27-catid-Alpha-NEX-7-and-NEX-6?\\_t=pfm%3Dcategory](http://store.sony.com/alpha-nex-7-with-18-55mm-lens-zid27-NEX7K/B/cat-27-catid-Alpha-NEX-7-and-NEX-6?_t=pfm%3Dcategory)>

