

**RECONSTRUCCION 3D DE UN ENTORNO A PARTIR DE
MAPAS DE PROFUNDIDAD**

Oscar Ignacio Morales Correa

Código: 1800915

Edison Jesus Rosero Quintero

Código: 1800950

**UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
INGENIERÍA MECATRONICA
ROBOTICA TERRESTRE
BOGOTÁ, D.C.
2011**

**RECONSTRUCCION 3D DE UN ENTORNO A PARTIR DE
MAPAS DE PROFUNDIDAD**

Oscar Ignacio Morales Correa

Edison Jesus Rosero Quintero

**Opción de Grado para optar al título de
Ingeniero En Mecatronica**

Director

**Nelson Fernando Velasco Toledo
INGENIERO EN MECATRONICA**

**UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
INGENIERÍA MECATRONICA
ROBOTICA TERRESTRE
BOGOTÁ, D.C.
2011**

NOTA DE ACEPTACION

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Bogotá D. C. 13 de Septiembre de 2011

– DEDICATORIA –

"En este mundo no se logra nada útil ni grande sin esfuerzo ni sacrificio. "
- Adolfo Kolping

Este trabajo lo dedico a toda mi familia y en especial a mis padres Jesús Rosero Díaz y Amilbia Quintero Montoya, quienes fueron mi principal motor, que me brindaron el apoyo moral y material para sacar adelante el buen desarrollo y culminación de mi carrera y sacar adelante la realización del proyecto de opción de grado, al igual agradezco inmensamente a mis hermanos Johana Milena Rosero Quintero y Julián Rosero Quintero por ser mis modelos a seguir como personas y profesionales y que también me apoyaron de alguna manera durante esta etapa de mi vida.

Edison Jesús Rosero Quintero

– DEDICATORIA –

"Puede ser que alguna vez pierda con un equipo de GANADORES pero NUNCA va a ganar con un equipo de perdedores" - John Maxwell

Gracias a mis padres Miguel Ignacio Morales y Doralba Correa por el apoyo incondicional que me brindaron durante el trascurso de mi carrera, económica como afectiva, a mis hermanos y amigos que me apoyaron para sacar adelante la realización de este proyecto, y a Ana María Amado quien me acompañó durante toda mi carrera, la cual fue mi principal pilar para lograr uno de los objetivos propuestos en mi vida y espero el de muchos más.

Oscar Ignacio Morales Correa

– AGRADECIMIENTOS –

Los autores expresan sus agradecimientos a:

Nelson Fernando Velasco Toledo, Ingeniero en Mecatrónica, Docente de la Universidad Militar Nueva Granada, Director de la Opción de grado, por su colaboración, seguimiento y asesoría durante el desarrollo del proyecto.

Y a las demás personas que hicieron posible llevar a buen término este proyecto y participaron de alguna u otra forma durante el desarrollo del mismo.

RESUMEN

Este trabajo presenta el desarrollo de un sistema de reconstrucción de ambientes 3D, a partir de la captura consecutiva de imágenes. El sistema se compone de dos módulos principales. El primero realiza la adquisición de datos de las distancias hacia cada uno de los objetos que compongan al ambiente, generando un mapa o nube de puntos, que son procesadas en un ambiente virtual; el usuario puede manipular los datos de la nube de puntos para lograr hacer cambio de texturas, colores y formatos de visualización. El segundo módulo tiene como objetivo realizar un enlace de comunicación variable mediante dos tipos de acceso a los datos como lo es por medio de una red de área local (LAN) o por medio de internet (WAN), entre el host y el cliente, con el fin de trasportar datos o simplemente ver la captura de la información o la reconstrucción completa del ambiente de forma remota.

Todo el sistema de reconstrucción de ambientes 3D y el enlace de comunicación, van integrados sobre una plataforma móvil terrestre como modo de aplicación a la robótica.

Índice general

RESUMEN	10
1. INTRODUCCION	11
1.1. PLANTEAMIENTO DEL PROBLEMA	11
1.2. OBJETIVOS	12
1.2.1. Objetivo General	12
1.2.2. Objetivos Especificos	12
1.3. ESTADO DEL ARTE	12
2. CONSTRUCCIÓN DIGITAL DE OBJETOS	16
2.1. MODELOS PROPUESTOS DE RECONSTRUCCIÓN	16
2.1.1. Nube de Puntos 3D	19
2.1.2. Extracción de Mapas de Profundidad	22
2.1.3. Visión Esteroscópica	24
2.2. KINECT XBOX	28
3. IMPLEMENTACIÓN	32
3.1. Dispositivo Mecánico	33
3.2. Conectividad del Sensor de profundidad (Kinect)	37
3.2.1. Documentación de los formatos de salida de la información entregada por el dispositivo	48
3.2.2. Rendimiento	49
3.3. Generación de la nube de puntos y extracción del mapa de pro- fundidad	49
3.4. Interfaz usuario	53
4. ACCESO AL SISTEMA	56
4.1. Acceso por medio de red de área local (LAN)	56
4.2. Acceso por medio de Internet (WAN)	58
5. PRUEBAS, RESULTADOS	60
6. CONCLUSIONES Y LINEAS FUTURAS	67

Índice de figuras

1.1.	110 FirstLook	13
1.2.	Pelican Quadrotor UAV	13
1.3.	ATE Group	14
1.4.	Digitalizadores Handyscan 3D	14
2.1.	Metodos para generar mapas de profundidad	19
2.2.	Automatización de las fases del procesamiento de un escaneado láser terrestre (adoptado de A.Gruen)	20
2.3.	Imagen de una nube de puntos sin organizar	21
2.4.	Mapa de profundidades	22
2.5.	Resultado de point splatting (izquierda), acercamiento al mismo (derecha)	22
2.6.	Parámetros de imágenes estéreo	25
2.7.	Ángulo óptico. Imagen sacada de la pagina Web de I-art coopora- tion. http://www.i-art.com	26
2.8.	Ángulo visual. Imagen sacada de la pagina Web de I-art coopora- tion. http://www.i-art.com	26
2.9.	Paralelaje	27
2.10.	Escaner 3D de triangulación	28
2.11.	Como detecta objetos	28
2.12.	Funcionalidades del Kinect	29
2.13.	Rango de visión horizontal	30
2.14.	Rangos de visión vertical	30
2.15.	Hardware del Kinect	31
3.1.	Diagrama de bloques del sistema actual	32
3.2.	Plataforma móvil terrestre y Radio control Futaba	34
3.3.	Elementos del soporte	34
3.4.	Estructura completa mas Kinect	35
3.5.	Integración del dispositivo	35
3.6.	Disposición del sistema	36
3.7.	Compartimiento ROUTER	36
3.8.	Ensamble final.	37
3.9.	Libreria OpenNi	38

3.10. http://www.openni.org/downloadfiles/openni-binaries/20-latest-unstable	39
3.11. https://github.com/avin2/SensorKinect	40
3.12. http://www.openni.org/downloadfiles/openni-compliant-middleware-binaries/33-latest-unstable	40
3.13. avin2-SensorKinect-0124bd2	41
3.14. OpenNI-Bin-Win32-v1.0.0.25.exe	41
3.15. SensorKinect-Win32-5.0.0.exe	41
3.16. NITE-Bin-Win32-v1.3.0.18.exe	42
3.17. 0KOIk2JeIBYClPWnMoRKn5cdY4=	42
3.18. Asistente para hardware encontrado	43
3.19. Instalar automáticamente el software (recomendado)	43
3.20. Instalación correcta.	44
3.21. Error de configuración	44
3.22. Ejemplo del OpenNi	45
3.23. Ejemplo NITE	46
3.24. Visualización inicial con infrarojo	46
3.25. Control del reloj (tracking)	47
3.26. Tracking teclado y mouse	47
3.27. Escena tridimensional	48
3.28. Característica general del mapa RGB-D.	49
3.30. RGB de imagen	50
3.29. Píxel	50
3.31. Profundidad de la imagen	51
3.32. Triangulación por Delaunay	53
3.33. Interfaz MeshLab	54
3.34. Texturas diferentes.	54
3.35. Control de la interfaz.	54
4.1. Red de área local (LAN)	57
4.2. Internet	59
5.1. Montaje Final (Vista ISO)	60
5.2. Montaje final (Vista Frontal y Trasera)	61
5.3. RGB	62
5.4. Mapa de profundidades	62
5.5. Máximos y mínimos de cobertura de la imagen	63
5.6. Mapa reconstruido	64
5.7. Ambiente reconstruido (Pasillo-Laboratorio)	65
5.8. Reconstrucción del pasillo	66
5.9. Medición virtual del ambiente reconstruido	66

Capítulo 1

INTRODUCCION

Hoy en día el ser humano necesita de herramientas que le ayuden a realizar actividades complementarias a su trabajo, ya sea por sus diversos niveles de complejidad o por algún grado de peligrosidad, que impidan o dificulten la labor de las personas; estas herramientas deben ser un complemento a las destrezas humanas, mejorando así la fuerza, visión, velocidad, precisión y exactitud. Una de las herramientas más utilizadas y desarrolladas tecnológicamente hoy en día son los robots móviles, que facilitan la posibilidad de desplazarse desde un inicio establecido, hasta una meta propuesta para algún propósito.

Este trabajo muestra el desarrollo de un sistema de visión de máquina y una red comunicación capaz de recolectar información de las distancias entre objetos que componen una ambiente específico con características definidas; esta recolección de información se hizo con el fin de realizar una reconstrucción 3D del entorno antes mencionado. Este dispositivo se instaló sobre una plataforma móvil manejada a control remoto, la cual navega por el entorno permitiendo que el dispositivo realice la toma de datos para su posterior procesamiento, y que el usuario final pueda tener un acceso remoto a los datos para navegar por el entorno reconstruido en 3D y poder realizar una exploración supervisada del desplazamiento de la plataforma.

1.1. PLANTEAMIENTO DEL PROBLEMA

La visión de máquina es una herramienta muy importante e indispensable para la navegación, reconocimiento y reconstrucción de ambientes en los cuales el ser humano no puede acceder por diferentes factores como el tamaño que puede llegar a ser reducido, o donde este en riesgo la vida; por tal motivo se le asignan tareas a herramientas como lo son las plataformas móviles.

No todas las plataformas móviles poseen visión de máquina, esto hace que sus destrezas en el desplazamiento sean menores, y que las tareas asignadas

sean limitadas en factores como lo es, el rango visual del operador; al igual no se podría hacer reconocimientos ni tareas supervisadas a largas distancias, la plataforma correría el riesgo de colisionar con obstáculos del ambiente o perder movilidad por factores ambientales diversos.

La visión de máquina permite realizar diversas aplicaciones en las cuales se puede analizar, visualizar y recrear entornos que sean de interés por alguna razón, además es posible realizar labores de navegación de forma remota, supervisada y a largas distancias ya que el rango de visión y control no iría estrictamente sujetas a las capacidades físicas del ser humano.

1.2. OBJETIVOS

Los objetivos se componen de uno general y tres específicos, donde cada item nos dara bases para alcanzar nuestras metas.

1.2.1. Objetivo General

- Realizar la reconstrucción 3D de un entorno a partir de mapas de profundidad, desde una plataforma móvil terrestre teledirigida.

1.2.2. Objetivos Especificos

- Integrar sobre una plataforma móvil terrestre un dispositivo que realice la adquisición de los mapas de profundidad del entorno.
- Establecer una arquitectura de red de comunicación offline desde la plataforma móvil terrestre hasta la estación remota.
- Implementar algoritmos de reconstrucción del entorno 3D para la estación remota que permita realizar el procesamiento de imágenes para los mapas de profundidad.

1.3. ESTADO DEL ARTE

En la actualidad se han desarrollado diversos proyectos en los que interviene la visión de maquina directamente en el funcionamiento de los mismos, que no solo facilitan sus destrezas sino que también ayudan a adquirir información adicional para un posterior análisis.

Entre ellos se encuentra la compañía multinacional iRobot, la cual se ha especializado en la construcción de robots terrestres militares, en el caso del 110 FirstLook [1], que realiza la observación autónoma e investigación recurrente de espacios cerrados analizando la morfología de objetos y de humanos.



Figura 1.1: 110 FirstLook

Este dispositivo se asemeja al sistema desarrollado ya que la plataforma de iRobot cuenta con un embebido interno, el cual posee un sistema de comunicacion de alta velocidad, con un rango de maniobrabilidad de 5 a 10km.

Se encontró por otra parte un sistema de tránsito aéreo desarrollado en un aeropuerto el cual viaja largas trayectorias recreando factores inciertos y aleatorios en las pistas de vuelo; este sistema llamado Pelican Quadrotor UAV [1.2], cuenta con un sistema de visión generada por el sensor Kinect implantado en su arquitectura principal.



Figura 1.2: Pelican Quadrotor UAV

Este dispositivo genera mapas de profundidad basandose en el metodo de puntos, o nube, debido a que recrea a partir de estos el ambiente en el cual esta inmerso el Quadrotor; lo cual lo asemeja a un nivel basico al desarrollo propuesto para la reconstruccion 3D del entorno, pero con la limitante de la velocidad de transferencia de la informacion.

ATE Group ha creado unos dispositivos de reconstrucción tridimensional basada en la simulación 3D en el área automotriz, aeroespacial, arquitectura y productos de consumo personal.



Figura 1.3: ATE Group

En el área automotriz se han encargado de la digitalización, diseño y modelado 2D y 3D, para la recopilación y el seguimiento de modificaciones efectuadas a un prototipo; preparación de datos para análisis de fatiga y simulación digital.

Otra área de aplicación de estos dispositivos es en la aeroespacial aplicando la ingeniería inversa de turbinas de gas, compartimientos de motor, cubiertas de mecanismos y cabinas de mando, las cuales no requieren un modelo 3D previo a los diseños, lo cual reduce costos y crea versiones, modificaciones y aplicaciones a escala.

En el área arqueológica se logra la digitalización 3D a color sin contacto de fósiles o artefactos en el sitio, con fines de reconstitución y reconstrucción. Todo lo anterior genera productos de consumo al público, ya sea para actualización de diseño o recopilación de información por medio de los Digitalizadores Handyscan 3D, como lo es el UNISCAN [3] el cual fue desarrollado para aplicaciones básicas, el REVSCAN [4] creado para realizar tareas de ingeniería inversa, el EXASCAN [5] con fines de alta precisión y definición para tareas de inspección, el VIUSCAN [6] para la digitalización de reproducción de colores, el MAXSCAN [7] desarrollado para la reconstrucción de partes grandes y el ERGOSCAN [8] para aplicaciones médicas.



Figura 1.4: Digitalizadores Handyscan 3D

El sistema propuesto en este documento ya comercializado llega a ser un modelo a igualar, primero por la velocidad de transferencia y el procesamiento de la información capturada por el dispositivo, segundo por el grado de detalle con el que el dispositivo reconstruye la información suministrada por el ambiente inferior al milímetro en precisión y exactitud, y por último el post-procesamiento que dicha reconstrucción permite generar.

Capítulo 2

CONSTRUCCIÓN DIGITAL DE OBJETOS

Esta sección reúne el desarrollo de métodos hechos hasta el momento, dando una breve reseña de posibles formas de manipulación de datos para la reconstrucción en el ámbito de programación; y por otro lado se presentan las características del dispositivo de adquisición de imágenes (Kinect) con su función de obtención de datos crudos.

La reconstrucción digital es el proceso de reproducción de objetos reales en un computador, manteniendo características físicas como lo son las dimensiones, el volumen y la forma, de los objetos que componen al ambiente. A nivel nacional e internacional el desarrollo en las técnicas de mallado y reconstrucción se basa en obtener un algoritmo capaz de elaborar una conexión de acuerdo a un conjunto de puntos generando diversos elementos con formas geométricas.

La eficacia de los métodos utilizados delimita la calidad final de la construcción. Si se crea un grupo de puntos cuya representación se generan a partir de datos erróneos o no definidos, existirá una incongruencia final de puntos en el mallado que no cumplan con las condiciones adecuadas que representen la superficie del objeto.

2.1. MODELOS PROPUESTOS DE RECONSTRUCCIÓN

Existen múltiples propuestas en la literatura del proceso de reconstrucción de objetos [9-17], en las cuales se define un conjunto de etapas basadas en técnicas de visión computacional.

El sistema de análisis de escenas estereoscópicas presentado por Koch [17]

utiliza técnicas de correspondencia de imágenes, segmentación de objeto, interpolación y triangulación para obtener el mapa densidad de puntos. Este sistema tiene como entrada secuencia de pares de imágenes estereoscópica y salida modelo de superficie. El sistema se divide en tres módulos: sensor de procesamiento, procesamiento de par de imagen, procesamiento de secuencia basada en modelo.

M. Pollefeys[15], presenta un proceso de reconstrucción, que consiste en etapas bien definidas. La entrada es una secuencia de imágenes y la salida del proceso es un modelo de la superficie. Las etapas son las siguientes: relación de imágenes, estructura y recuperación de movimiento, correspondencia densa y construcción de modelo.

Otra propuesta la entrega Remondino [16]. Describe un sistema de reconstrucción con las siguientes etapas: adquisición y análisis de secuencia de imágenes, calibración y orientación de las imágenes, proceso de matching y la generación de puntos y modelado.

Las fases del sistema de reconstrucción propuesto utilizan métodos conocidos en la literatura para cada una de las fases en el desarrollo del proceso. La calibración es un proceso cuyo objetivo es determinar la matriz de calibración compuesta por los parámetros intrínsecos; para determinar estos valores se utilizó el método de Zhang [18]: “la técnica requiere sólo una cámara para observar un patrón plano mostrando pocas (al menos dos) diferentes orientaciones”. Este método fue seleccionado por su simpleza y sus buenos resultados comparados con otros métodos de calibración de cámara [20].

“Puntos de interés simplemente significa cualquier punto en la imagen para la cual cambia la magnitud de la intensidad; convencionalmente las “esquinas” tales como L-esquinas, T-uniones o Y-uniones satisfacen estas características, también los puntos negros en fondos blancos” [21]. La detección de puntos característicos o de interés es fundamental para el proceso de reconstrucción, debido a que de ello depende fuertemente la cantidad de puntos que determinan el objeto.

El método de Harris [22] basado en intensidades entrega mejores resultados [21], donde los parámetros que varían para la elección son el operador gradiente y la medida de confianza. La implementación de Harris [22] utilizó el operador de Prewitt² y medida de confianza de Noble [33]. La correspondencia de puntos es una de las etapas donde se centra mayor atención de estudio debido a su complejidad y a la importancia en el resultado final de la aplicación. En la literatura se describen dos tipos de correspondencia por correlación [26, 33] y por rasgos [34].

La correspondencia de puntos característicos es realizada por correlación en torno a los puntos en la ventana de búsqueda, con resolución de ambigüedades por máximos en ambas direcciones utilizando implementación de Kovesi [23].

La reconstrucción de puntos permite determinar la profundidad a partir de un par de puntos correspondientes utilizando triangulación [33-36]; la complejidad radica en la información que se posea del sistema referido a los parámetros intrínsecos de cámara (K) y los parámetros extrínsecos (R , T) entre pares de vistas. La factorización [30, 31] es un proceso que permite determinar R y T a partir de la matriz esencial. El método de Tsai [30] permite determinar los parámetros de movimiento entre el par de vistas y minimizar el error cuadrático que optimiza los valores resultantes. Con los parámetros podemos calcular la profundidad relativa utilizando el método de Weng [31].

La formación de la superficie crea triángulos con los puntos formando una malla, utilizando el método de Delaunay [32], al cual se aplica optimización de los triángulos, eliminando los triángulos que no forman parte del objeto. Los puntos reconstruidos de cada par de vistas son unidos mediante puntos comunes entre las vistas sucesivas utilizando cambio de referencia [33]. El objetivo es proponer un sistema de reconstrucción de objeto 3D desde un conjunto de imágenes. A partir de un objeto se captura una secuencia de vistas sucesivas de distintos ángulos de visión, cada una de las cuales representa una parte del objeto en el marco de referencia imagen 2D. El conjunto de vistas se utilizan para generar el correspondiente modelo poliedro que representa al objeto.

Para poder realizar la construcción tridimensional, se necesita asemejar un dispositivo a la vista humana, esto quiere decir que es necesario tener dos dispositivos de adquisición de imágenes para lograr de esta forma obtener profundidades.

Nuestro cerebro procesa las diferencias entre ambas imágenes y las interpreta de forma que percibimos la sensación de profundidad, lejanía o cercanía de los objetos que nos rodean. Este proceso se denomina estereopsis. La distancia interpupilar más habitual es de 65 mm, pero puede variar desde los 45 a los 75 mm.

En la estereopsis intervienen diversos mecanismos. Cuando observamos objetos muy lejanos, los ejes ópticos de nuestros ojos son paralelos. Cuando observamos un objeto cercano, nuestros ojos giran para que los ejes ópticos estén alineados sobre él, es decir, convergen. A su vez se produce la acomodación o enfoque para ver nítidamente el objeto. Este proceso conjunto se llama fusión [37].

En la reconstrucción de ambientes 3D, se manejan varias técnicas para el tratamiento de la información suministrada por el sensor, estas técnicas nos permiten modificar los datos, logrando de esta manera dar formas, texturas y colores a los objetos. Los métodos más relevantes y significativos utilizados en el desarrollo del sistema para este trabajo se mencionan a continuación.



Figura 2.1: Metodos para generar mapas de profundidad

2.1.1.1. Nube de Puntos 3D

La nube de puntos es un grupo de puntos dentro de un sistema de coordenadas tridimensional. Estos puntos se identifican normalmente como coordenadas X, Y, y Z y son representaciones de la superficie externa de algún objeto.

Las nubes de puntos se crean habitualmente con un sistema de escáner láser tridimensional. Este sistema mide de forma autónoma una gran cantidad de puntos en la superficie de un objeto, y genera un archivo de datos con una nube de puntos.

Las nubes de puntos tienen variadas aplicaciones, entre las que se incluyen la obtención de modelos tridimensionales en CAD de piezas fabricadas, la inspección de calidad en metrología, y muchas otras en el ámbito de la visualización, animación y texturización.

Aunque las nubes de puntos se pueden revisar y texturizar directamente [40], normalmente no se utilizan de esta forma en la mayoría de las aplicaciones tridimensionales, ya que se convierten en mallas de polígonos o en mallas triangulares irregulares, modelos NURBS, o modelos de CAD mediante un proceso denominado reconstrucción de superficies.

Hay varias técnicas para convertir una nube de puntos en una superficie tridimensional. Algunos procedimientos como la triangulación Delaunay o las formas alfa construyen una red de triángulos a partir de los vértices de la nube de puntos, mientras que otros convierten la nube de puntos en un volumen de vóxel y reconstruyen la superficie implícita mediante un algoritmo de Marching

cubes. [41]

Procesamiento de nubes de puntos 3D

El procesamiento de una nube de puntos quiere decir la transformación de la nube de puntos cruda registrada en un resultado final. Este resultado final puede acoger varios formatos: nube de puntos limpiada, planos 2D estándar (planos, elevaciones, secciones transversales. . .), modelos 3D totalmente texturizados para animaciones y navegación.

Los creadores de escáneres muestran videos impresionantes de nubes de puntos que se transforman en modelos texturizados en menos de un segundo. Este proceso es mucho más largo en la realidad y la mayoría de las veces manual. En la figura de abajo (Fig. 2.2), se muestra una visión de las diferentes fases del procesamiento del escaneado láser y su grado de automatización.

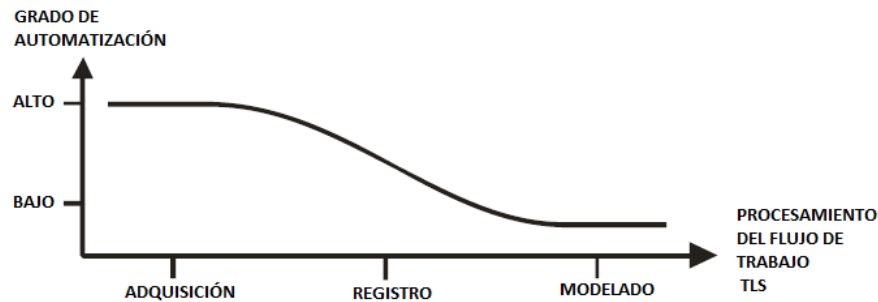


Figura 2.2: Automatización de las fases del procesamiento de un escaneado láser terrestre (adoptado de A.Gruen)

El procesamiento de una nube de puntos 3D, se puede dividir en dos categorías. Los resultados finales, que son aquellas coordenadas en tres dimensiones de la posición de cada elemento, se pueden extraer directamente de la nube de puntos sin más procesamiento, o creando primero un modelo 3D de la superficie a partir de la nube de puntos y extrayendo los resultados de este modelo. Qué método se elige dependerá de los resultados pedidos. Por ejemplo, cuando sólo se piden un número limitado de secciones transversales, es mejor extraerlas directamente de la nube de puntos. Sin embargo, cuando se necesita un mayor número de secciones (más de 50), el segundo método es más eficiente porque hay herramientas automáticas para generar múltiples secciones a partir de un modelo triangulado. Además, el modelo de la superficie añade más valor y comprensión que sólo la nube de puntos registrada.[42]

Representaciones de las nubes de puntos

El resultado de una captura de un escáner es una gran cantidad de puntos en el espacio, cada uno teniendo unas coordenadas X,Y,Z (Fig. 2.3) y, normalmente, un valor de reflectividad. Algunos escáneres proporcionan incluso información del color en forma de valores RGB (rojo, verde y azul).

La nube de puntos se puede representar dibujando todos los puntos en la pantalla, pero esto conlleva una impresión caótica y el usuario puede tener dificultades para reconocer las estructuras en la nube. Cuando cada punto se muestra con su valor de reflectividad o color, el conjunto de toda la estructura se entiende mejor.[42]

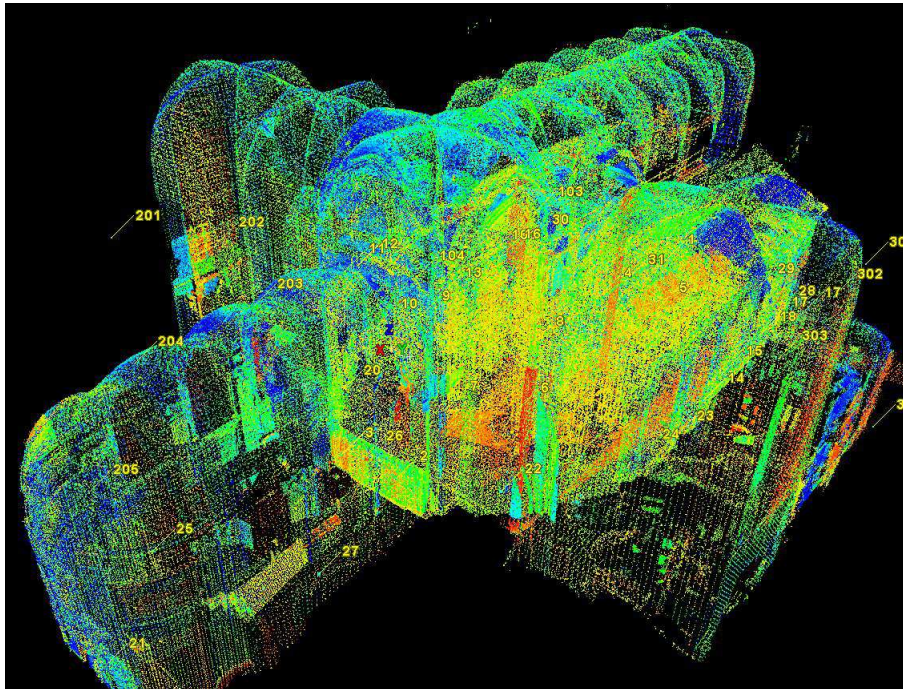


Figura 2.3: Imagen de una nube de puntos sin organizar

Como la mayoría de escáneres toman una escena en filas y columnas, una manera de representar una nube de puntos de una forma muy simple es con un mapa de profundidades. Un mapa de profundidades es una estructura en forma de matriz (2D) en la que cada píxel representa la distancia del punto 3D al escáner en forma de un valor de gris (Fig. 2.4). Debido a que este tipo de representación incorpora información del entorno de cada punto, es de gran utilidad en los algoritmos de procesamiento de la nube de puntos y se conoce como nube de puntos organizada.[42]



Figura 2.4: Mapa de profundidades

Utilizando complejos algoritmos de modelización (triangulación), los puntos vecinos se pueden conectar para formar superficies. Esto proporciona una representación más cercana a la realidad porque las estructuras superficiales o modelos no son transparentes y, por tanto, los puntos que quedan detrás de otros no pueden ser vistos. Calculando la dirección normal de la superficie, se pueden utilizar sombras artificiales para enfatizar detalles de la superficie.

Como la generación de modelos, especialmente a partir de nubes de puntos no organizadas, es compleja y puede tomar un tiempo considerable, ha habido intentos de encontrar alternativas para conseguir más rápidamente una representación aproximada de la nube sólo para visualizarla y analizarla. Como resultado, se lanzó la idea de point splatting, que generan “surfels” (“SURface ELeMents), pequeños elementos de superficie para cada punto de la nube creada a partir de los datos crudos del escáner láser (Fig. 2.5). Cada surfel se representa mediante una pequeña forma primitiva de superficie (círculo, elipse. . .) en 3D que hereda la normal de la superficie que forma con sus vecinos. Ésto resulta en una representación de la superficie muy rápida.[42]

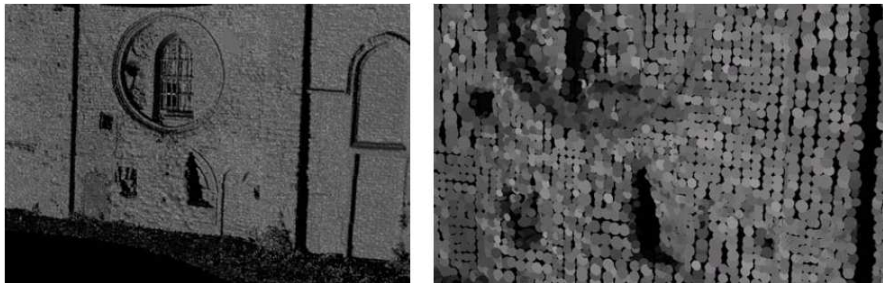


Figura 2.5: Resultado de point splatting (izquierda), acercamiento al mismo (derecha)

2.1.2. Extracción de Mapas de Profundidad

Existen diversas formas de abordar la reconstrucción tridimensional de una escena. En primer lugar habrá que atender al tipo de información disponible

según el sensor empleado. Según su principio de funcionamiento es posible diferenciar dos familias de sensores:

- Aquellos que proporcionan directamente una medida de la distancia a la superficie del objeto. Los dispositivos empleados son elementos activos, compuestos por un emisor y un receptor de un determinado tipo de señal. Su principio de funcionamiento consiste en medir el retardo con que se recibe una señal emitida por el sensor tras reflejarse en los objetos de la escena. A esta familia pertenecen los sensores de rango láser, por ultrasonidos, radar, sonar, etc.
La ventaja definitiva de este tipo de sensores es la obtención directa del mapa de distancias que separa al observador (sensor) de cada punto de la escena. En su contra se pueden mencionar su alto precio, su poca versatilidad y la desigual calidad de los resultados obtenidos dependiendo del sensor empleado y las características de la escena a observar.
- La alternativa a las medidas proporcionadas por los sensores de rango la constituye la determinación indirecta de distancias. Estos métodos parten de la información proporcionada por una o varias imágenes de la escena sujeta a estudio. En este caso el sensor usado suele ser una cámara sensible a una región del espectro electromagnético, habitualmente el rango visible.
En este caso el inconveniente de la obtención indirecta del mapa de profundidades queda paliado por el relativo bajo coste y la versatilidad de los sensores empleados.

Centrándonos en esta segunda posibilidad, determinación indirecta de distancias, el número de imágenes, las características a observar y el resultado obtenido dan lugar a una multiplicidad de técnicas.

Las técnicas basadas en luz estructurada emplean la deformación que sufre un patrón luminoso de características conocidas al proyectarse sobre la escena estudiada. Existen métodos que parten de una sola imagen, aunque normalmente se suelen considerar varias imágenes tomadas con distintos patrones conocidos. Estas técnicas usan como elemento sensor una cámara, por lo tanto la obtención de medidas es indirecta, aunque requieren de un patrón de iluminación conocido.

El siguiente grupo de técnicas son las denominadas "Forma a partir de 'X'" en las que se estudia la aparición de la característica 'X'. Al igual que en el grupo anterior existen técnicas basadas en el estudio de una sola imagen, principalmente en la determinación de Forma a partir de sombreado, aunque como norma general se recurre al estudio de grupos de imágenes sobre los que se mide la variación de una determinada característica. Además del sombreado de las imágenes, otras características objeto de interés de este tipo de técnicas son la aparición de bordes o siluetas dependiendo de la situación relativa cámara-escena.

Por último, las técnicas de visión estereoscópica necesitan al menos dos imágenes. Se basan en el establecimiento de correspondencias entre la situación sobre distintas imágenes de la proyección de un conjunto determinado de puntos de la escena. A partir de estas correspondencias se obtienen por triangulación las medidas observador-escena. El problema en este tipo de métodos es el establecimiento de un correcto emparejamiento de estas proyecciones.

Es en este último grupo de técnicas en el que se ha centrado el trabajo desarrollado por considerarlo el más adecuado al problema concreto a resolver: reconstrucción volumétrica de una escena a partir de las distintas vistas que de ésta se obtienen situando una cámara en distintas posiciones de su entorno.[43]

2.1.3. Visión Esteroscópica

La determinación de distancias mediante emparejamiento de vistas estereoscópicas ha sido una de las áreas de investigación más activas en visión por computador. Se han propuesto multitud de algoritmos que abordan el problema según distintos enfoques y con distintas suposiciones sobre las características de la escena a explorar. El objetivo de esta sección es presentar un panorama de conjunto de las técnicas utilizadas en visión estéreo. Para ello se establecerán una serie de criterios clásicos de clasificación y se describirán someramente algunos de los algoritmos que más aceptación han tenido y en los que se ha basado el trabajo realizado. Revisiones exhaustivas de algoritmos clásicos pueden encontrarse en los artículos de Barnard [44], Dhond [45] y Brown [46].

Una primera clasificación puede realizarse teniendo en cuenta el número de puntos sobre los que se establecen correspondencias. Atendiendo a este criterio, es posible hacer una primera distinción entre métodos basados en identificación de características y métodos basados en emparejamiento de áreas.

Los primeros realizan una extracción de características de alto nivel en cada una de las imágenes analizadas. Posteriormente se intenta establecer un emparejamiento entre los conjuntos de características presentes en cada una de las imágenes. Los segundos, por el contrario, trabajan con toda la imagen, intentando encontrar zonas de máxima similitud según una métrica seleccionada.

La necesidad de eliminar falsos emparejamientos y la búsqueda de algoritmos más robustos y estables ha propiciado el desarrollo de métodos en los que el número de imágenes empleadas es mayor de dos, siendo habitual encontrar trabajos que usan la información proporcionada por tres imágenes. Algunos de ellos usan los datos proporcionados por la tercera imagen simplemente para corroborar los resultados obtenidos con los datos proporcionados por las dos primeras. Otros por el contrario hacen uso de toda la información disponible de manera conjunta.

Un campo de investigación relacionado con la visión estéreo es el de la geometría epipolar, que permite reducir el espacio de búsqueda de dos a una dimensión, y la llamada calibración epipolar o calibración débil. Con esta técnica es posible obtener reconstrucciones de la escena sin la necesidad de conocer explícitamente los parámetros intrínsecos o extrínsecos de cada una de las cámaras utilizadas.

Imagen Estereoscópica

La imagen estereoscópica es llamada de esta forma ya que existe una fusión de la imagen de la cámara derecha con la imagen de la cámara izquierda, cada uno de los elementos que componen la imagen estéreo recibe el nombre de campo.

En la siguiente figura (2.6) se puede observar los parámetros de las imágenes estéreo, en donde (DIO) hace referencia a la distancia intraocular, (d) es la distancia al centro de proyección, (Paralax) que es el paralelaje entre cada imagen formada por cada cámara y depende de los factores anteriores y por ultimo (HIT) que hace referencia a la traslación horizontal de la imagen.

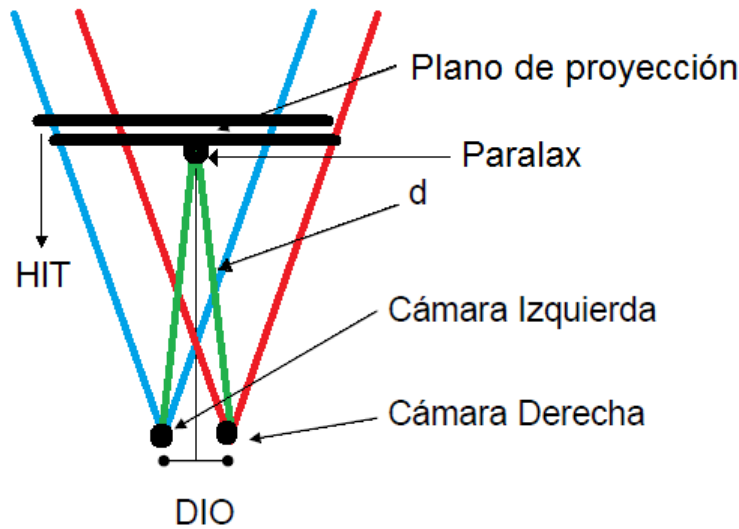


Figura 2.6: Parámetros de imágenes estéreo

Como se ve en la figura (2.7), el objeto 1 está más cerca de las cámaras que el objeto 2, esto debido a que el ángulo θ_1 es mayor que el ángulo θ_2 , a estos ángulos se les conoce como ángulos ópticos. El ángulo óptico está formado por el objeto al que estamos observando y las dos cámaras izquierda y derecha. Gracias a este ángulo, nos damos cuenta que un objeto está más cerca o lejos de nosotros.

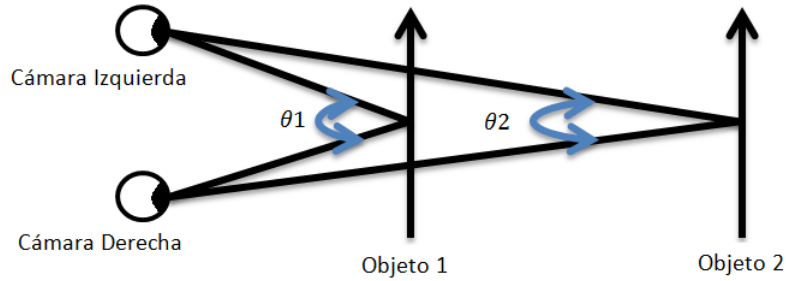


Figura 2.7: Ángulo óptico. Imagen sacada de la pagina Web de I-art cooperation. <http://www.i-art.com>

De la misma forma se conoce en física el concepto de ángulo visual, el cual hace referencia al ángulo que se forma entre la parte superior e inferior del objeto y uno de nuestras cámaras. Con la ayuda de este ángulo se es capaz de distinguir si un objeto es más grande que otro. Tal y como se observa en la figura de abajo (2.8), el ángulo θ_1 es menor que θ_2 , así que podemos concluir fácilmente que el objeto 1 es más pequeño que el objeto 2.

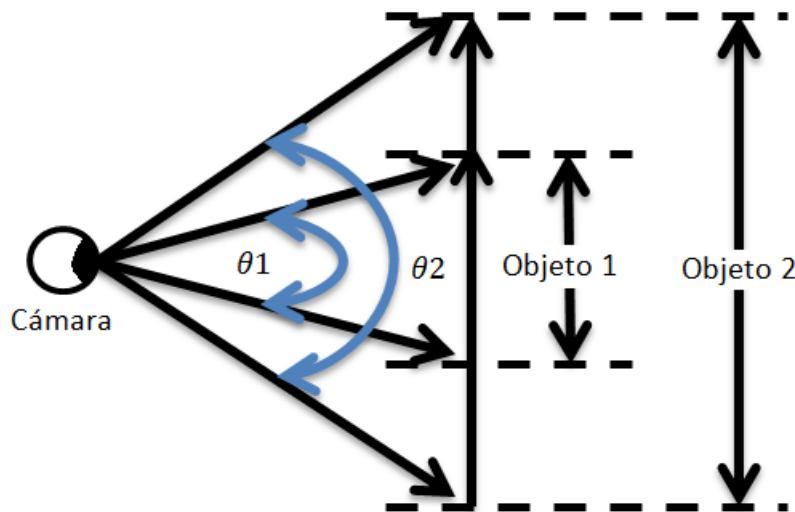


Figura 2.8: Ángulo visual. Imagen sacada de la pagina Web de I-art cooperation. <http://www.i-art.com>

El paralelaje o parallax es la distancia entre las dos proyecciones de un punto en el plano de proyección, esto conlleva que exista una disparidad entre las dos proyecciones del punto en los focos de las cámaras, en la siguiente imagen (Fig. 2.9) se puede ver los tres tipos de paralelaje que dependen de la distancia

intraocular entre las cámaras, hay que tener en cuenta que entre más DIO, habrá mas parallax.

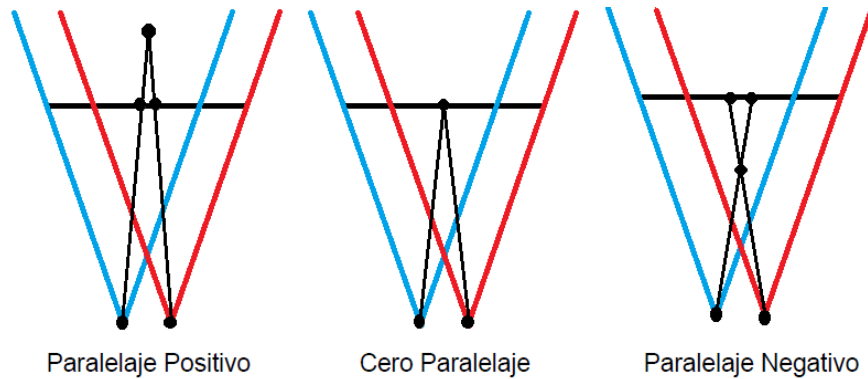


Figura 2.9: Paralelaje

Triangulación

El escaner 3d de triangulación es un escaner activo que usa la luz laser para examinar el objeto. En este caso el brillo del laser en el objeto se examina mediante una camara fotográfica para determinar su posición. Dependiendo de lo lejano esté el punto del objeto en que brilla el laser, incidirá en diversos sitios del campo visual de la cámara.

Esta tecnología se llama de triangulación porque el punto donde brilla el laser, el emisor laser y la cámara forman un triangulo. De este triangulo conocemos el lado que une la cámara con el emisor láser, el ángulo de la esquina del emisor láser también es conocido, y el ángulo de la esquina de la cámara se puede determinar examinando la localización del punto en el campo visual de la cámara. Así con estos tres valores se obtiene la forma y tamaño del triángulo formado y se determina la posición tridimensional de cada punto del objeto. En la mayoría de los casos, en lugar de analizar un solo punto, se analiza un segmento, con lo que se acelera el proceso de captura.

Con respecto a los escaneres de tiempo de vuelo, los escaneres de triangulación son más precisos (del orden de 10 micrometros), pero tienen un campo de acción de unos cuantos metros; mientras que los de tiempo de vuelo pueden operar en radios de acción de hasta kilómetros con precisiones del orden de milímetros.

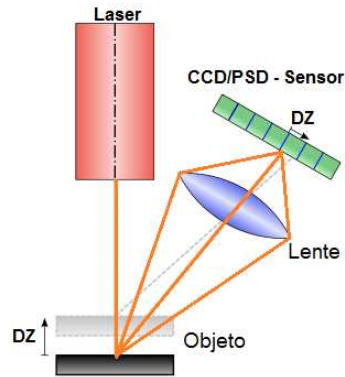


Figura 2.10: Escáner 3D de triangulación

2.2. KINECT XBOX

Existen diversos sensores capaces de medir profundidades mediante sensores, uno de estos es el dispositivo Kinect del XBOX 360 el cual cuenta con una cámara RGB, un sensor de profundidad, un micrófono multi-array y un procesador personalizado que ejecuta el software patentado, que proporciona captura de movimiento de todo el cuerpo en 3D, reconocimiento facial y capacidades de reconocimiento de voz [38].

El sensor de Kinect reproduce video a una frecuencia de 30 Hz, en colores RGB 32-bit y resolución VGA de 640×480 pixels, el canal de video monocromo es de 16-bit, resolución QVGA de 320×240 pixels con hasta 65,536 niveles de sensibilidad. El límite del rango visual del sensor de Kinect está entre 1.2 y 3.5 metros de distancia, con un ángulo de vista de 57° horizontalmente y un ángulo de 43° verticalmente, mientras que el pivote puede orientarse hacia arriba o abajo ampliando hasta 27° . El array del micrófono tiene cuatro cápsulas, y opera con cada canal procesando 16-bit de audio con un ratio de frecuencia de 16 kHz.

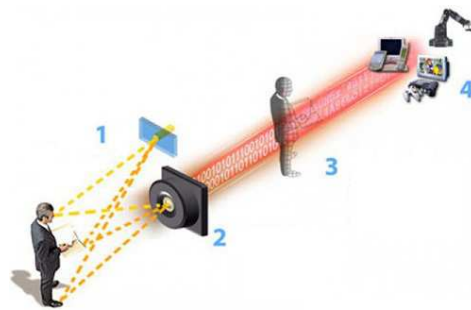


Figura 2.11: Como detecta objetos

Como se observa en la figura 2.11 se logra identificar el esquema de funcionamiento básico del sensor kinect, en 1 un foco de luz infrarroja ilumina al objeto que refleja la luz del sensor, en 2 el sensor del chip calcula la distancia a partir del tiempo de salida y llegada de la luz, en cada pixel, en 3 un software de imagen basado en un mapa de profundidad percibe e identifica los objetos en tiempo real y por último en 4 el dispositivo final reacciona a la señal.

El Kinect fue creado en principio como complemento al Xbox, cuya principal función es convertir al cuerpo en controles de juego en esta consola, capturando los movimientos del cuerpo, cabeza, pies, cara y voz, los cuales reemplazan por completo los controles convencionales de teclas. Funciona por medio de sensores, cámaras y micrófonos de avanzada tecnología patentados por su compañía creadora Microsoft.

El dispositivo Kinect fue configurada posteriormente por algunos desarrolladores que decidieron darle mayores aplicaciones, ofreciendo al público una serie de librerías para controlar parcialmente el dispositivo, y brindarle al mercado una amplia variedad de aplicaciones; todo lo anterior da pie a integrarlo al proyecto, ya que cuenta con lentes de color y sensación de profundidad; lo cual se ajusta a la necesidad de reconstruir ambientes virtualmente.

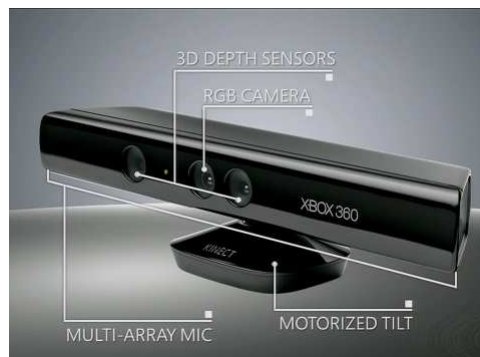


Figura 2.12: Funcionalidades del Kinect

Éste también cuenta con un campo de visión horizontal de 57 grados y vertical de 43 grados los cuales se adaptan a la plataforma móvil terrestre, ya que como se muestra en la figura(2.11-2.12), cubre con el rango de visión deseado para capturar la información del ambiente plano en el cual la plataforma tiene la posibilidad de recorrer sin problemas; y por último y una de las características más importantes, debido a su gran alcance en distancia visual es el que otorga el sensor de profundidad que logra tener un rango de 1,2 a 3,5 metros.

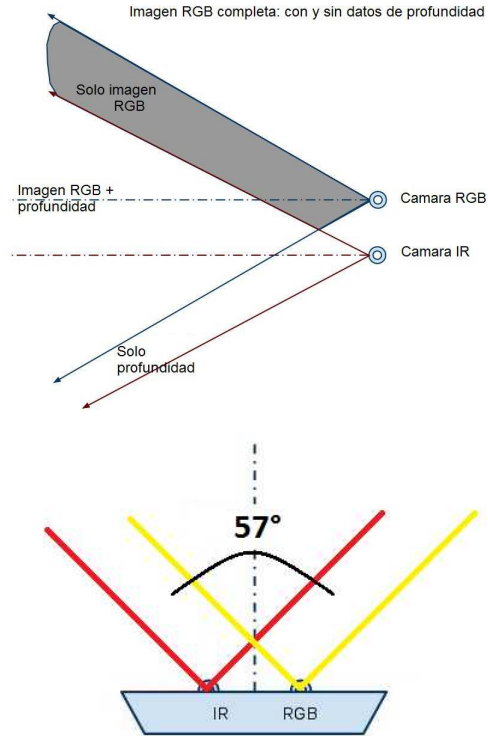


Figura 2.13: Rango de visión horizontal

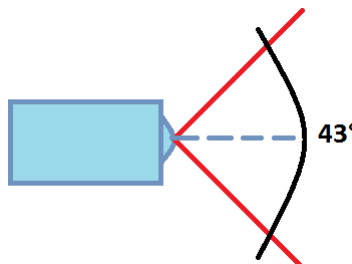


Figura 2.14: Rangos de visión vertical

Por otra parte este dispositivo posee un flujo de datos óptimo de 640x480 a 32 bits de color a 30fps el cual nos asegura una fiabilidad en los datos adquiridos. Como se puede apreciar en la figura 2.13, las características de hardware son realmente variadas y buenas para integrarlas como solución al problema de la ausencia de supervisión visual de la plataforma.

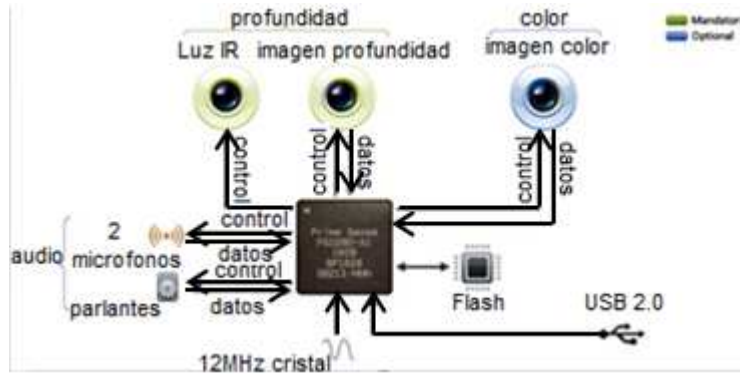


Figura 2.15: Hardware del Kinect

El sensor actúa como un sonar, la operación no es teóricamente complicada, si se conoce el tiempo de cada salida y llegada de la luz tras reflejarse en un objeto, sabiendo la velocidad absoluta de la luz, se puede tener la distancia a la cual se encuentra ese objeto. En un amplio campo visual con objetos, la cámara Kinect trata de reconocer a qué distancia están los objetos, distinguiendo el movimiento en tiempo real. Kinect puede llegar a distinguir la profundidad de cada objeto con diferencias de 1 centímetro y su altura y anchura con diferencias de 3 milímetros. El hardware de Kinect está compuesto por la cámara y el proyector de luz infrarroja, añadido al firmware y a un procesador que utiliza algoritmos para procesar las imágenes tridimensionales.

El procesador es capaz de interpretar los movimientos que se registran en los objetos capturados por la cámara de Kinect en eventos con significado que aparecen en pantalla. Los movimientos buscados por el algoritmo son contextualizados.

Para profundizar y recopilar más información de sensor Kinect, por favor referirse al Anexo 1 en el que se encuentra un manual del kinect.

Capítulo 3

IMPLEMENTACIÓN

En este capítulo se presenta la distribución del desarrollo de este proyecto. Se especifica cada uno de los procesos implementados, tales como la programación del dispositivo, análisis mecánico y de software, y la comunicación para el transporte y manipulación de la información que procesa el dispositivo, dando como resultado final la incorporación de un sistema de visión 3D que amplía las capacidades de navegación de la plataforma móvil terrestre que se elige como aplicación final.



Figura 3.1: Diagrama de bloques del sistema actual

A continuación se explicaran cada una de las partes que intervienen en el sistema, dando de esta forma una idea de lo que se va a desarrollar en cada una de las partes.

- *Dispositivo Mecánico-Modificación de la plataforma:* Ubicación estratégica del dispositivo sobre la plataforma buscando obtener la mejor panorámica visual del entorno; esto implica que ninguna parte de la plataforma móvil puede interferir dentro del rango visual del dispositivo, ya que esto crea formas y superficies inexistentes dentro del entorno.
- *Sensor de profundidad Kinect:* Se obtienen las distancias necesarias para generar las profundidades a partir de la emisión invisible de un láser, que genera puntos aleatorios sobre la superficie que permite interpretar la

distancia, el cual está programado en base a un conjunto de librerías, las cuales facilitan la manipulación y transformación de los datos obtenidos.

- *Generación de la nube de puntos*: Los datos almacenados son transformados por medio de la programación generada en un compilador de lenguaje de máquina, en el cual se logra obtener una secuencia de imágenes con puntos en común que recopiladas se unen para formar el ambiente.
- *Interfaz usuario*: El usuario cuenta con una aplicación en la cual puede navegar libremente por el espacio reconstruido, teniendo la posibilidad de medir objetos y distancias, con gran precisión realizando un reconocimiento del espacio recorrido por la plataforma.

3.1. Dispositivo Mecánico

Se utilizó la plataforma móvil terrestre, ver figura 3.2, construida como proyecto de grado en la Universidad Militar Nueva Granada por Juan Camilo Hernández en el año 2009 [39].

Esta plataforma móvil terrestre posee características como la tracción por orugas, al igual, es teleoperado mediante un radio control.

Tiene un espacio para cargas de 40cm X 50cm X 20cm, esto con el fin de embarcar la electrónica y fuentes de energía necesarias para el funcionamiento del robot, y espacio suficiente para el desarrollo de futuras aplicaciones con esta plataforma.

La plataforma está diseñada para sortear obstáculos no superiores a los 15 cm de altura, por esta razón posee unas orugas de 72 cm de largo y con una inclinación de 142°; el sistema de orugas consta de 14 ruedas (7 por oruga), 5 rodillos sobre los que se desliza la correa y dos poleas (una conductora y otra conducida).

Ésta plataforma se controla mediante un radio control Futaba, el cual cuenta con seis canales para servomotores, típicamente este radio control es muy usado en aeromodelismo, carros etc. Este control aplica para este proyecto básicamente por la distancia que puede alcanzar, debido que se espera tener tele operación en un rango de un kilómetro. La adquisición de datos de este control consiste en poder codificar el ancho de pulso que entrega el receptor, el receptor proporciona un tren de pulsos a una frecuencia de 60 Hz, cada ancho de pulso tiene un rango de un milisegundo, siendo un milisegundo el valor de ancho de pulso en su valor mínimo y dos milisegundos en el valor máximo, también hay que tener en cuenta el valor pico de voltaje que entrega el receptor que es de tres voltios, este valor permite hacer una conexión directa con el micro controlador sin necesidad de otro elemento electrónico ya que el valor de voltaje está en el rango de un 1 lógico [39].



Figura 3.2: Plataforma móvil terrestre y Radio control Futaba

■ *Integración del dispositivo a la plataforma*

Para lograr el acople ideal entre la plataforma y el dispositivo de visión 3D, es necesario identificar las distancias en las que la visión este totalmente libre, esto quiere decir que ninguna parte de la plataforma puede quedar dentro del rango visual del dispositivo; si dentro del rango visual del sensor aparece alguna parte de la plataforma, esto crea superficies constantes e inexistentes dentro de la recreación del ambiente.

Al igual, se hizo necesario crear, diseñar e implementar una estructura sencilla para soportar el Kinect y lograr ajustar la altura y distancia ideal en la plataforma móvil terrestre; para ello se diseñó el soporte que consta de los siguientes elementos:

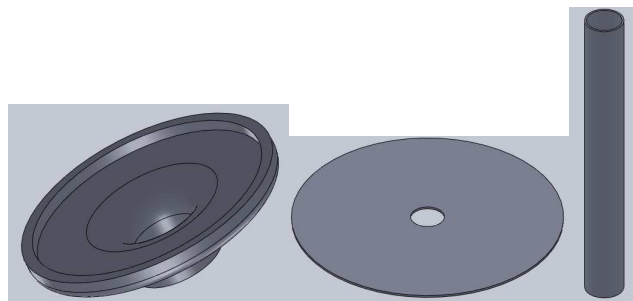


Figura 3.3: Elementos del soporte



Figura 3.4: Estructura completa mas Kinect

Para no afectar el rendimiento mecánico y de potencia de la plataforma ya existente, fue necesario implementar el soporte con materiales ligeros que no aumenten significativamente el peso de la plataforma, los materiales están compuestos de polímeros ligeros como lo son el PVC, Policarbonato y aluminio, estos materiales son resistentes y a la vez muy ligeros.

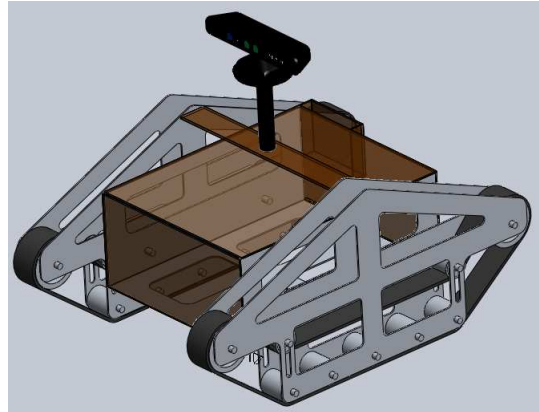


Figura 3.5: Integración del dispositivo

Ya con esta estructura implementada, el Kinect se encuentra a una altura de 627.08 mm (62.708 cm) con respecto al suelo, y a una distancia del frente del espacio de carga de 252.96 mm (25.296 cm); estas medidas permiten lograr

una panorámica en la que no interactúa ninguna parte de la plataforma móvil.

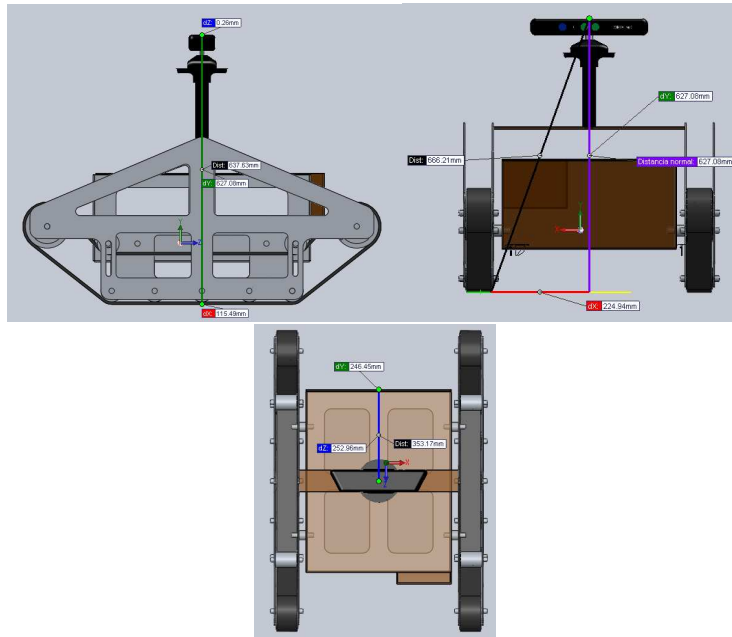


Figura 3.6: Disposición del sistema

También se creó un compartimento en acrílico, que va sujeto en la parte trasera del espacio para cargas de la plataforma móvil, este soporte está diseñado para sujetar exteriormente al router D-LINK DIR-600 encargado de la comunicación del dispositivo.

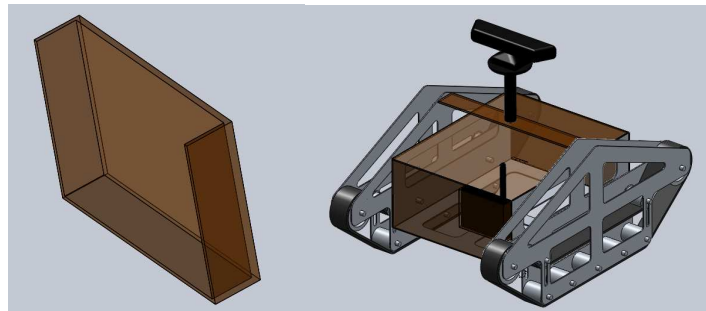


Figura 3.7: Compartimento ROUTER

Por otra parte se creó una base para el computador que va sobre la plataforma, el cual ira almacenando la información suministrada por el sensor.

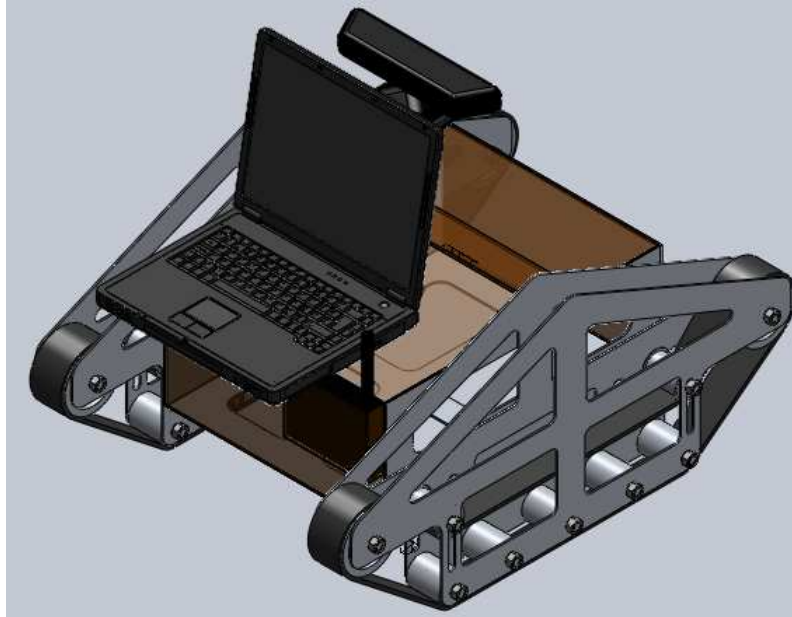


Figura 3.8: Ensamble final.

Los planos de todos estos diseños se pueden encontrar en el ANEXO 2.

3.2. Conectividad del Sensor de profundidad (Kinect)

El sensor Kinect es una cámara que reconstruye en 3D la escena que observa. Es un dispositivo que creó Microsoft para su consola de videojuegos Xbox. A pesar de ser un aparato comercial diversas comunidades de desarrolladores de software crearon drivers open source hackeando esta cámara para poder utilizarla con cualquier sistema operativo (Mac OSX, Linux o Windows).

La librería OpenNI, permite comunicarse con los sensores de audio, video y sensor de profundidad de Kinect, mientras que proporciona una API que sirve de puente entre el hardware del equipo, NITE Middleware y las aplicaciones e interfaces del S.O. La idea es facilitar el desarrollo de aplicaciones que funcionen con interacción natural, como gestos y movimientos corporales.

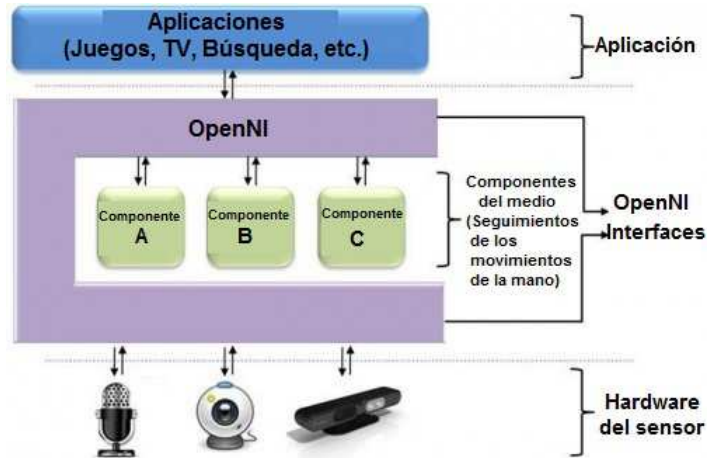


Figura 3.9: Librería OpenNi

Actualmente OpenNI permite la captura de movimiento en tiempo real, el reconocimiento de gestos con las manos, el uso de comandos de voz y utiliza un analizador de escena que detecta y distingue las figuras en primer plano del fondo.

- **Windows XP / Windows 7**

En Windows XP y Windows 7 el Kinect se instala de igual forma.

1. Se descargan los siguientes archivos:

Nota: toda la instalación se realizó con las últimas versiones (inestables), sin embargo, la instalación se realiza de igual forma para versiones estables. Es importante que si se escoge instalar la versión estable, los tres archivos deben tener la versión estable. Análogamente para la versión inestable. En teoría se pueden hacer combinaciones, sin embargo, es muy posible que finalizada la instalación, el Kinect no funcione.

OpenNI: <http://www.openni.org/downloadfiles/openni-binaries/20-latest-unstable>

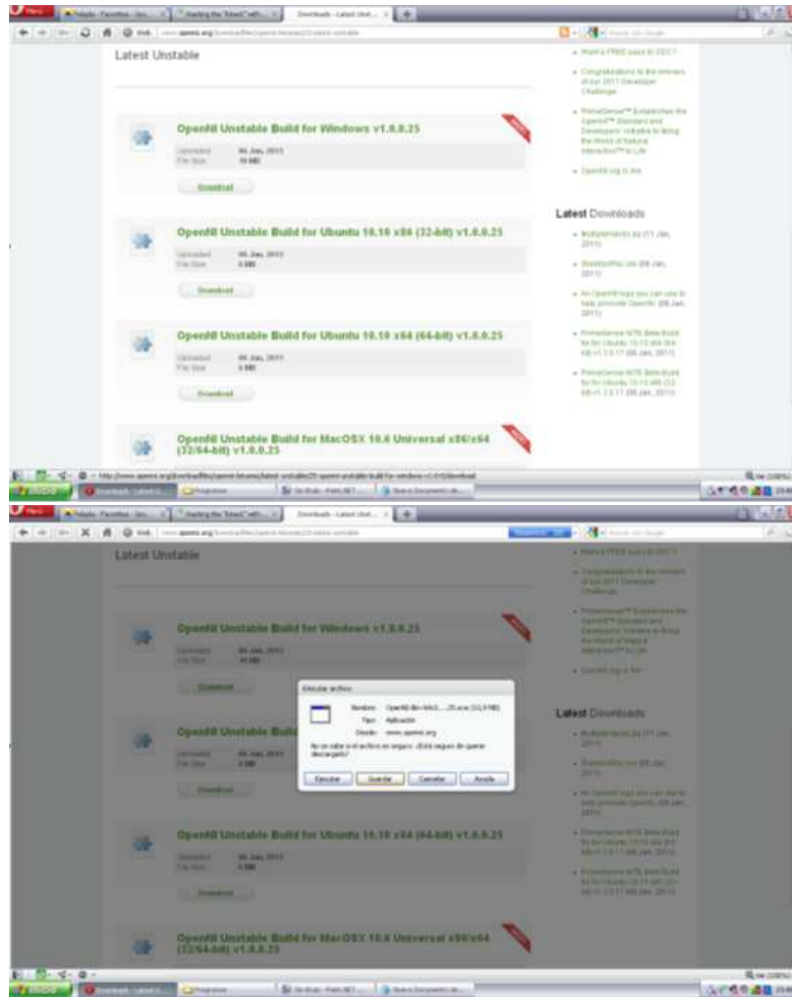


Figura 3.10: <http://www.openni.org/downloadfiles/openni-binaries/20-latest-unstable>

Kinect drivers: <https://github.com/avin2/SensorKinect>

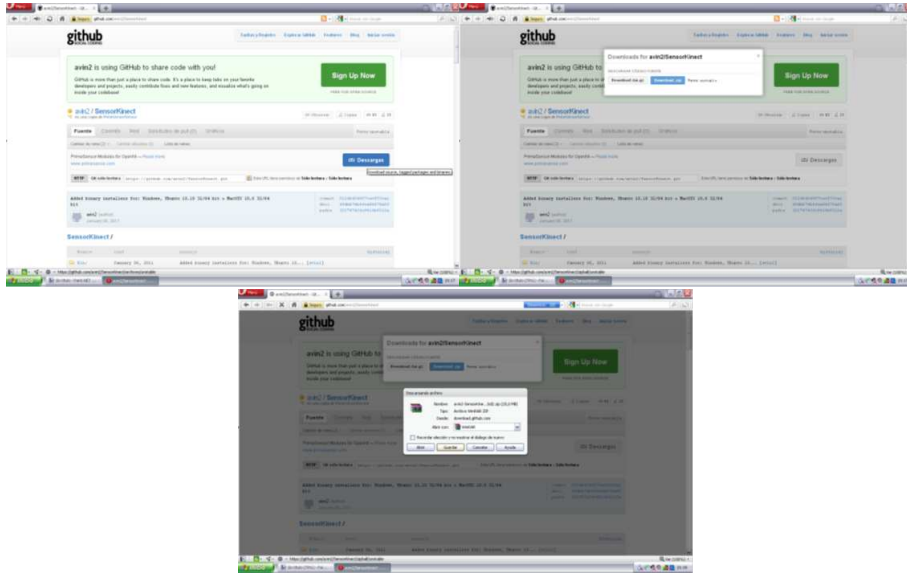


Figura 3.11: <https://github.com/avin2/SensorKinect>

NITE:<http://www.openni.org/downloadfiles/openni-compliant-middleware-binaries/33-latest-unstable>



Figura 3.12: <http://www.openni.org/downloadfiles/openni-compliant-middleware-binaries/33-latest-unstable>

Una vez descargados los 3 archivos, se descomprime **avin2-SensorKinect-0124bd2**.

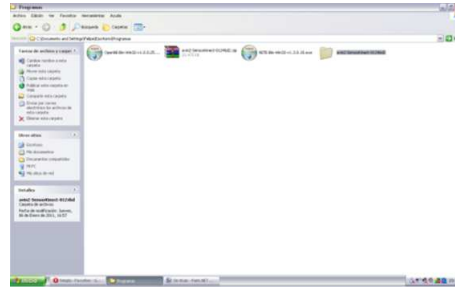


Figura 3.13: avin2-SensorKinect-0124bd2

2. Se instalan los archivos recién descargados en el siguiente orden: **OpenNI-Bin-Win32-v1.0.0.25.exe**



Figura 3.14: OpenNI-Bin-Win32-v1.0.0.25.exe

Para instalar el Sensor, se debe ir a la carpeta descomprimida **avin2-SensorKinect-0124bd2** y entrar a **Bin**. Ahora se ejecuta **SensorKinect-Win32-5.0.0.exe** y se instala normalmente.

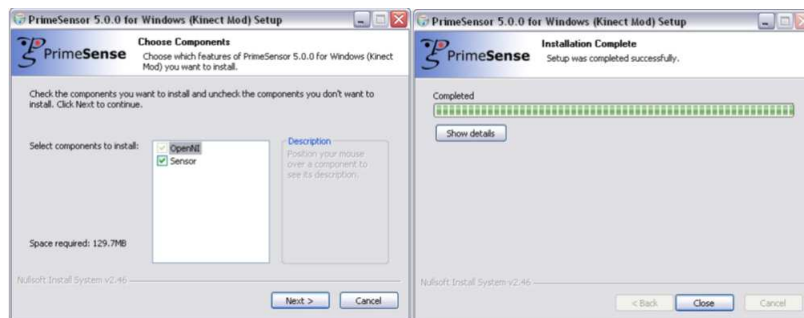


Figura 3.15: SensorKinect-Win32-5.0.0.exe

Por último se instala NITE-Bin-Win32-v1.3.0.18.exe.

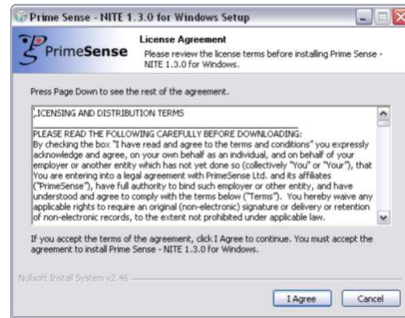


Figura 3.16: NITE-Bin-Win32-v1.3.0.18.exe

En esta parte el programa pide el número de licencia, por lo que se ingresa: **0KOIk2JeIBYCIpWVnMoRKn5cdY4=** ; este número de licencia es válida para todas las versiones.

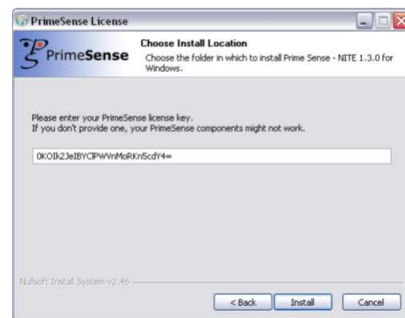


Figura 3.17: 0KOIk2JeIBYCIpWVnMoRKn5cdY4=

3. Ahora se conecta el Kinect. En este punto aparece automáticamente el Asistente para hardware encontrado, el cual nos indica que encontró el Motor, el Audio y la Cámara del Kinect.

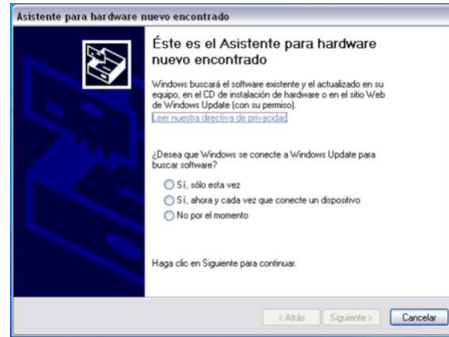


Figura 3.18: Asistente para hardware encontrado

Se selecciona la primera opción: **Si, solo esta vez** En la ventana posterior, se selecciona **Instalar automáticamente el software (recomendado)**.

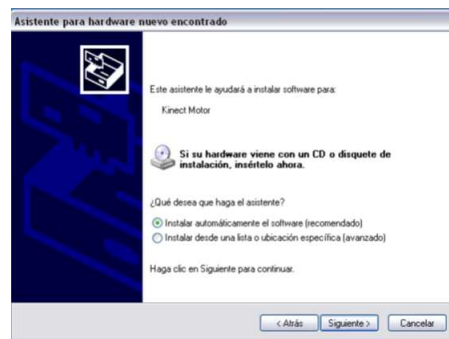


Figura 3.19: Instalar automáticamente el software (recomendado)

Con esto Windows instala correctamente el driver del motor del Kinect.

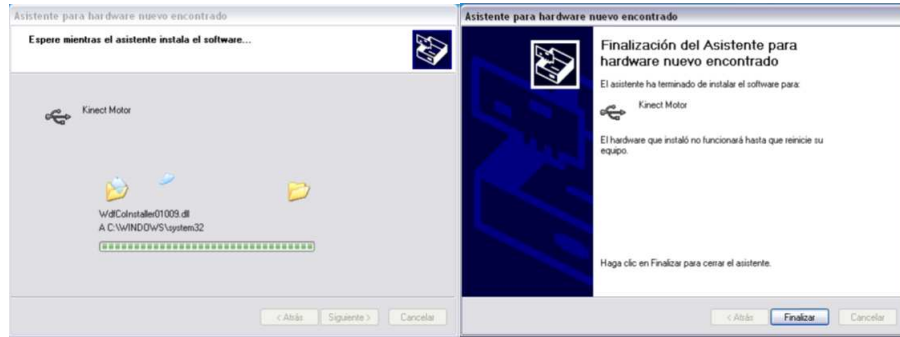


Figura 3.20: Instalación correcta.

Luego de instalar el motor, aparece que encontró el Audio, sin embargo, Windows no instala este driver. Esto no es problema ya que no es necesario. Finalmente aparece la cámara, la cual se instala análogamente al driver del motor.

Si se observa el administrador de dispositivos se debería ver algo por el estilo:

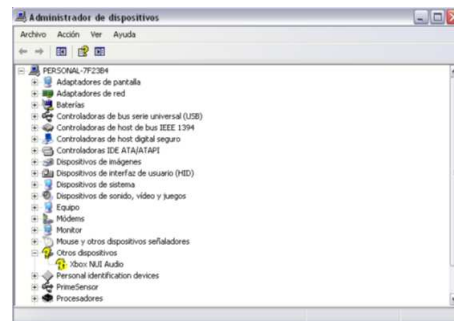


Figura 3.21: Error de configuración

Observación: si por algún motivo Windows no reconoce los drivers, éstos se pueden instalar manualmente desde la carpeta **avin2-SensorKinect-0124bd2/Plataform/Win32/Driver**, instalando **dpinst-x86.exe** o **dpinst-amd64.exe** dependiendo si la arquitectura es 32 ó 64 bits.

4. El último paso antes de poder probar el Kinect es la modificación de 3 archivos XML (**Sample-Scene.xml**, **Sample-Tracking.xml** y **Sample-User.xml**).

Éstos se encuentran en la siguiente dirección: **C:\Archivos de Programas\Prime Sense\NITE\DATA**

Hay que realizar la misma modificación para cada archivo. Con el editor

preferido se abre el archivo **Sample-Scene.xml** y se buscan las siguientes líneas:

```
<License vendor="PrimeSense" key="Insert Key"/>  
<MapOutputMode xRes="320" yRes="280" FPS="30"/>
```

Las cuales se reemplazan correspondientemente por:

```
<License vendor="PrimeSense" key="0KOIk2JeIBYCIWVnMoRKn5cdY4="/>  
<MapOutputMode xRes="640" yRes="480" FPS="30"/>
```

Luego se guarda el archivo y se repiten los mismo paso con los archivos **Sample-Tracking.xml** y **Sample-User.xml**.

Finalmente se pueden probar los ejemplos y comprobar que la instalación fue exitosa. Los ejemplos que OpenNI trae por defecto se encuentran en **C:\Archivos de programa\OpenNI\Samples\Bin\Release**



Figura 3.22: Ejemplo del OpenNi

También se pueden probar los ejemplos que trae NITE, los cuales se encuentran en **C:\Archivos de programa\OpenNI\Samples\Bin\Release**

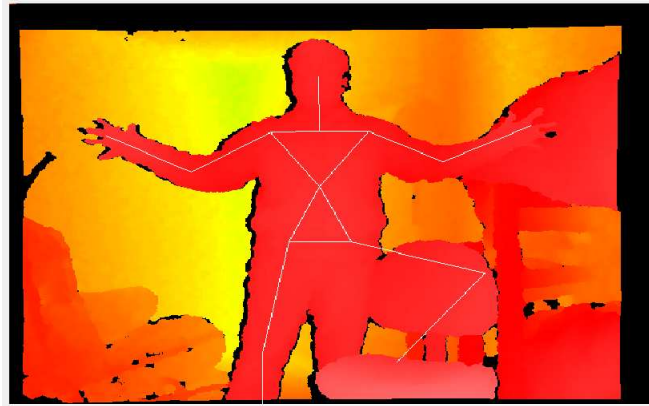


Figura 3.23: Ejemplo NITE

La posición mostrada en la imagen 3.22, es la posición de calibración que utiliza Kinect para posteriormente realizar tracking.

Observación: En algunos casos los ejemplos no funcionan o funcionan parcialmente cuando son ejecutados directamente, si esto pasa se debe ejecutar directamente desde la consola de comandos de Windows. Si estos test no funcionan significa que algo ha salido mal, así que es necesario repetir los pasos anteriores.

EJECUCION DE APLICACIONES

Para ejecutar cualquier aplicación gratuita se deberá ir a la carpeta contenedora de dichos programas, donde inmediatamente se ejecutaran los comandos de cámara para poder probar la adquisición de datos.

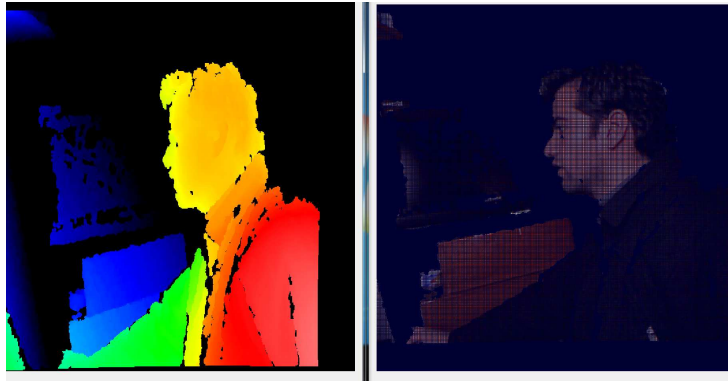


Figura 3.24: Visualización inicial con infrarojo

El sensor Kinect luego de ser conectado verifica automáticamente si los pasos

de instalación quedaron bien hechos y él de inmediato se ejecutara por consola el ejemplo de tracking, el cual muestra como a través de los movimientos de las manos se controlan las manecillas de un reloj.

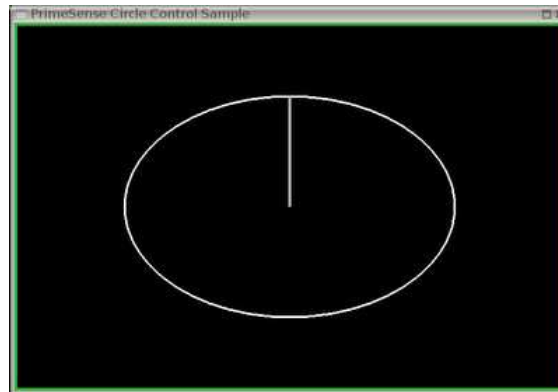


Figura 3.25: Control del reloj (tracking)

Otra aplicación que puede ser de gran ayuda a la hora de mover objetos en un ambiente real es a través del movimiento de las manos controlar los cursores del mouse o del teclado, según sea la aplicación futura a la cual este destinada.

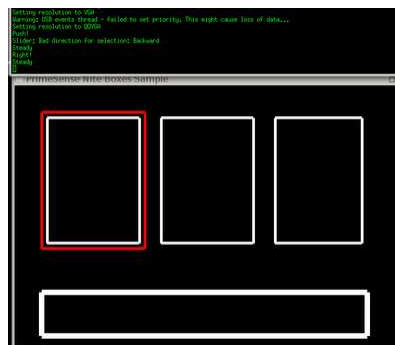


Figura 3.26: Tracking teclado y mouse

Por último se encuentra una aplicación de mapas de profundidad básico, el cual genera una apariencia de profundidad a una foto capturada por el sensor que es mostrada en un plano tridimensional.



Figura 3.27: Escena tridimensional

3.2.1. Documentación de los formatos de salida de la información entregada por el dispositivo

El driver entregado por Primesense permite obtener; depthMap, mapa IR y mapa RGB.

Un depthMap corresponde a una matriz escalonada de píxeles de profundidad. Esto quiere decir, que una “matriz 1D” (arreglo) representa una matriz 2D. Cada píxel de profundidad es de 2 bytes y representa la distancia en milímetros de ese píxel al sensor.

El número de píxeles está determinado por la resolución. Estos datos de profundidad se pueden obtener por medio del objeto llamado DepthMetaData.

Una imagen RGB se compone de Rojo, Verde y Azul. Cada uno de estos colores es de 8 bits, por lo que un píxel tiene 24 bits, que equivale a 3 bytes.

Por otro lado, una imagen del sensor infrarrojo, se compone de 16 bits en escala de grises.

Para saber cómo trabajar con OPENNI y OpenCV en conjunto, se recomienda leer el pequeño manual adjunto a continuación: http://opencv.jp/opencv-2svn_org/opencv_user.pdf

3.2.2. Rendimiento

Las pruebas se realizaron en un Notebook DELL INSPIRON15, con sistema operativo Windows 7 Professional SP1.

Es necesario tener unos requerimientos mínimos para el correcto funcionamiento del sistema, ya que la activación del sensor, la generación de nubes de puntos y creación de los mapas de profundidad exigen un rendimiento de máquina óptimo por el tamaño de los archivos generados.

Las características mínimas deseables del sistema son las siguientes:

Procesador: Intel(R) Core(TM)2 Duo T6400 @ 2.00GHz 2.00 GHz

Memoria RAM: 4,00 GB (3,46 GB utilizable)

Tipo de sistema: Sistema operativo de 32 bits

3.3. Generación de la nube de puntos y extracción del mapa de profundidad

Kinect Windows SDK Beta como se conoce esta herramienta, cuenta con Raw Sensor Streams el cual permite tener acceso a los sensores de color y profundidad pero a bajo nivel para construir a partir de ellos; teniendo en cuenta lo anterior, se generó un diagrama de flujo el cual representa el proceso de reconstrucción 3D que se aplica en este proyecto.

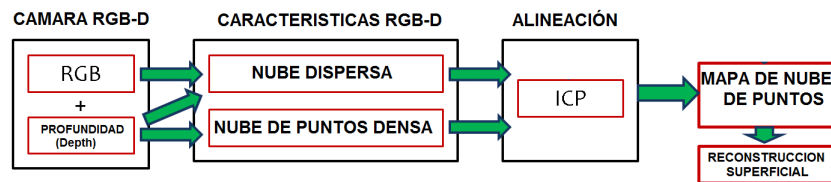


Figura 3.28: Característica general del mapa RGB-D.

El algoritmo se basa en la adquisición de los datos RGB y de profundidad mediante los sensores que se encuentran dentro de la arquitectura del Kinect, para formar tanto nubes densas como nubes dispersas y formar superficies más definidas, este tipo de reconstrucción adopta el nombre de triangulación óptica con luz estructurada.

El Kinect trabaja una cámara RGB que obtiene imágenes en color y 2 cámaras de infrarrojos para medir la distancia a la que se encuentran los elementos



Figura 3.30: RGB de imagen

del ambiente que están dentro del campo de visión .

Por medio del SDK de Kinect para Windows, podemos obtener los datos de las cámaras y trabajar con ellos para utilizarlos en nuestras aplicaciones.

Las imágenes que se obtienen del sensor se almacenan en un vector de bytes. Las imágenes se componen de un conjunto de píxeles. Cada pixel de la imagen tiene 4 componentes que representan los valores de los colores rojo, verde y azul más una componente que corresponde con el valor de transparencia (alfa), en el caso de imágenes RGBa, o un valor vacío, si es de tipo RGB.

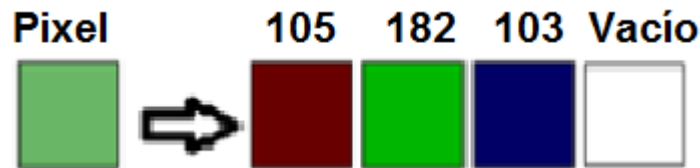


Figura 3.29: Píxel

Cada componente del píxel tiene un valor decimal de 0 a 254 lo que corresponde a un byte. De esta forma el vector de bytes que obtenemos del sensor, en el caso de la cámara RGB, es una representación de esos píxeles organizados de arriba abajo y de izquierda a derecha donde los 4 primeros elementos del vector serán los valores rojo, verde, azul y alfa del píxel de arriba a la izquierda mientras que los 4 últimos serán del píxel de abajo a la derecha.

Cuando se utilizan las cámaras de profundidad el procedimiento varía. Al igual que con la cámara RGB también obtendremos un vector de bytes pero en esta ocasión esos bytes no corresponden con los valores de los componentes de un píxel sino con la distancia del píxel al sensor.

Teniendo las dos cámaras de infrarrojos, cada píxel se corresponde con 2 bytes en el vector, siendo éstos el valor de la distancia de ese píxel a cada

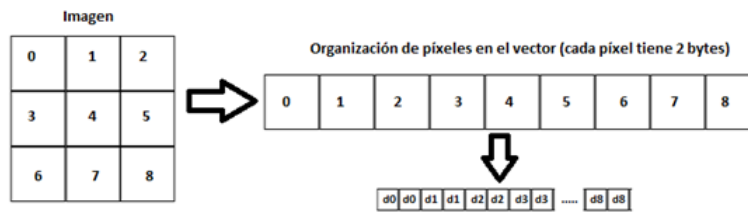


Figura 3.31: Profundidad de la imagen

cámara. La organización de los píxeles es la misma que con la cámara RGB, los 2 primeros bytes es la distancia del píxel de la posición de arriba a la izquierda al sensor y los 2 últimos son del píxel de abajo a la derecha.

Los datos que se obtienen del sensor se encuentran codificados, estos sirven para saber qué tan lejos se encuentra un píxel del sensor, con lo que se tiene que calcular la distancia a partir de esos datos. Para calcular la distancia debemos de realizar una serie de operaciones, para estas operaciones se utiliza el método Depth ya que simplemente se tienen los datos de profundidad, para calcular la distancia de un píxel en una imagen tipo Depth hay que hacer una operación de lógica OR con los bytes correspondientes al píxel, realizando antes un desplazamiento de 8 bits en el segundo byte; el rango de distancias que acepta el sensor es de 850 mm a 4000 mm.

Obtenidos los datos de la cámara RGB y la de profundidad se generan nubes de puntos densas y dispersas según el tiempo de muestreo o retardo entre toma y toma, las cámaras recopilan información a 30fps, y cuyo tiempo de muestreo del ambiente es de 41ms el cual sumado al tiempo del proceso y el movimiento del Kinect es de 200 a 300 ms.

Con la nube de puntos generada de cada una de las muestras, se implementa un algoritmo mediante el cual se alinean dos nubes de puntos tridimensionales, expresadas en distintas referencias y correspondientes a un objeto dado. El objetivo es encontrar la matriz de rotación y el vector de traslación que aplicados a una nube de puntos, nos permitan alinearla con la otra.

El algoritmo que se implementa es el de ICP (Iterative Closest Point), requiere localizar el punto más cercano de una nube de puntos a un punto dado de otra nube. El algoritmo ICP siempre converge monótonamente a un mínimo local de una distancia cuadrática media. La tasa de convergencia es rápida durante las primeras iteraciones. Dado un conjunto adecuado de estados iniciales de rotaciones y traslaciones se puede minimizar globalmente la distancia cuadrática media.

En este caso es necesario realizar múltiples muestras para alinearlas entre sí,

al tomar varias muestras implica aumentar el tiempo empleado en la ejecución del programa, y por otro lado, hacerlo robusto en presencia de puntos extraños (outliers).

Y por último se repite el muestreo de forma consecutiva utilizando el algoritmo IPC para lograr la unión de diferentes imágenes de un mismo ambiente.

Este algoritmo de reconstrucción involucra dos fases: una es la medida de la superficie por medio de un escaneado 3D y la otra, es una reconstrucción de los datos adquiridos.

Primero que todo todas las nubes de puntos que constituyen cada imagen deben ser alineadas en el mismo sistema de referencia, y este es el objeto central del algoritmo IPC. Posteriormente se integran las imágenes y se crea un modelo no repetido en el que no hay superficies ocultadas.

Como punto de partida se tienen dos nubes de puntos RI y RII. Se pretende alinear RI con RII. Para ello es necesario encontrar una matriz de rotación y un vector de traslación, denotando al conjunto como T_{LMS} , tal que $T_{LMS}(R^I)$ se alinee con R^{II} . El algoritmo pretende encontrar iterativamente este valor de T_{LMS} .

1. Inicialización: $T_{LMS} \leftarrow T_0$. El programa permite una inicialización ejecutada por el usuario y con el O_0 siempre en el kinect.
2. Un conjunto de N_S puntos P_{RS}^I es extraído de R^I de forma aleatoria: $P_{RS}^I \leftarrow RS(R^I, N_S)$.
3. Se probarán distintas rotaciones y traslaciones iniciales en este paso, es decir repetir los pasos a), b), c) que seguidamente describimos en el caso de que en el paso 1 no se haya elegido la opción de inicialización. En caso contrario se probará únicamente con la traslación y rotación inicial introducida por el usuario.
 - a) El conjunto de puntos P_{RS}^I es usado por el algoritmo ICP con la segunda imagen RII para estimar los parámetros del movimiento. $T_{ICP,n} \leftarrow ICP(P_{RS}^I, T_{LMS}, R^{II})$.
 - b) El movimiento estimado es evaluado por $dticp(P_{RS}^I, T_{ICP,n}, R^{II})$.
 - c) Si $dticp(P_{RS}^I, T_{ICP,n}, R^{II}) < dtlms(R^I, T_{LMS}, R^{II})$, siendo $dtlms$ la función que nos marca un valor para la distancia entre P_{RS}^I y R^{II} , entonces se sustituye $T_{LMS} \leftarrow T_{ICP,n}$. Se observan ventajas en tiempo de ejecución a la hora de comparar en la distancia $dticp$ únicamente la muestra de R^I, P_{RS}^I , en vez de utilizar toda la imagen R^I .
4. Repetimos los pasos 2-3 tomando n de 1 hasta N_T , siendo N_T es el número de pruebas.

El movimiento resultante T_{LMS} satisface $\text{dtlms}(P_{RS}^I, T_{LMS}, R^{II}) = \min_{1 \leq n \leq N_T} \text{dticp}(P_{RS}^I, T_{ICP,n}, R^{II})$ y este hecho constituye el resultado de la estimación de movimiento dtlms entre R^I y R^{II} .

Por último es necesario aplica el método de Delaunay, para lograr realizar una triangulación de los puntos en las superficies para lograr una superficie solida de la imagen final.

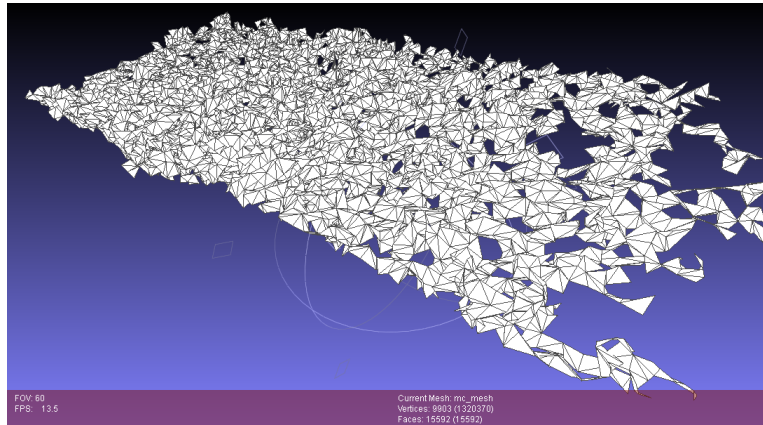


Figura 3.32: Triangulación por Delaunay

3.4. Interfaz usuario

MeshLab le proporciona al usuario en un sólo programa, una gran cantidad de utilidades (tools) muy completas para la edición, limpieza, reparación, inspección, renderización y conversión de formatos en este tipo de estructuras de mallas. MeshLab es un sistema totalmente gratis y de código abierto que sirve para procesar y editar mallas triangulares no estructuradas, soportando los formatos más habituales (PLY, STL, OFF, OBJ, 3DS y COLLADA), además de los filtros más comunes (quitar duplicados, vértices sin referencias, caras nulas, etc.), división de superficies y suavizado de la malla resultante. El sistema se ayuda en la librería pública VGC para ejecutar las tareas de procesamiento de mallas ‘vcg.sf.net’, y su objetivo principal es ayudar y facilitar el tratamiento de modelos no estructurados de gran tamaño que provengan de un escaneo 3D. En la última versión se han añadido mejoras en el entorno gráfico de modo que ahora se puede cambiar el color de parte de la malla para diferenciar las partes de la que está compuesta.

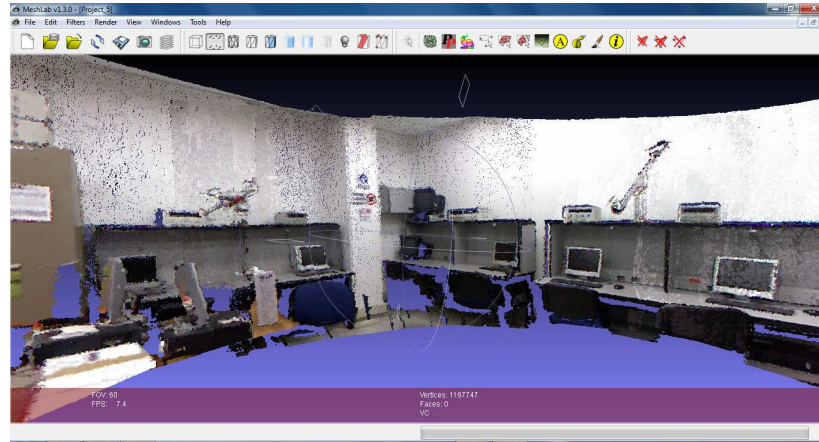


Figura 3.33: Interfaz MeshLab

Con esta herramienta de software, el usuario tiene la posibilidad de interactuar con el ambiente en 3D, logrando navegar por él, medir superficies y modificar las texturas de los archivos.

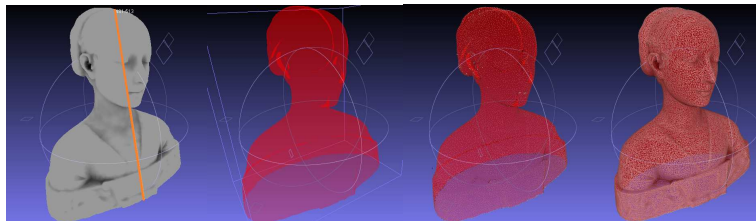


Figura 3.34: Texturas diferentes.

Al iniciar MeshLab, se activa automáticamente el modo de cámara de la interfaz, el usuario puede navegar por el entorno utilizando las funciones de mouse que se utiliza para alejarnos, acercarnos y rotar el objeto, el modo cámara se puede desactivar para habilitar otras funciones como por ejemplo modificar la malla, solido o nube de puntos, con el modo cámara desactivado las funciones del mouse cambian para permitir modificar los objetos.



Figura 3.35: Control de la interfaz.

Herramientas interactivas de edición:

Estas herramientas se pueden activar desde el menú de edición o de la barra de herramientas (todos los botones a la derecha). La activación de una de estas herramientas por lo general cambia la forma de interactuar con la malla, las acciones del ratón se procesan directamente en la herramienta elegida.

Capítulo 4

ACCESO AL SISTEMA

En el desarrollo del proyecto se generan dos alternativas diferentes de entrada a los datos y a la visualización, dando como resultado dos tipos de comunicación para diferentes aplicaciones; teniendo esta idea, es de donde se derivan dos alternativas que son la comunicación por Red de área local (LAN) e Internet, las cuales permiten acceder de forma remota a los datos pero con diferentes características, aplicaciones y tiempos.

4.1. Acceso por medio de red de área local (LAN)

La red de área local realiza una interconexión entre computadores y periféricos. Su extensión puede ser limitada físicamente a un edificio o a un entorno máximo de 200 metros; la aplicación realizada para el proyecto está creada para interconectar la plataforma móvil terrestre y la estación de trabajo del usuario.

Por medio de esta Red de área local el usuario tiene la posibilidad de conectarse a la plataforma móvil y extraer la información del ambiente captada por el sensor; Esta red LAN ésta configurada para operar de 10Mbps o a 100Mbps.

Por medio de la utilización de la aplicación de Windows para la conexión a escritorio remoto, el usuario tiene la posibilidad de manipular el computador que se encuentra instalado en la plataforma móvil, observando de estar forma el escritorio y por ende el proceso de reconstrucción, también puede acceder a la interfaz gráfica para que pueda manipular, modificar y navegar con el entorno reconstruido, esto con el fin de que en la estación del usuario no se cuente con los programas y aplicaciones necesarias para visualizar el proceso.

En esta comunicación tipo LAN, la plataforma móvil navega por el entorno manejado a control remoto por el usuario, la plataforma realiza la respectiva adquisición de datos, almacenándolos en el procesador CPU, al mismo tiempo el usuario puede acceder remotamente en tiempo real a la visualización generada

en 3D ejecutando la aplicación en la CPU de la plataforma móvil, al igual puede tener acceso al archivo PLY generado por el sistema de reconstrucción y transportar los datos desde la plataforma móvil terrestre hasta la estación remota fija.

En esta red de área local es posible interconectar varios computadores a la plataforma móvil, para que varios usuarios, si es necesario, tengan acceso a la información, es posible restringir la información a otros usuarios, se puede lograr que solo tengan acceso a una determinada información o simplemente tengan solo visualización; es importante señalar que el número de usuarios presentes en la red LAN tiene una influencia notable sobre el rendimiento que los usuarios puedan esperar.

Para conectarse a un equipo remoto, dicho equipo debe estar encendido, tener una conexión de red, el Escritorio remoto debe estar habilitado, debe tener acceso de red al equipo remoto y tener permiso para conectarse. Para obtener permiso para conectarse, debe estar en la lista de usuarios lo que mejora notablemente la seguridad de los datos almacenados y la transformación o manipulación de los mismos.

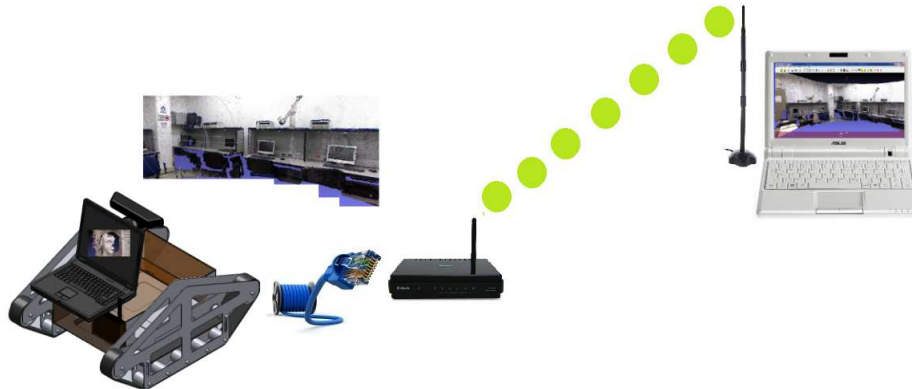


Figura 4.1: Red de área local (LAN)

En la comunicación de tipo LAN en la que el dispositivo se encuentra conectado a un computador y esta a su vez conectado a un router el cual por configuración permite el traslado de información a otro computador tiene como ventaja que los datos o información que se transfieren en dicho proceso pueden ser transmitidos y recepcionados hasta la estación remota fija en un mínimo de tiempo el cual permite ir almacenado mayor cantidad de bits, lo cual tiene como ventaja que se puede usar todo el ancho de banda para una sola aplicación, cuya ventaja es considerada principalmente por no tener ningún tipo de ruptura lo cual garantiza la fiabilidad de la reconstrucción.

Todo lo anterior se ve reflejado en como la red LAN es aplicada a la su-

pervisión en la información que se genera trama a trama en un ambiente, esto quiere decir que el usuario posee la capacidad vigilar el estado de un ambiente en cierto periodo de tiempo.

4.2. Acceso por medio de Internet (WAN)

En esta conexión se utiliza como flujo y transporte de datos una red WAN de tipo internet, la plataforma móvil tiene acceso a internet al igual que la estación remota fija, esto implica que cada uno posee una dirección IP, con esto ya es suficiente para direccionar las IP de cada equipo y de esta forma lograr enlazar tanto el Host como el cliente.

Esta red, al igual que la red LAN, permite al usuario acceder a escritorio remoto y a los datos que se encuentren en el host, esto ayuda al usuario a trasportar datos desde la plataforma móvil terrestre hasta la estación remota fija; cuando el cliente no posee el software necesario para correr la aplicación o para visualizar el archivo PLY generado en el host, el usuario por medio del escritorio remoto logra visualizar en tiempo real la reconstrucción del ambiente en 3D, al igual tiene acceso al archivo PLY generado por el programa para que pueda abrirlo en la interfaz gráfica de Meshlab que está instalado en el procesador (PC) de la plataforma, esto en el caso en que la plataforma móvil fuese autónoma o incluso si fuese controlada a control remoto por otra persona y a distancias extremadamente grades como por ejemplo de una ciudad a otra o incluso una distancia mayor.

Para que este tipo de red funcione, tanto el host como el cliente tienen que estar conectados a internet al mismo tiempo, y por medio de una herramienta de software libre llamado LogMeIn se sincronizan ambos ordenadores con sus respectivas direcciones IP, logrando de esta forma crear el enlace de comunicación. Por medio de LogMeIn se logra transmitir de lado a lado los archivos y acceder al escritorio remoto de la plataforma.

La idea de implementar la red WAN es que el usuario sea capaz de acceder a la plataforma móvil desde cualquier punto geográfico y manipular sus datos.

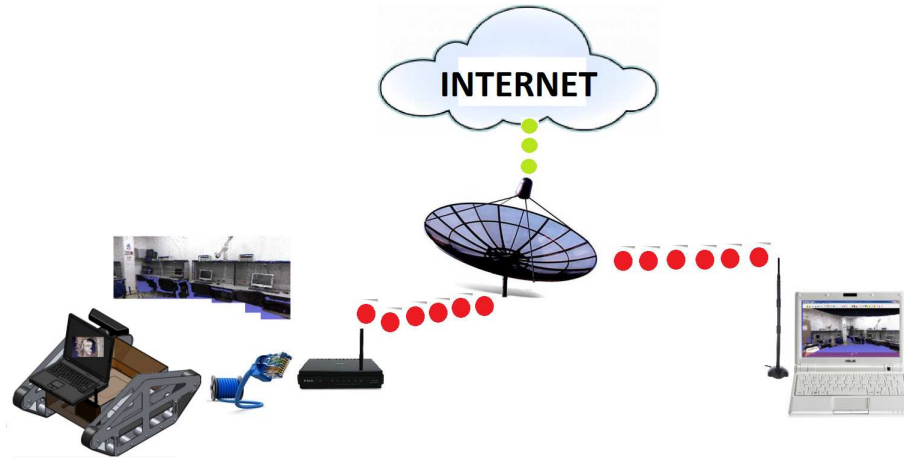


Figura 4.2: Internet

Actualmente con el avance de la tecnología y las redes de comunicación es posible tener el mundo en una sola ventana, lo cual se ve representado en las aplicaciones en donde se muestran lugares geográficos lejanos, en donde el usuario se traslada a otro lugar simplemente conectando un ordenador a una red amplia de comunicación, lo cual permite acceder a información que se encuentra a miles de kilómetros.

Capítulo 5

PRUEBAS, RESULTADOS

En este capítulo se muestra los resultados experimentales y aspectos que se tuvieron en cuenta en la reconstrucción de los ambientes que fueron puestos a prueba, en este caso, los pasillos de la Universidad Militar Nueva Granada y el laboratorio de robótica e inteligencia artificial.

Como resultado al diseño mecánico, se logra montar el dispositivo Kinect, el procesador (PC) y el enrutador para la comunicación, a la plataforma móvil terrestre la cual se puede ver en la figura 5.1. y 5.2

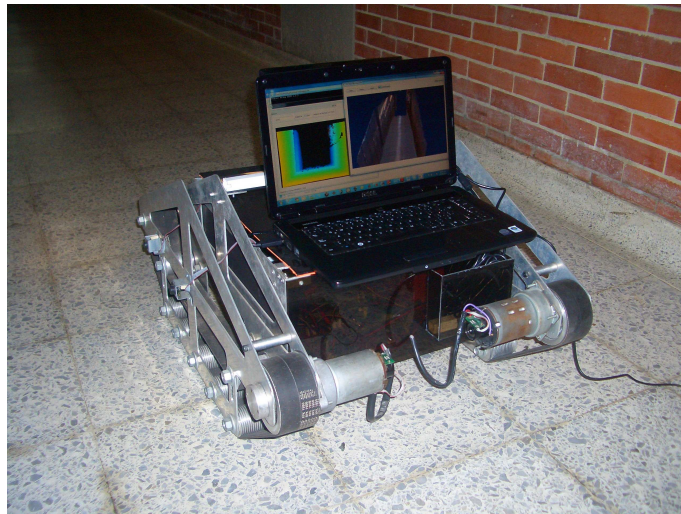


Figura 5.1: Montaje Final (Vista ISO)



Figura 5.2: Montaje final (Vista Frontal y Trasera)

Después de calibrar el dispositivo Kinect, y estar configurada con el ordenador Host, se procedió a activar las cámaras, como lo es la RGB, y la infrarroja, las cuales generan tres tipos de visualización:

La primera es la RGB la cual se muestra en la imagen 5.3, en la cual se usan tres colores para reproducir 16.7 millones de colores y formar las imágenes reales, las cuales permiten tener de forma inmediata lo que la cámara está mostrando en el momento de inicializar el programa.



Figura 5.3: RGB

Luego el sensor genera una imagen de mapas de profundidad, la cual escala los colores dependiendo de la distancia a la cual se encuentre los objetos de ella, generando la siguiente imagen 5.4, y que también se ve representada en la figura 5.5 donde se puede modificar la distancia de captura, tanto la máxima como la mínima permitiendo de esta forma dar rangos variables cuando se quiere capturar o reconstruir espacios de proporciones variables, lo cual hace que aumente el grado de calibración del dispositivo.

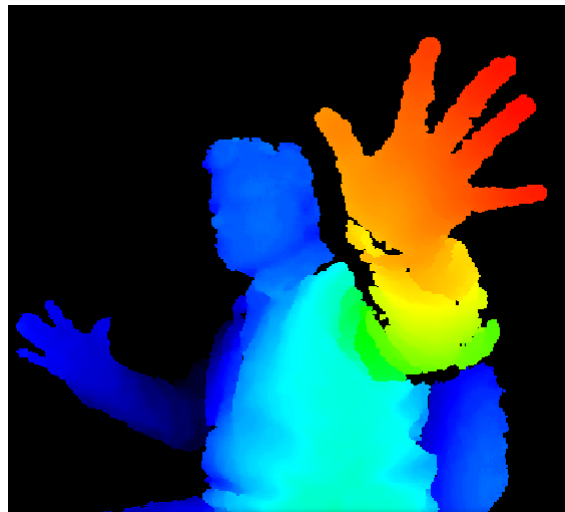


Figura 5.4: Mapa de profundidades



Figura 5.5: Maximos y minimos de cobertura de la imagen

Y por último la imagen de generación del mapa tridimensional del ambiente actual el cual se va generando a medida que la plataforma se va desplazando por el terreno, de esta forma el ambiente se recrea de forma virtual y se va generando una maya de puntos las cuales va almacenando en un archivo .PLY, el cual genera según la misma distancia recorrida, ya que entre más espacio o terreno recorrido mayor será el espacio en memoria que utilizara el dispositivo, por lo cual se requiere de una cantidad de memoria mínima para reconstruir espacios determinados.



Figura 5.6: Mapa reconstruido

Todo lo anterior lo cual se muestra en la pantalla principal de la plataforma móvil, es enrutado vía LAN, las cuales transmiten y reciben las imágenes generadas por el sensor y todo el procesamiento que se encuentra dentro del ordenador, y es visualizado en primera instancia por el usuario que se encuentra en la plataforma cliente, la cual le permite tener acceso inmediato a la reconstrucción, pero sin poder transformar dicha información; más adelante, después de tener un recorrido ya almacenado, el usuario es capaz de tomar dichos datos, en este caso la nube de puntos ya reconstruida, y tener la posibilidad de cambiar a gusto propio para hacer análisis posteriores, ya sean mediciones o rediseños de estructuras.

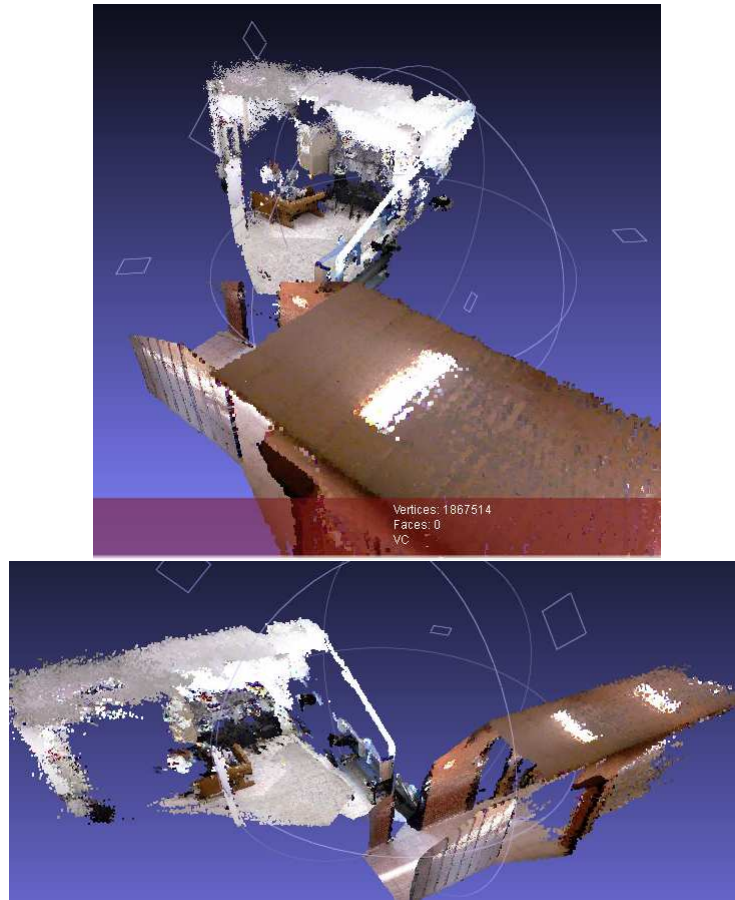


Figura 5.7: Ambiente reconstruido (Pasillo-Laboratorio)

Y por último se muestra la manipulación que se le puede dar a los datos ya capturados y guardados con anterioridad, los cuales permiten tomar mediciones y compararlas en dado caso con las reales, las cuales se observan en la siguiente imagen donde un pasillo es reconstruido y anexado a la plataforma MeshLab, donde se puede mejorar el aspecto visual de la reconstrucción.



Figura 5.8: Reconstrucción del pasillo

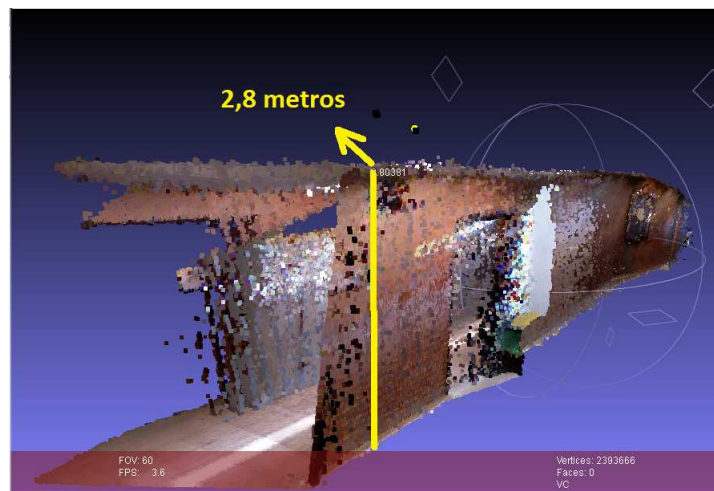


Figura 5.9: Medición virtual del ambiente reconstruido

Capítulo 6

CONCLUSIONES Y LINEAS FUTURAS

Basados en los objetivos propuestos, se logró abordar el problema planteado para el desarrollo y aplicabilidad de este proyecto, dando la solución, desarrollo e implementación del dispositivo con sus respectivas tareas para que al final se lograra crear la reconstrucción de un ambiente en 3D.

Como primer objetivo cumplido, se diseñó e implemento los soportes del sistema; el soporte del kinect se creó con el fin de tener la mejor panorámica sobre el ambiente, esto significo que dentro del rango visual del sensor no podía aparecer ninguna parte de la plataforma móvil o del propio sistema, ya que si esto sucedía se creaban superficies constantes e inexistentes dentro el entorno y por ende no se podía realizar una reconstrucción correcta del ambiente, teniendo los rangos visuales del sensor Kinect, fue posible establecer la ubicación exacta en la plataforma móvil terrestre; por otra parte se diseñó un soporte para el enrutador, este soporte se instaló en la parte exterior de la plataforma para aprovechar las ventajas que brinda la señal inalámbrica en cuanto a su velocidad de transporte de datos, por último se creó una base sobre la cubierta superior de la plataforma para colocar el computador encargado del procesamiento de los datos adquiridos por el sensor, esta plataforma se diseñó con el fin de brindar mayor soporte y estabilidad al procesador al momento de navegar por un ambiente evitando que el computador puede caer de la plataforma; todos es tos soportes se diseñaron con materiales ligeros como lo son el PVC y el Acrílico con el fin de no aumentar significativamente el peso de la plataforma y llegar a afectar su locomoción.

Para continuar con el desarrollo del proyecto y el cumplimiento de los objetivos planteados, se logró implementar dos tipos de arquitecturas de red, las cuales se crearon para permitirle al usuario acceder a la información de diferentes formas; las arquitecturas que se lograron implementar fueron de tipo LAN e

Internet, cada una de estas arquitecturas de red le permiten al usuario acceder a la plataforma móvil terrestre de forma remota, ya sea para sustraer los datos generados por el programa o para navegar por el entorno ya reconstruido, en ambos casos el cliente tiene acceso al escritorio remoto de la plataforma móvil y tiene la capacidad de ejecutar la aplicación para dar comienzo a la reconstrucción; la red LAN nos da la posibilidad de que el host como el cliente compartan entre ellos programas, información y recursos, la desventaja más notable con la red LAN es que para que ocurra el proceso de intercambiar la información los ordenadores deben estar cerca geográficamente; con la red de Internet no se limita a ningún espacio geográfico para establecer la comunicación entre los ordenadores.

Las ventajas de estos dos tipos de red, es que la navegación de la plataforma móvil terrestre y la reconstrucción del ambiente en 3D, pueden ser vistas de forma online gracias al acceso a los escritorios remotos.

Para el desarrollo de una aplicación futura se podría lograr que la plataforma móvil fuese autónoma, realizara recorridos ya planeados y que el usuario por medio de la red de Internet pudiera acceder a todo el proceso de la reconstrucción 3D en el tiempo que lo requiera.

Durante la investigación de métodos para implementar algoritmos de reconstrucción del entorno en 3D y dar cumplimiento al objetivo propuesto, se implemento un método para este desarrollo; la nube de puntos que fue el método utilizado para la reconstrucción 3D en este proyecto es la mejor propuesta para el almacenamiento de la información en cuanto al transporte y manipulación por parte del usuario debido a que la malla generada se vuelve tan simple que el usuario puede transformar y agrupar varias escenas rápidamente lo cual facilita de mejor forma la generación del mapa de profundidad y por ende la reconstrucción del entorno.

La principal ventaja de la nube de puntos es que el área de trabajo no se limita, esto quiere decir que se puede generar nuevas nubes de puntos a partir de otras ya creadas, lo cual permite tener un mejor detalle en la calidad final de la reconstrucción 3D, al igual por medio del levantamiento de la nube de puntos es posible tener directamente las medidas que se generan por el recorrido del dispositivo.

Bibliografía

[1]

[1] Gomis Martí; José María; Company Calleja, Pedro Pablo. Universidad Politécnica de Valencia. Reconstrucción de modelos sólidos 3d a partir de vistas ortográficas 2d utilizando técnicas de combinación de cuerpos elementales.

[2] Gomis J.M., Leiceaga X. Y Vejarano J. " Generación automática de axonometrías oblicuas a partir de sus vistas ortográficas" encuentro egraf 96, (1997).

[3] María Saiz Muñoz. Reconstrucción tridimensional mediante visión estéreo y técnicas de optimización madrid, junio de 2010.

[4] Ferran Naya, Manuel Contero, Estrategias de reconstrucción 2d en interfaces caligráficas. Universidad politécnica de valencia camino de vera, 46022 valencia.

[5] David Vela Ortega, Benito Gerónimo Marcos, Apolinar Ramírez Saldívar. Obtención del mapa de profundidad a través del flujo óptico.

[6] Beauchemin, S.S y J. L. Barrón. The computation of optical flow. Acm computing surveys, vol. 27, n° 3, september 1995.

[7] Demek, J., (ed.), Manual of detailed geomorphological mapping, international geographical union, Commission on geomorphological survey and mapping, Prague, 344 p. 1972.

[8] "Digital maging and communications in medicine (dicom)". Standar dicom specification, National electrical manufacturers association, 1999.

[9] S. Gibson, R. Hubbard, J. Cook and T. Howard. "Interactive reconstruction of virtual environments from video sequences". Computers & Graphics 27, pp. 293-301. 2003.

[10] K. Wong, P. Mendonça and R. Cipolla. "Reconstruction of surfaces of revolution from single uncalibrated views". Consultation date: August 2007.

Available in: ftp://svr-ftp.eng.cam.ac.uk/pub/reports/kykwong_bmvc02.pdf

[11] G. Wang, H. Tsui and Z. Hu. "Reconstruction of a structured scenes from two uncalibrated mages". *Pattern Recognition Letters*. Volume 26 N^o 2, pp. 207-220. January 2005.

[12] Y. Xue, M. Sun and A. Ma. "The reconstruction of three-dimensional complex geological objects using Delaunay triangulation". *Future Generation Computer Systems*. Special issue Geocomputation Volume 20 N^o 7, pp. 1227-1234. October 2004.

[13] Y. Yemez and F. Schmitt. "3D reconstruction of real objects with high resolution shape and texture". *Image and Vision computing*. Volume 22 N^o 13, pp. 1137-1153. November 2004.

[14] R. Bunschoten and B. Kröse. "3D scene reconstruction from cylindrical panoramic images". *Robotics and autonomous systems*. Volume 41 N^o 2, pp. 111-118. November 2002.

[15] M. Pollefeys. "3D recording for archaeological fieldwork". *IEEE Computer Graphics and applications*. Vol. 23 N^o 3, pp. 20-27. May-June 2003.

[16] F. Remondino. "3-D reconstruction of static human body shape from image sequence". *Computer Vision and Image Understanding*. Volume 93 N^o 1, pp. 65-85. January 2004.

[17] R. Koch. "3-D Surface Reconstruction from Stereoscopic Image Sequences". *ICCV*, pp. 109-114. 1995.

[18] Z. Zhang. "A Flexible New Technique for Camera Calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22 N^o 11, pp. 1330-1334. 2000.

[19] M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crapton y K. Ahlers. "Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System". *IEEE Trans. On Visualization and Computer Graphics*. Vol. 1 N^o 3, pp. 255-273. September 1995.

[20] D. Aracena, P. Campos and C. Tozzi. "Comparison of digital camera calibration techniques". *Revista de la Facultad de Ingeniería - Universidad de Tarapacá*. Vol. 13 N^o 1. 2004.

[21] C. Schmid, M. Mohr and C. Bauckhage. "Evaluation of Interest Point Detectors". *International Journal of Computer Vision*. Vol. 37 N^o 2, pp. 151-172. 2000.

- [22] C. Harris and M. Stephen. "A combined corner and edge detector". In Alvey vision conference, pp. 147-151. 1988.
- [23] P. Kovesi. "Image Correlation from Local Frequency Information". Proceedings of The Australian Pattern Recognition Society Conference: DICTA '95, pp. 336-341. Brisbane, Australia. December 1995. Available in: <http://www.cs.uwa.edu.au/pub/robvis/papers/pk/DICTA95.ps.gz>
- [24] Z. Zhang, R. Deriche, O. Faugeras and Q. Luong. "A Robust Technique for Matching Two Uncalibrated Images Through the recovery of the Unknown Epipolar Geometry". Artificial Intelligence Journal. Vol. 78, pp. 87 - 119. 1995. Available in: <http://citeseer.ist.psu.edu/article/zhang94robust.html>
- [25] R. Hartley. "In Defense of the 8-point Algorithm". Proceeding of the IEEE transactions on pattern analysis and machine intelligence. Vol. 19 N^o 6. 1995.
- [26] M. Pilu, "A Direct Method for Stereo Correspondence based on Singular Value Decomposition". Proc. CVPR, pp. 261-266. 1997.
- [27] R. Hartley, R. Gupta, T. Chang, "Stereo from Uncalibrated cameras". Consultation date: August 2007. Available in: <http://users.rsise.anu.edu.au/~hartley/Papers/cvpr92/final.pdf>
- [28] G. Scott and H. Longuet-Higgins. "An algorithm for associating the features de two patterns". Proc. Royal Society London. Volume B244, pages 21-26. 1991.
- [29] X. Zhuang and T. Huang. "Two-view motion analysis: a unified algorithm". Optics, Image Science, and Vision. Volume 3 N^o 9, pp. 1492-1500. September 1986.
- [30] R. Tsai and T. Huang. "Estimating three-dimensional motion parameters of a rigid planar match". IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP30, pp. 525-534. August 1982.
- [31] J. Weng and T. Huang. "Motion and structure from two perspective views: algorithms, error analysis, and error estimation". IEEE Transactions on Patterns Analysis and Machine Intelligence. Volume 11 N^o 5, pp. 451 - 476. May 1989.
- [32] B. Delaunay. "Sur la sphere vide". Bull. Acad. Sci. USSR VII 7, pp. 793 - 800. 1934.
- [33] E. Trucco and A. Verri. "Introductory to techniques for 3-D computer vision". Prentice Hall. 1998.

- [34] O. Faugeras. “Three-Dimensional Computer Vision: A Geometric Viewpoint”. The MIT Press. 1993.
- [35] N. Ayache. “Vision for mobile robots”. The MIT Cambridge Mass. 1991.
- [36] L. Shapiro and G. Stockman. “Computer Vision”. Prentice Hall. 2001.
- [37] Explicación de la Visión estereoscópica. Consultation date: August 2007. Available in: <http://www.adpsystems.net/Pdf/Visi%C3%B3n%20%20estereosc%C3%B3pica.pdf>
- [38] Características y especificaciones técnicas de kinect. Consultation date: September 2009. Available in: <http://esencialbit.com/noticias/caracteristicas-y-especificaciones-tecnicas-de-kinect/>
- [39] Juan Camilo Hernández Mejía, Diseño y Construcción de una Plataforma Móvil Teleoperada con Realimentación Visual para incursión en ambientes poco estructurados, Universidad Militar Nueva Granada, Facultad de Ingeniería, Programa de Ingeniería en Mecatrónica, Bogotá D.C. Noviembre 2009.
- [40] Rusinkiewicz, S. and Levoy, M. 2000. QSplat: a multiresolution point rendering system for large meshes. In Siggraph 2000. ACM , New York, NY, 343-352. DOI= <http://doi.acm.org/10.1145/344779.344940>
- [41] Meshing Point Clouds A short tutorial on how to build surfaces from point clouds.
- [42] 3D RiskMapping. “Teoría y Práctica del Escaneo Láser Terrestre”. Versión 5. Septiembre 2008.
- [43] Luis Sánchez Rodríguez “Obtención de mapas de profundidad densos mediante visión activa por movimiento controlado de una cámara. Aplicación a tareas de reconocimiento.” Abril 2003
- [44] S.T. Barnard y M.A. Fischler. Computational stereo. ACM Computer Surveys, 14(4):553-572, Diciembre 1982.
- [45] U.R. Dhond y J.K. Aggarwal. Structure from stereo: A review. IEEE Transactions on Systems, Man and Cybernetics, 19(6):1489-1510, Noviembre 1989.
- [46] L.G.Brown. A survey of image registration techniques. ACM Computer Surveys, 24(4):325-376, 1992.