



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

**“Estudio de Imágenes para la Reconstrucción a Bajo Costo
de Rostros usando Visión Estereoscópica y Luz
Estructurada”**

INFORME DE PROYECTO DE GRADUACIÓN

Previo a la obtención del título de:

INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

Presentado por:

Carlos Alberto Avilés Galarza

GUAYAQUIL – ECUADOR

2012

AGRADECIMIENTO

Agradezco a Dios,
por haberme guiado y acompañado a lo largo de mi carrera,

A mi padre,
Por ser una fuente de inspiración durante mi carrera universitaria.

A mi madre,
Por su constante e incondicional apoyo.

Al Sr. PhD. Sixto García Aguilar,
Por su gran paciencia y ayuda en el desarrollo del presente proyecto.

DEDICATORIA

Le dedico este trabajo a Dios por ser la luz que me ilumina siempre.

A mi familia que siempre me ha apoyado incondicionalmente en todas mis decisiones.

TRIBUNAL DE SUSTENTACIÓN

M. Sc. Sara Ríos

PRESIDENTE

Ph. D. Sixto García

DIRECTOR DE PROYECTO DE GRADUACIÓN

Ph. D. Boris Vintimilla

MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este informe de proyecto de graduación me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Carlos A. Avilés Galarza

RESUMEN

En el presente trabajo se ha realizado un estudio de imágenes con el fin de proveer una solución de escaneos 3D de rostros a bajo costo, y ofrecer dicha funcionalidad como herramienta de aplicación para futuros proyectos de investigación que estén relacionados al área de gráficos por computadora. Se trata de un sistema de reconstrucción de rostros a bajo costo utilizando las técnicas de luz estructurada y visión estereoscópica.

Como parte del proyecto, se implementó un prototipo que digitaliza rostros de manera rápida y precisa que son almacenados en archivos con extensión PLY para poder ser usados en herramientas como modeladores tridimensionales o aplicaciones relacionadas con el uso de modelos de rostros.

En el desarrollo de este proyecto se empezó con una revisión de la literatura acerca de cómo obtener modelos de rostros de una manera rápida. Inicialmente se optó por utilizar una técnica que involucraba el barrido de una proyección de puntos, la cual consistía en tomar varias fotografías de manera secuencial a medida que un número fijo de puntos verticales recorría un rostro de manera horizontal¹. Estas fotografías eran tomadas simultáneamente por cámaras que estaban separadas una distancia específica para poder generar una visión estereoscópica. Esta técnica

¹ Al hacer esto se obtenían varias fotografías con el avance de los patrones.

permite obtener información tridimensional de la superficie a escanearse debido a su perspectiva, que muestra una variación del patrón que es procesada por el ordenador para poder triangular las posiciones espaciales de todos los puntos. Luego de realizar varias pruebas se encontró que la técnica no era suficientemente controlable debido a que las cámaras que se estaban usando no poseían un control de balance de blanco el cual afectaba el diámetro de los puntos en las imágenes, y que a su vez proveía escaneos con una calidad muy baja.

Al investigar técnicas más avanzadas sobre escaneos en 3D, obviando la tecnología laser debido a sus altos costos, se optó por escoger la técnica de luz estructurada la cual es similar a la proyección de puntos pero omite la parte del barrido. En este caso se encontró que solo era necesario hacer una sola adquisición simultánea del rostro con las dos cámaras.

Esta última técnica consiste específicamente en la proyección de un patrón que está conformado por una serie de líneas blancas y negras que se van alternando. Al procesar las fotos tomadas, la computadora puede calcular muchos más puntos a partir de las líneas, las cuales se procesan al igual que la técnica de proyección de puntos. Una vez hechas las pruebas se obtuvieron resultados más precisos, pero se presentaron otros tipos de problemas. Estos problemas eran causados por las sombras generadas en el rostro a partir de la proyección y también por un problema de indexado de líneas al momento del procesamiento debido a que no es sencillo determinar

si una línea específica en la fotografía tomada desde la cámara 1 es la misma que la línea que se está recorriendo en la otra fotografía tomada desde la cámara 2.

La solución a estos problemas se encontró complementando la luz estructurada con los colores verdes, azul y rojo separados por una franja negra para aprovechar su alto contraste entre sí, lo cual a su vez le permite a las funciones de seguimiento de líneas recorrer la línea y no confundirse con líneas vecinas.

INDICE GENERAL

AGRADECIMIENTO.....	ii
DEDICATORIA.....	iii
RESUMEN.....	vi
INDICE GENERAL.....	ix
INDICE DE FIGURAS.....	xii
INDICE DE TABLAS.....	xvi
INTRODUCCION.....	xvii
Capítulo 1.....	1
1. Justificación y Análisis del Problema.....	1
1.1 Antecedentes.....	1
1.2 Descripción del Problema.....	2
1.3 Justificación.....	3
1.4 Propuesta de Solución.....	4
Capítulo 2.....	5
2. Revisión de la Literatura.....	5
2.1 Visión Estereoscópica.....	7
2.1.1 Sistema Visual Humano.....	7
2.1.2 Visión Estéreo.....	8
2.2 Luz Estructurada.....	11
2.2.1 Patrones.....	13

2.2.2	Indexado.....	16
2.2.3	Reconstrucción 3D.....	17
2.3	Librerías.....	20
2.3.1	OpenCV.....	20
2.3.2	OpenGL.....	21
2.3.3	MeshLab.....	22
	Capítulo 3.....	25
3.	Análisis y Diseño del Sistema.....	25
3.1	Análisis y Diseño de la Aplicación.....	25
3.1.1	Análisis Matemático.....	26
3.1.2	Arquitectura del Sistema.....	32
3.1.3	Componentes del Sistema.....	34
3.1.3.1	Módulo Adquisición de Datos.....	35
3.1.3.2	Módulo de Procesamiento de Datos.....	40
3.1.3.3	Módulo de Reconstrucción 3D.....	42
3.1.3.4	Módulo de Exportación.....	45
3.1.4	Formato del Archivo con extensión PLY para exportación.....	47
3.2	Limitaciones del Diseño.	49
3.3	Alcance del Diseño.....	50
3.4	Diseño de Plataforma para los equipos.....	51
3.5	Diseño de Pruebas.....	56
	Capítulo 4.....	58

4.	Implementación y Pruebas.....	58
4.1	Implementación.....	58
4.1.1	Manejo de Pixeles.....	59
4.2	Pruebas.....	62
4.2.1	Pruebas en Ambiente Controlado.....	62
4.2.2	Pruebas en Ambiente Real.....	68
4.3	Análisis de Resultados.....	77
4.3.1	Pruebas Reales y Simuladas.....	77
4.3.2	Uso de un Arreglo de Cámaras giradas con un ángulo	78
4.3.3	Uso de Colores en el Patrón.....	79
4.3.4	Cantidad de Líneas.....	80
4.3.5	Luz Estructurada Modulada.....	83
	CONCLUSIONES.....	84
	RECOMENDACIONES.....	86
	BIBLIOGRAFIA.....	88

INDICE DE FIGURAS

Figura 1. Sistema de Visión Humano.....	7
Figura 2. Cámaras en Visión Estéreo con poca área de convergencia.	9
Figura 3. Cámaras en Visión Estéreo con Rotación y con mayor área de convergencia.....	10
Figura 4. Reconocimiento de Luz Estructurada en la que la información del patrón es resaltada.	11
Figura 5. Toma de patrones con cámara en posición canónica.....	12
Figura 6. Patrón de luz estructurada RGB que consiste en líneas verticales de color Rojo, Verde y Azul separadas por líneas Negras.....	13
Figura 7. Proyección del Patrón RGB.....	14
Figura 8. Ejemplo de Proyección del Patrón RGB.	15
Figura 9. Detección de Líneas Proyectadas.	16
Figura 10. Indexado de Líneas.	17
Figura 11. Interfaz de MeshLab.	22
Figura 12. Visualización de Vértices de una Nube de Puntos en MeshLab..	23
Figura 13. Visualización de un Rostro en MeshLab.....	24
Figura 14. Ejemplo de Proyección de Barrido de una Serie de Puntos en Arreglo Vertical	25
Figura 15. Modelo de Triangulación Ideal.....	27

Figura 16. Problema que se presenta al separar las cámaras de su posición canónica.	29
Figura 17. Modelo de Triangulación tomando en cuenta la Rotación de las Cámaras.	30
Figura 18. Distancias de cada Línea.....	31
Figura 19. Esquema de la Arquitectura del Sistema.	32
Figura 20. Esquema E/S del Módulo de Adquisición de Datos.	35
Figura 21. Filtro que detecta niveles de colores para Clasificarlos.	36
Figura 22. Eliminación de Colores que no sean RGB y se infiere niveles cercanos a estos.....	37
Figura 23. Se resaltan líneas a nivel de pixel.	39
Figura 24. Función de Procesamiento Clasifica las líneas.....	41
Figura 25. Esquema E/S del Módulo de Reconstrucción 3D.	42
Figura26. La Reconstrucción de Polígonos une cada vértice con los vértices vecinos.....	44
Figura 27. Esquema del Módulo de Exportación.	45
Figura 28. Reconstrucción de polígonos usando la Nube de Puntos 3D.	46
Figura 29. Modelo 3D del Usuario.	47
Figura 30. Proyección del Prototipo.	50
Figura 31. Ilustración del funcionamiento del equipo.	51
Figura 32. Prototipo usando un Proyector LCD y 2 cámaras.....	53
Figura 33. Prototipo proyectando Luz Estructurada.....	54

Figura34. Pruebas Iniciales.....	63
Figura 35. Problema al usar el modelo ideal cuando las cámaras han sido rotadas.....	64
Figura 36. El problema ocasiona un estiramiento en la zona de la nariz y una contracción al nivel de los ojos.	65
Figura 37. Perspectiva de los efectos causados por el problema que aparece al usar el modelo ideal cuando las cámaras han sido rotadas y no se ha tomado en cuenta el nuevo modelo matemático.....	66
Figura 38. Pruebas usando el modelo matemático que considera la rotación de cámaras.	67
Figura 39. Proyección del Patrón RGB sobre un usuario en condiciones reales.	68
Figura 40. Problemas de mal indexado de líneas.	69
Figura 41. Reconstrucción en Baja Resolución.	70
Figura 42. Malla que muestra la cantidad de polígonos generados.....	70
Figura 43. Modelo 3D del Usuario resaltando las líneas detectadas.	71
Figura 44. Modelo 3D de Resolución Baja con Textura.....	71
Figura 45. Nube de Puntos 3D de un escaneo con mayor resolución.	72
Figura 46. Malla del modelo 3D mostrando cantidad de polígonos sobre el rostro.....	73
Figura 47. Nube de Puntos 3D.....	73

Figura 48. Modelo 3D de Usuario en mayor resolución con una proyección de 41 líneas.	74
Figura 49. Modelo 3D al que se le proyectó una cantidad de líneas superior a 45.....	75
Figura 50. Malla del usuario al que se le proyectó una cantidad de líneas superior a 45.....	75
Figura 51. Usuario usado en la explicación de los módulos.	76

INDICE DE TABLAS

Tabla 1. Dispositivos usados en el prototipo.....	55
Tabla 2. Datos obtenidos con proyección de 20 líneas.....	72
Tabla 3. Datos obtenidos con proyección de 41 líneas.....	74
Tabla 4. Datos obtenidos con proyección de 49 líneas.....	76
Tabla 5. Datos obtenidos con proyección de 22 líneas.....	77

INTRODUCCIÓN

En el campo relacionado con visión por computadora, el cual ha sido ampliamente investigado en la última década, existe un gran potencial para el desarrollo² de aplicaciones que puedan contribuir con el modo de interacción con las computadoras.

Este campo que se basa principalmente en el análisis de imágenes o video para la obtención de información relevante, está muy relacionado con áreas como gráficos por computadora, inteligencia artificial y realidad aumentada. Estos últimos requieren de herramientas de visión como por ejemplo un scanner 3D. Adquirir este tipo de dispositivos involucra costos muy altos que en algunos casos superan los \$100.000, los cuales no siempre están al alcance de la mayoría de los centros de investigación.

Un scanner 3D que es un dispositivo de alta precisión que permite recrear la geometría de las superficies que son capturadas ha ido evolucionando. Luego de varios años de investigación, existen varias soluciones más económicas y cuyos resultados ofrecen una calidad media-alta que en la mayoría de los casos es más que suficiente para que su aplicación permita obtener datos de mayor precisión muy importantes en otros campos.

² Este desarrollo involucra poseer cierta tecnología que permita investigar e innovar

El principal objetivo de este proyecto de grado es ofrecer una solución de bajo costo y cuyas digitalizaciones en 3D de rostros puedan proveer de datos suficientes a investigaciones relacionadas con los campos de inteligencia artificial y gráficos por computadora.

CAPITULO 1

1. JUSTIFICACION Y ANÁLISIS DEL PROBLEMA

En este capítulo se revisan los antecedentes, la descripción del problema con su justificación y la propuesta de solución que incluye el desarrollo de un prototipo.

1.1. ANTECEDENTES

El desarrollo de escáneres es un área que ha sido investigada durante varios años [8]. Existen varios tipos de escáner que están catalogados según la jerarquía en la imagen. Cada tipo de equipo tiene una técnica específica, y cada uno posee sus ventajas y desventajas. Todas estas técnicas pertenecen al área de visión por computadora. [2]

En definición, un escáner 3D es un dispositivo que analiza las características geométricas de un objeto para poder censar datos de su superficie y guardarlos como una nube de puntos en el espacio [6]. Esta nube de puntos permite su reconstrucción en una malla (mesh), la misma que consiste en un arreglo digital que conforma un modelo 3D.

Existen muchas aplicaciones para el manejo de estos modelos, los cuales son importantes para las industrias del entretenimiento, diseño industrial, ingeniería en reversa, prótesis, prototipos, control de calidad, seguridad, etc.

1.2. DESCRIPCION DEL PROBLEMA

Hoy en día en la ESPOL no existe un sistema que permita digitalizar objetos a bajo costo. Existen proyectos en el Centro de Tecnologías de Información que necesitan poder realizar escaneos 3D de objetos para investigaciones en áreas que incluyen inteligencia artificial, visión por computadora y realidad aumentada. Esta tecnología podría servir de base para la realización de proyectos como el desarrollo de un Sistema Inteligente de Recepción que incluya un avatar virtual. El objetivo de ese proyecto es manejar un personaje 3D el cual pueda interactuar con visitantes del centro y pueda relacionarse de manera adecuada.

La utilidad principal de esta herramienta es poder digitalizar uno o varios sujetos y poder catalogar la información 3D de varias de sus expresiones para poder ser introducido en un sistema inteligente.

Por lo indicado, se requiere de una herramienta de bajo costo que pueda capturar el rostro de una persona con una calidad media-alta. Con la tecnología de escáneres basados en luz estructurada y visión estereoscópica es posible construir un pequeño prototipo que esté conformado por 2 cámaras y 1 proyector. Estos equipos son muy accesibles para centros de

investigación y universidades y solo se requerirá del software que maneje la adquisición de datos, procesamiento y reconstrucción del resultado.

1.3. JUSTIFICACIÓN

Este proyecto tiene como objetivo proveer la funcionalidad de escanear rostros en una resolución relativamente buena a bajo costo. Se busca hallar las técnicas más eficientes y precisas en lo que se refiere a la digitalización de objetos. Esta funcionalidad permitirá obtener información geométrica sobre las caras escaneadas. A partir de esto se busca que la herramienta entregue un archivo con información que pueda ser usado en herramientas de visualización y manejo de modelos tridimensionales.

El dispositivo se puede implementar con 2 cámaras que tengan una resolución razonable cuyos costos no sean excesivos por lo que se opta por equipos disponibles en el mercado (off-the-shelf), además de un proyector LCD y una plataforma que mantenga fijos los dispositivos mencionadas según las especificaciones que maneja el escáner.

El archivo que generará el escáner y que contiene la nube de puntos 3D deberá tener un formato general que pueda ser leído en otras herramientas para su manejo posterior.

El sistema podrá ser usado en proyectos de investigación del área de visión por computadora y áreas relacionadas.

1.4. PROPUESTA DE SOLUCIÓN

Luego de efectuar una revisión sobre la técnica de triangulación de puntos en el espacio para la adquisición de datos con el uso de visión estereoscópica y cómo esta funciona en la técnica de proyección de patrones, se plantea lo siguiente:

- Diseño de las especificaciones que debe seguir el prototipo.
- Implementación del Sistema que manejará las imágenes
 - Adquisición de Imágenes
 - Uso de Filtros para resaltar información relevante
 - Procesamiento de la información para calcular nube de puntos espaciales.
 - Reconstrucción del modelo 3D de superficie escaneada.
- Construcción del prototipo.
- Realización de pruebas sobre varias personas para analizar la efectividad.

CAPITULO 2

2. REVISIÓN DE LA LITERATURA

En el presente capítulo se hace una revisión de la literatura que involucra los conceptos de visión estereoscópica y luz estructurada. Además se revisa el detalle de las librerías usadas en la implementación y las herramientas que ayudan en la visualización y manejo de información tridimensional.

Los escáneres 3D se dividen de acuerdo a los distintos tipos de tecnologías que utilizan para la obtención del modelo tridimensional [1] [2] [12]. Así tenemos:

- Escáneres de Contacto
- Escáneres de No-Contacto

Los escáneres de contacto como su nombre lo indica depende del contacto físico para censar las características de la superficie que se está digitalizando. Esta tecnología en la mayoría de casos involucra brazos robóticos que van siguiendo la superficie y guardando las distancias de cada sección del modelo. [2]

Los escáneres de No-Contacto no dependen de ningún contacto. Estos se dividen en dos categorías. [2]

- Activos
- Pasivos

Los escáneres Activos se basan en la emisión de algún tipo de radiación o luz, para poder detectar y usar esa medición con el fin de inferir sobre información 3D del ambiente o de un objeto. De estos hay varios tipos, como los de luz modulada, luz estructurada, tiempo de vuelo y triangulación, que son los ejemplos más conocidos.

La ventaja de este tipo de escáneres es que emiten radiación o luz para resaltar y detectar correctamente información importante para la digitalización de la superficie del objeto.

Los escáneres Pasivos no emiten ningún tipo de radiación y dependen de las características de la información obtenida en el proceso de adquisición como luz ambiental, o la visualización de una serie de imágenes de la silueta del modelo. Estos escáneres poseen algoritmos más avanzados y complejos de procesamiento para crear modelos con alta fidelidad.

2.1. VISION ESTERESCÓPICA

2.1.1. SISTEMA VISUAL HUMANO

El sistema visual humano (SVH) es lo que le permite a las personas visualizar su mundo día a día. Es un sistema aparentemente simple pero que maneja algunas tareas complejas. Consiste básicamente de los 2 ojos, el trayecto óptico y ciertas secciones del cerebro. Cada ojo capta la luz reflejada de una escena u objeto según su punto de vista. El tener 2 puntos de vista como se muestra en la Figura 1, es lo que le permite al cerebro censar distancias de objetos que están siendo observados, esto es visión estereoscópica. [6]

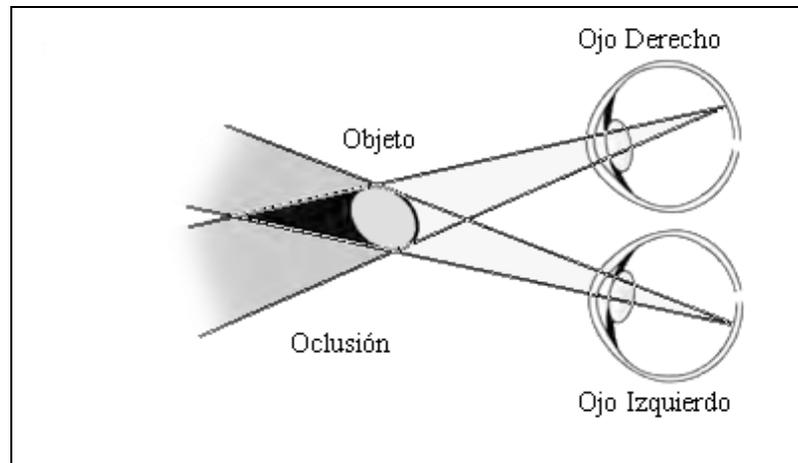


Figura 1. Sistema de Visión Humano. [6]

Uno de los objetivos de la visión por computadora es poder emular el sistema de visión humano, el mismo que es considerado un analizador y procesador de imágenes muy sofisticado, por lo que su entendimiento ayudará en el

diseño de sistemas de visión por computadora más precisos, efectivos y eficientes.

2.1.2. VISIÓN ESTEREO

La visión estereoscópica es posible gracias a las imágenes obtenidas desde 2 puntos de vista separados una cierta distancia, cuya combinación permite obtener información tridimensional de lo que está siendo observado. [8]

Al emular el comportamiento del SVH, se tiene el sensor CCD(Charge-coupled device³) de 2 cámaras que están separadas cierta distancia al igual que los ojos. Ambas cámaras están conectadas por medio de cables (nervios ópticos) al ordenador que es el encargado de hacer el procesamiento necesario para analizar la información adquirida al igual que el cerebro humano.

En la visión estéreo se puede tener información útil solo en el área que converge el campo de visión de ambas cámaras tal como se muestra en la región sombreada de la Figura 2. [4]

³ Dispositivo para el movimiento de carga eléctrica. Es un tecnología para imágenes digitales.

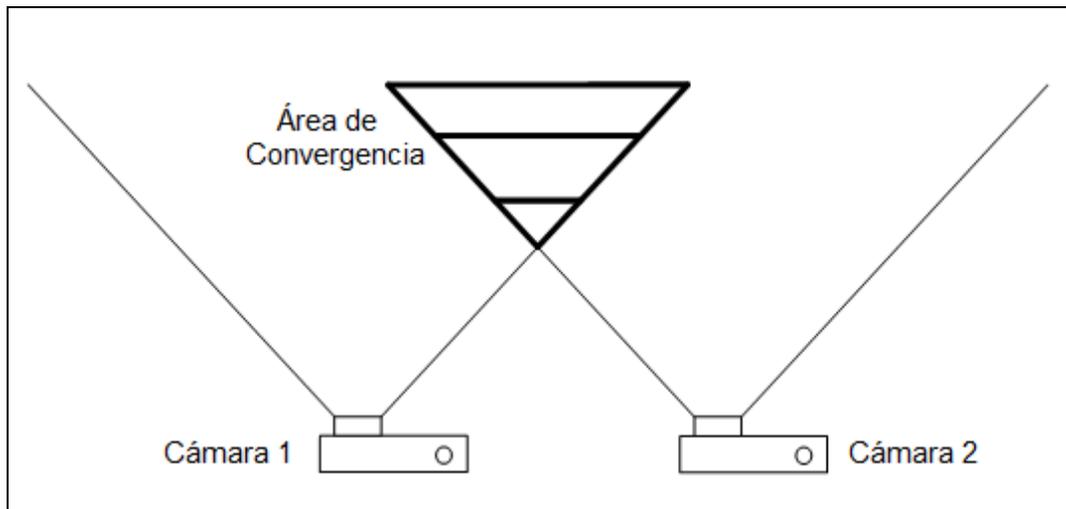


Figura 2. Cámaras en Visión Estéreo con poca área de convergencia.

El procesamiento que se realiza con imágenes obtenidas desde dos puntos de vista distintos es logrado por medio del método conocido como triangulación, que consiste en el cálculo de la distancia de un punto según su variación en ambas imágenes. Para poder hacer uso de este método de manera correcta, los puntos que estén siendo triangulados deben ser visibles por ambas cámaras. En el caso de la figura 2, la información que se encuentra fuera del área de convergencia de las dos cámaras es insuficiente para usar el método de triangulación. Los casos en que un punto no pueda ser visualizado por una cámara puede ser debido a:

- El punto está fuera de la región en que convergen ambas cámaras.
- El punto está siendo obstruido por otro punto; esto es definido como oclusión

- El material en esa zona absorbió la luz e impide que la cámara pueda visualizarlo.
- En esa zona hay sombra ocasionada por la geometría del objeto y la fuente de luz.

En la Figura 2 se puede apreciar que ambas cámaras están de forma perfectamente paralela. Este caso es conocido como un sistema con un arreglo estéreo en la posición canónica [6].

Para poder usar el método de triangulación se dedujo una simple ecuación matemática, pero hubo la necesidad de rotar las cámaras para obtener más información 3D necesaria para una reconstrucción adecuada. Al rotar las cámaras como se observa en la Figura 3 la fórmula debió ser ajustada.

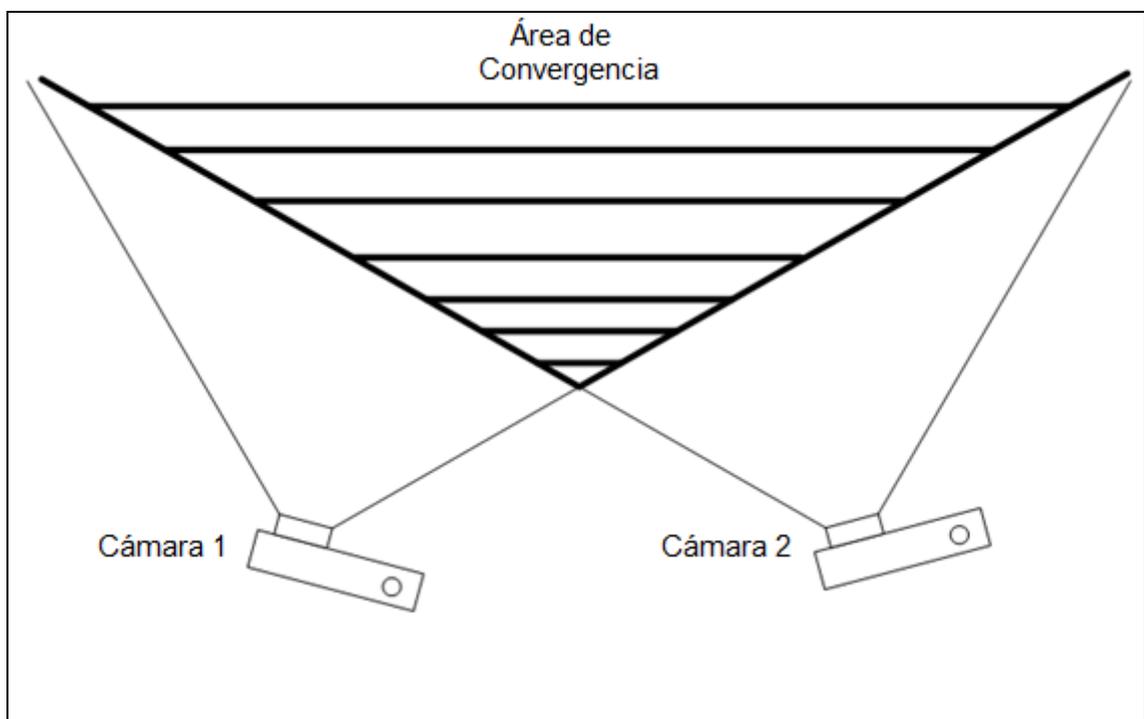


Figura 3. Cámaras en Visión Estéreo con Rotación y con mayor área de convergencia.

2.2. LUZ ESTRUCTURADA

La luz estructurada consiste en la proyección de cualquier tipo de patrón sobre una superficie para poder estimar la información geométrica de dicha superficie [8] [9]. En este caso se proyecta un patrón conocido, usualmente de líneas horizontales o verticales (ver Figura 4), de tal modo que una imagen adquirida desde una perspectiva diferente a una toma frontal facilita visualizar deformaciones en el patrón los cuales al ser analizados por el computador, permite calcular la profundidad de puntos detectados sobre la superficie. [5] [11] [13]

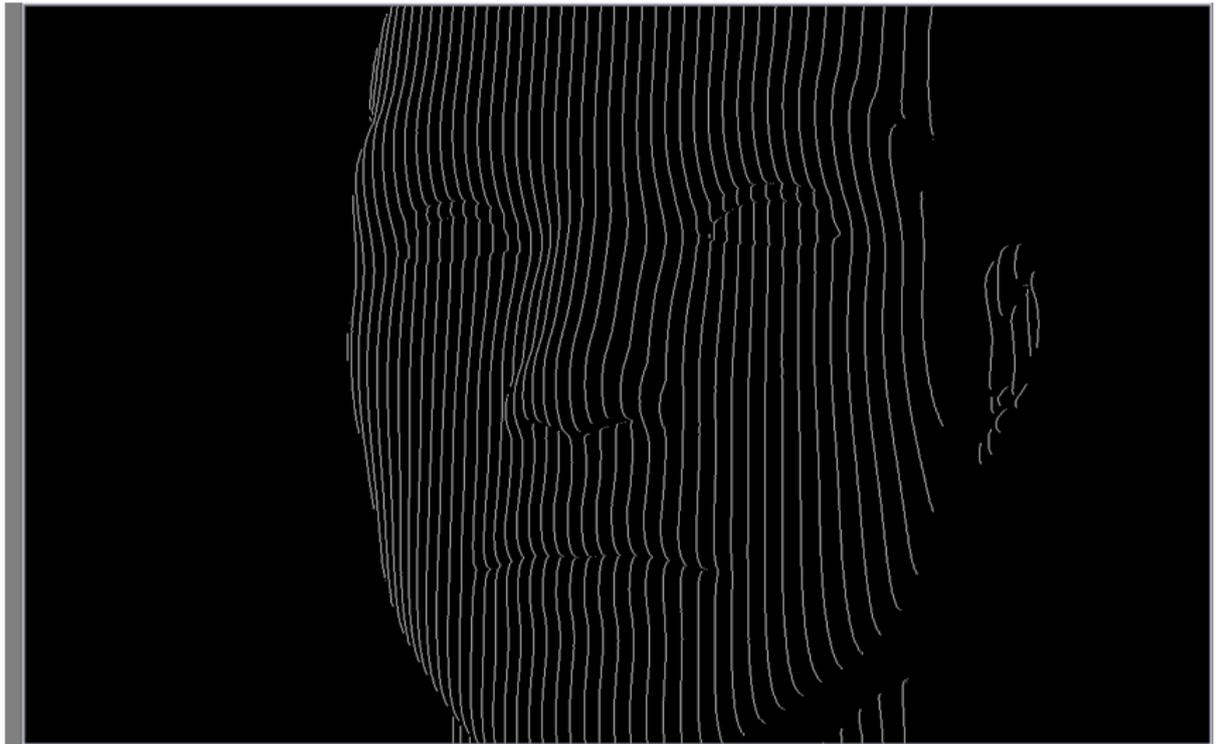


Figura 4. Reconocimiento de Luz Estructurada en la que la información del patrón es resaltada.

Conociendo esto, se puede definir un escáner 3D que está basado en luz estructurada como un dispositivo que permite proyectar el patrón sobre un objeto. Además del sistema de adquisición de las imágenes tomadas en perspectiva no puede faltar el computador que pueda procesar las deformaciones en los patrones para poder inferir sobre la forma de la superficie del objeto que está siendo escaneado.

En la Figura 5 se muestra una simulación con cámaras en posición canónica. Se puede observar cómo las líneas se deforman sobre la superficie del rostro y las regiones de sombra. Es notorio que las deformaciones de las líneas no son muy significativas y no hay una cantidad muy grande de información tridimensional para esta disposición de las cámaras.

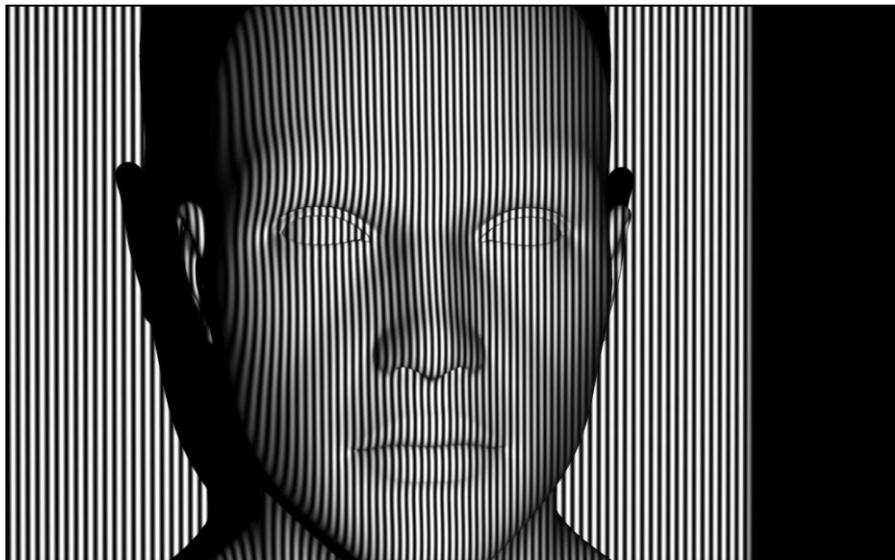


Figura 5. Toma de patrones con cámara en posición canónica.

El no disponer de mucha información tridimensional ocasiona problemas en la etapa de reconstrucción, como por ejemplo, una muy baja resolución en la superficie de los modelos generados. Esta baja resolución ocasiona que las líneas detectadas no sean continuas y afecta de manera negativa el resultado final.

2.2.1. PATRONES

Los patrones en general son líneas verticales alternadas entre blanco y negro que es el caso en la Figura 5. Para disminuir el riesgo que se presente el problema de correspondencia de líneas se agregan los colores Rojo, Verde y Azul que son alternados con líneas negras como se puede apreciar en la Figura 6.

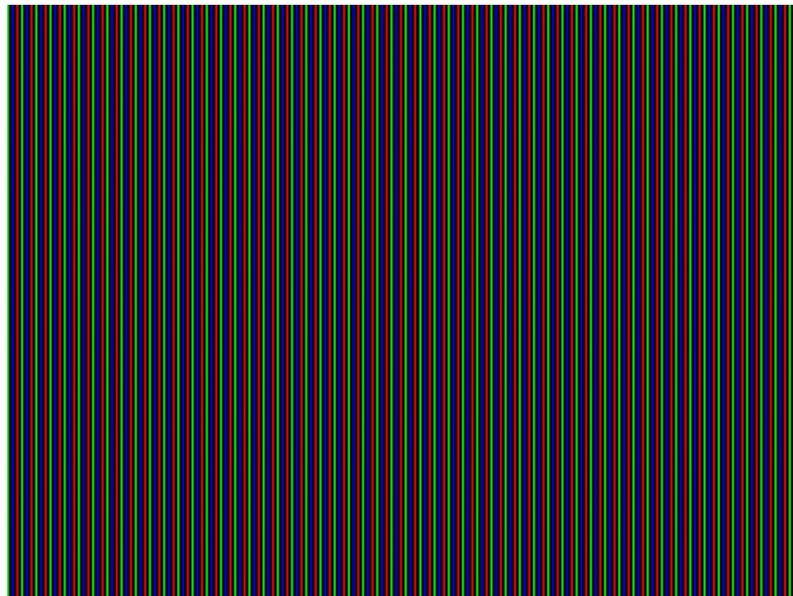


Figura 6. Patrón de luz estructurada RGB que consiste en líneas verticales de color Rojo, Verde y Azul separadas por líneas Negras.

El patrón RGB⁴ que se usa permite que haya una menor confusión de cada línea con sus líneas vecinas (más próximas).

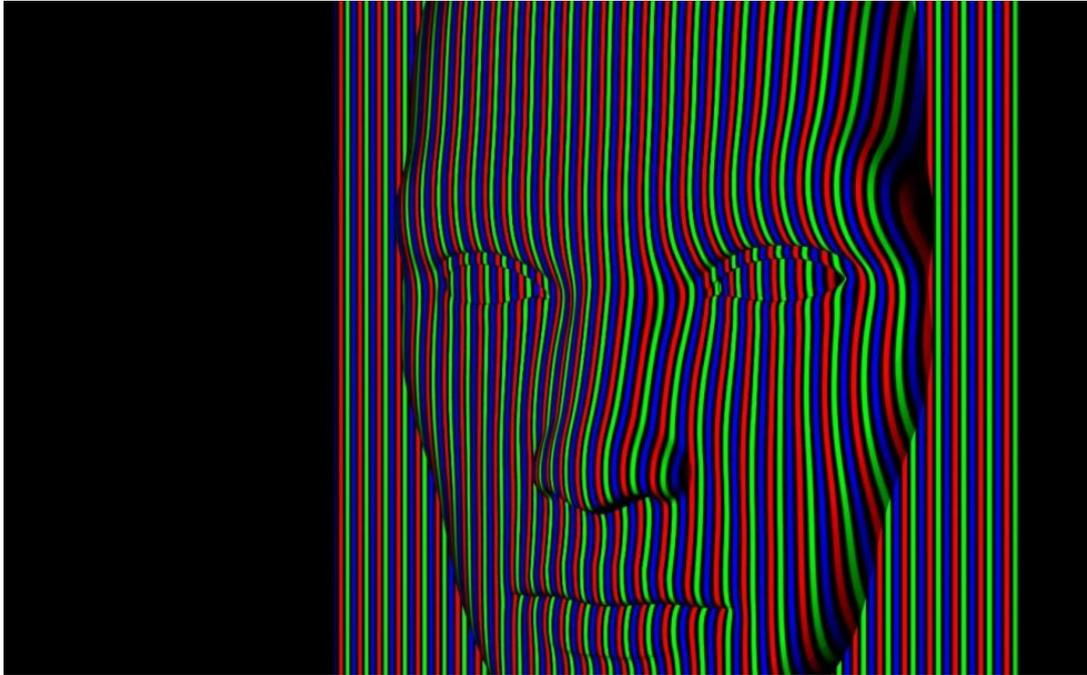


Figura 7. Proyección del Patrón RGB.

En la Figura 7 se puede observar la proyección del patrón sobre un rostro simulado con una mayor resolución que en el caso de la figura 6.

Una de las razones principales para escoger estos colores específicamente fue porque son colores con alto contraste entre sí y facilitan su clasificación por el ordenador cuando las imágenes son leídas. Se debe mencionar adicionalmente que el tono de los colores no es afectado mayormente por el color de la piel de los usuarios como se observa en la Figura 8. Al hacer

⁴ RGB por Red-Green-Blue que son los colores rojo, verde y azul respectivamente.

pruebas con colores menos contrastados estos variaban según el color de la superficie escaneada y dificultaba el proceso de censar de qué color se trataba, y a su vez ocurría un gran porcentaje de confusión en la ejecución de los algoritmos.



Figura 8. Ejemplo de Proyección del Patrón RGB.

2.2.2. INDEXADO

El indexado de líneas es un proceso que se realiza con el fin de comparar adecuadamente el mismo punto visto desde dos ángulos diferentes (dos tomas). Para lograr esto se debe identificar cada línea individualmente que está siendo leída en las 2 imágenes, asignándole un número. A partir de esta numeración se puede comparar correctamente las variaciones de los patrones. [4]

Para hacer esto en forma efectiva es esencial resaltar las líneas y eliminar información ambigua o ruido que pueda haber ingresado en el proceso de adquisición de datos. En la Figura 9 puede observarse que las líneas son claramente visibles luego de haber pasado por filtros que son explicados más adelante.

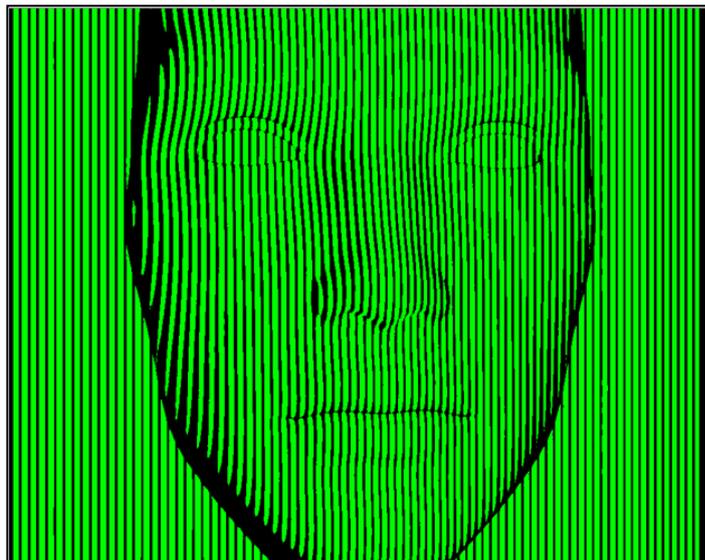


Figura 9. Detección de Líneas Proyectadas.

Las líneas resaltadas se las procesa para dejarlas con un grosor de 1 píxel tal como se observa en la Figura 10. El proceso de indexado se lo muestra con distintos colores para resaltar la diferenciación entre las líneas.

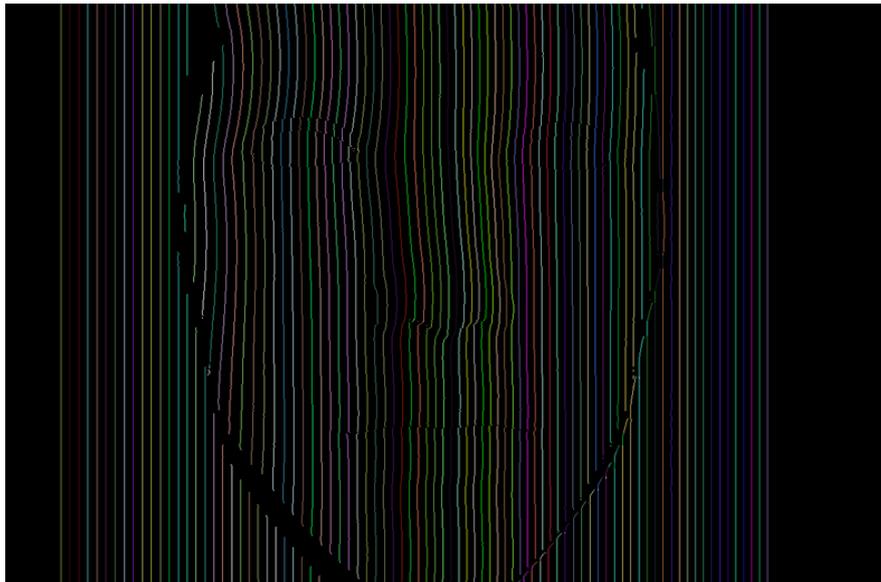


Figura 10. Indexado de Líneas.

2.2.3. RECONSTRUCCIÓN 3D

La reconstrucción 3D es el proceso mediante el cual se generan modelos tridimensionales basados en la información geométrica de la escena o de un objeto que fue capturado en el mundo real.

La información geométrica se establece sobre una nube de puntos en el espacio, la misma que representa una serie de vértices en un sistema de coordenadas tridimensional cuya distribución es representativa de la geometría del objeto, escena o superficie [17].

Los vértices se obtienen usando un dispositivo de escaneo 3D. Según el método empleado, el ordenador procesará la información adquirida y generará la nube de puntos o vértices necesarios para la reconstrucción. Estos vértices permiten visualizar las características del objeto o escena usando polígonos formados a partir de la unión de dichos vértices. Los tipos de polígonos generalmente empleados son los triángulos y los paralelogramos.

El modelo tridimensional formado a partir de los vértices y polígonos se conoce como malla, el mismo que puede ser usado en alguna otra herramienta tal como modeladores 3D.

El estudio de generación de mallas es parte del campo de geometría computacional que tiene como objetivo el optimizar la distribución de los polígonos en un modelo tridimensional. [18]

Existe una extensa literatura sobre la generación de mallas dado que se trata de un campo de intensa investigación. En [19] se presenta una excelente introducción al concepto de generación de mallas, en donde se revisan los temas de tipos de dominios geométricos, tipos de mallas, y los métodos numéricos. Existen algunas referencias sobre los métodos numéricos como por ejemplo [20] que trata de los métodos de las diferencias finitas, sobre la generación de malla estructurada [21] y la no estructurada [22], en la que se discute ciertas ventajas que tiene esta última sobre la generación de malla estructurada, siendo una de ellas su simplicidad. A su vez, existen sitios web interesantes en referencia a este tema tales como [23] y [24] que tienen información siempre actualizada.

2.3. LIBRERÍAS

A continuación se dan detalles sobre las librerías más importantes en el desarrollo de la aplicación.

2.3.1. OPENCV

OpenCV es una librería que es muy usada en el campo de visión por computadora. Esta librería posee funciones muy importantes para la adquisición y procesamiento de imágenes. Posee funcionalidades que le permiten trabajar en tiempo real. Originalmente escrita en C, OpenCV es portable, lo que le permite ser usada en otras plataformas como Python, Ruby, Java, C#. [16]

Las aplicaciones hechas con OpenCV involucran:

- Toolkits de características en 2D y 3D
- Sistemas de Reconocimiento Facial
- Reconocimiento de Gestos
- Interacción Hombre Máquina
- Robótica
- Detección de movimiento
- Identificación de objetos
- Segmentación e Identificación

- Visión Estereoscópica (Percepción de Profundidad usando 2 cámaras)
- Estructura del movimiento

A su vez posee una librería que incluye aprendizaje de maquina (machine learning).

Esta librería fue muy útil en el manejo de las imágenes, en la clasificación de colores y en el filtrado de la adquisición de datos.

2.3.2. OPENGL

OpenGL es una librería que maneja gráficos en 3D, es portable y muy rápida. Es estrictamente definida como una interfaz de software para hardware de gráficos [7]. Es desarrollado por Silicon Graphics Inc que es reconocida como una empresa líder global en gráficos por computadora y animación.

Esta librería ofrece herramientas que permite crear, modificar y visualizar objetos 3D en distintas plataformas. Funciona como un lenguaje basado en procedimientos en vez de ser descriptivo. Posee un API que incluye aproximadamente 120 comandos y funciones que sirven para dibujar gráficos primitivos como puntos, líneas y polígonos en 3 dimensiones. Adicionalmente, OpenGL tiene funcionalidades como efectos de iluminación, shading, mapeo de texturas, animación y otros efectos especiales.

El uso de esta librería es esencial cuando es necesario digitalizar rostros. Los modelos tridimensionales están conformados por polígonos que debe ser definidos usando las primitivas provistas por este software. Al crear este arreglo de polígonos se requiere también poder visualizar los resultados para poder hacer análisis y cambios a las implementaciones realizadas.

2.3.3. MESH LAB

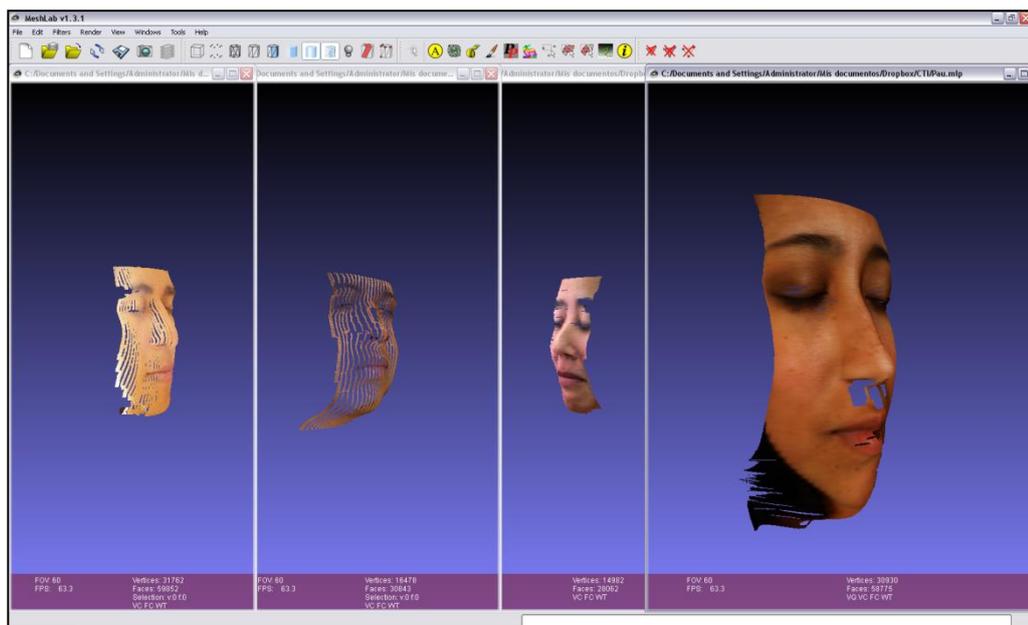


Figura 11. Interfaz de MeshLab.

MeshLab es un software avanzado que permite la visualización, análisis, y manejo de mallas (meshes). Es muy usado para el manejo de una nube de puntos. MeshLab es un software libre con el objetivo de proveer herramientas

para el manejo de nubes de puntos después de escaneos 3D. Esta última propiedad lo hace idóneo para el manejo de escaneos de rostros. Este software está disponible en múltiples plataformas. A partir de éste se puede exportar modelos a otros programas como software de gráficos por computador para modelado y animación. Esto es muy útil para poder visualizar mucho mejor los resultados y poder tener una realización (renders) de los modelos. Los modeladores proveen la funcionalidad de texturizar los modelos con fotos tomadas a los usuarios.

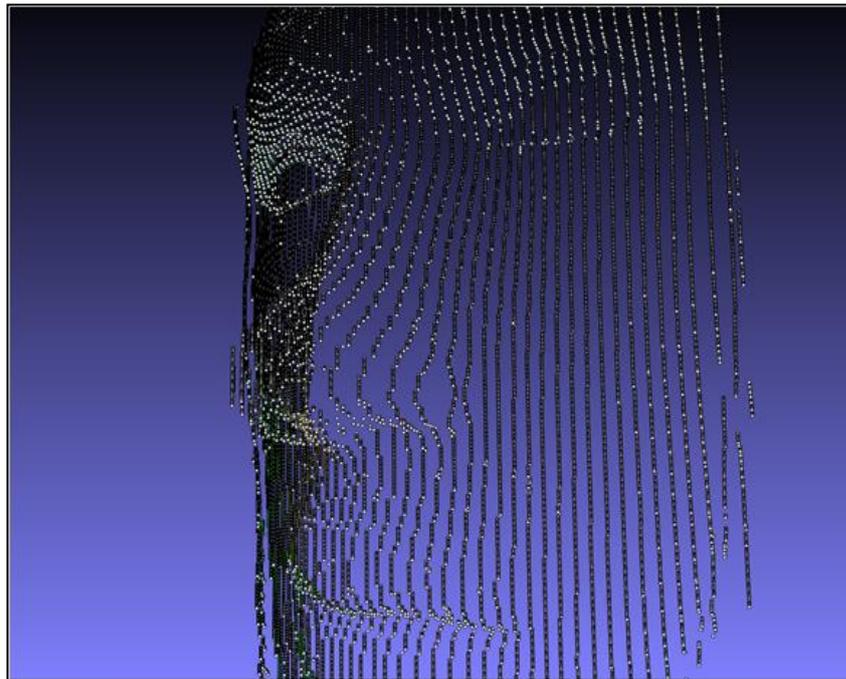


Figura 12. Visualización de Vértices de una Nube de Puntos en MeshLab.

En las Figuras 11 y 12, se pueden visualizar escaneos realizados por la herramienta desarrollada y el manejo de la nube de puntos 3D respectivamente. Para la generación de la imagen de la Figura 13 se usaron herramientas que vienen incluidas en este software para visualizar de mejor manera información relevante en modelos reconstruidos.

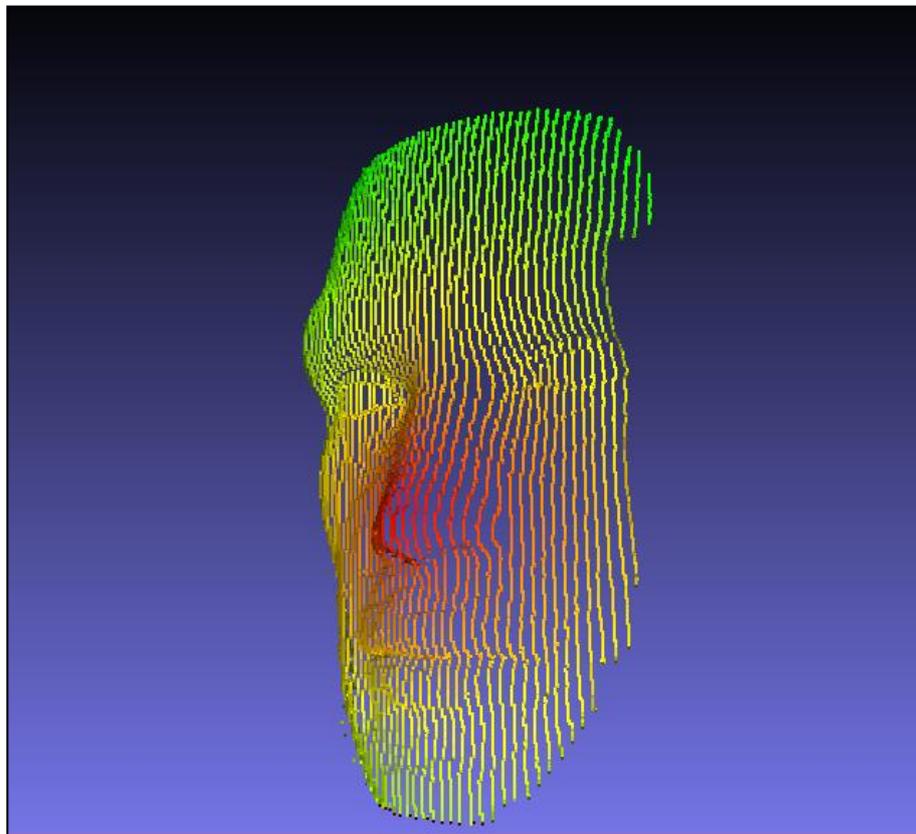


Figura 13. Visualización de un Rostro en MeshLab.

CAPITULO 3

3. ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

En el inicio del proyecto se comenzó implementando un barrido de puntos utilizando un modelo matemático ideal de visión estereoscópica que involucra el posicionamiento canónico de las cámaras.

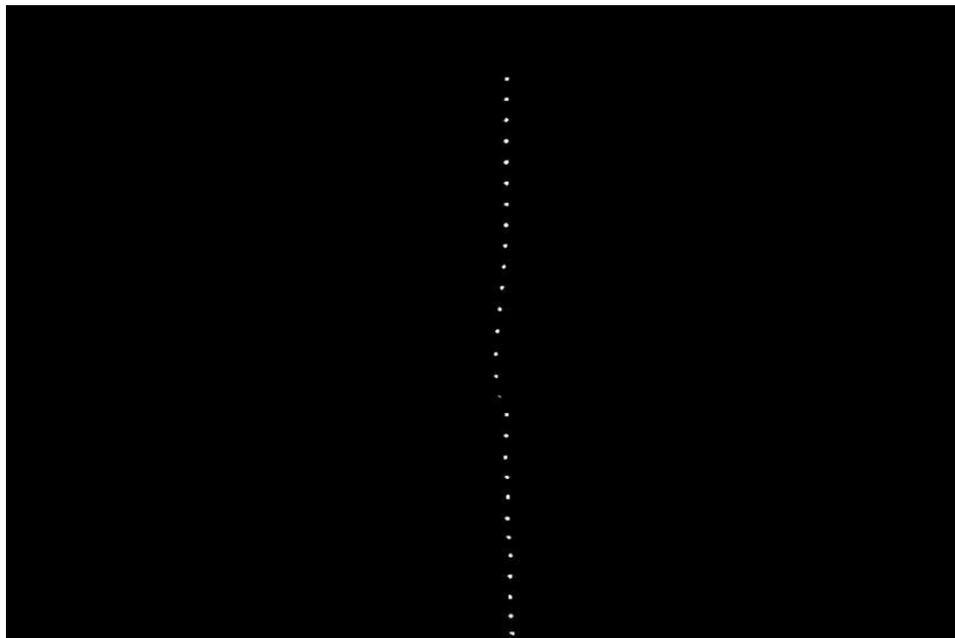


Figura 14. Ejemplo de Proyección de Barrido de una Serie de Puntos en Arreglo Vertical

En la Figura 14 se observa un arreglo de puntos sobre un rostro ordenados de manera vertical correspondiente a una toma desde la cámara derecha.

Ese arreglo era desplazado sobre el rostro de izquierda a derecha, y en cada uno de los desplazamientos era fotografiado y almacenado en el ordenador para su posterior procesamiento. Para el proceso de reconstrucción se usó el mismo principio de triangulación considerando la posición canónica de las cámaras.

El método indicado no funciona adecuadamente en condiciones reales debido a que en varias fotos tomadas se presentan problemas debido al balance de blanco en las cámaras que afectan el diámetro de los puntos obtenidos y proveen mediciones incorrectas de las variaciones del patrón. Por tanto, se optó por un patrón de líneas verticales ya que de esta forma el proceso de adquisición consiste en una sola toma de ambas cámaras. Esto soluciona efectivamente los problemas ocasionados por el balance de blanco cuando se toman varias fotos.

3.1.1. ANÁLISIS MATEMÁTICO

En la implementación inicial se consideró únicamente el caso en que las cámaras tenían un arreglo de manera canónica. De esta manera se obtuvo una fórmula para calcular la profundidad de cada punto según su variación de desplazamiento en el eje X. La deducción matemática en base al modelo de la figura 15 se detalla a continuación.

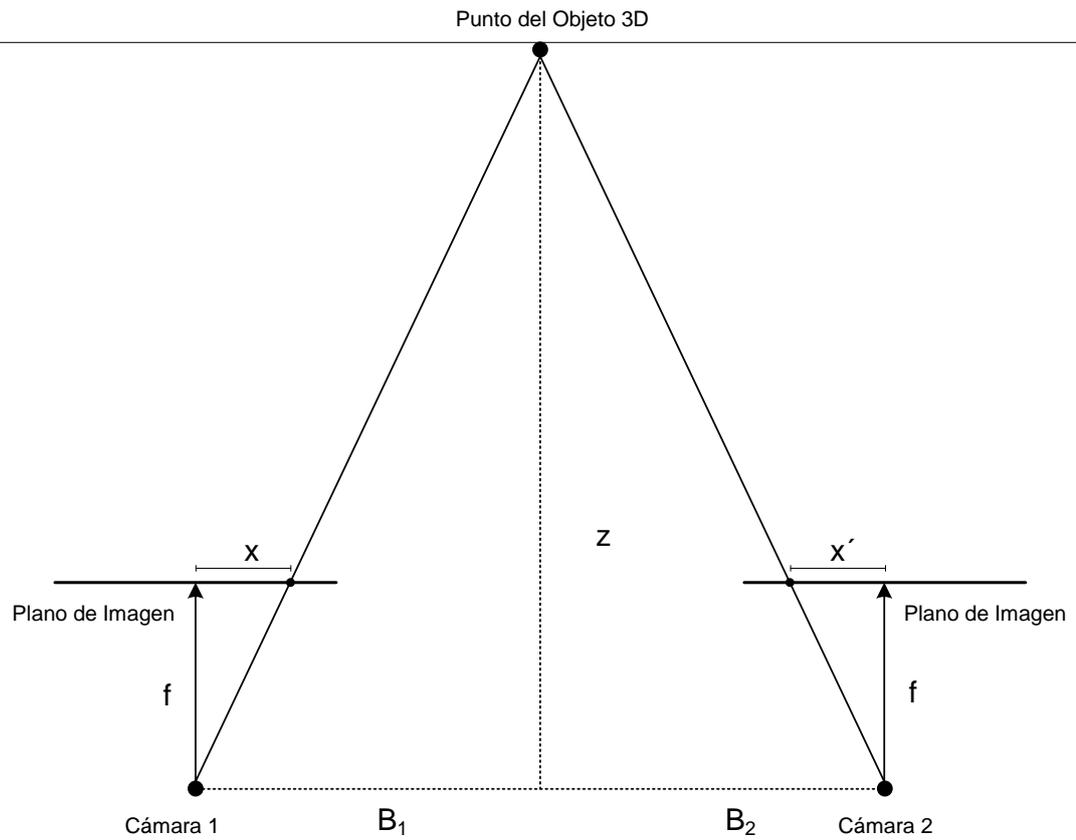


Figura 15. Modelo de Triangulación Ideal.

$$\frac{x}{f} = \frac{B_1}{z} \quad (1)$$

$$\frac{x'}{f} = \frac{B_2}{z} \quad (2)$$

$$B = B_1 + B_2 \quad (3)$$

$$z = \frac{fB_2}{x'} \quad (4)$$

$$z = \frac{fB_1}{x} \quad (5)$$

$$z = \frac{fB}{x+x'} \quad (6)$$

En la ecuación 1 se hace una relación entre el triángulo pequeño y el grande debido a que son congruentes. En la ecuación 2 se hace lo mismo pero con los triángulos del lado derecho.

El concepto matemático principal aplicable en este caso es el de congruencia de triángulos que da como resultado una ecuación sencilla para el cálculo de profundidad (ecuación 6), pero que solo rige cuando las cámaras se encuentran en la posición canónica.

Este método resultó muy efectivo para calcular la profundidad de los puntos leídos en ambas imágenes, pero no había suficiente información tridimensional debido a la proximidad de las dos cámaras cuya solución parcial se encontró separando más las cámaras. Inicialmente, se separaron las cámaras 20 cm una de otra. A esta distancia el rostro se encontraba dentro del área de convergencia. Se encontró que aumentando la separación a distancias mayores de 35cm incrementaba la información 3D a los niveles que eran adecuados para la reconstrucción.

La mayor separación de las cámaras presentó el inconveniente que el rostro salía de la región de convergencia como se observa en la Figura 16.



Figura 16. Problema que se presenta al separar las cámaras de su posición canónica.

Como solución final se decidió separar más las cámaras (40 cm entre sí) y rotarlas (14° en el eje de la distancia entre el dispositivo y el rostro) por lo que apareció la necesidad de ajustar la ecuación al nuevo escenario que considere la rotación de las cámaras (ángulos de rotación de los equipos de adquisición).

En la Figura 17 se aprecia el nuevo modelo que considera los ángulos. Básicamente, se usó la misma ecuación del modelo canónico en la figura 15, pero ahora las distancias X y X' deben ser encontradas en términos de los desplazamiento de los puntos considerando el ángulo.

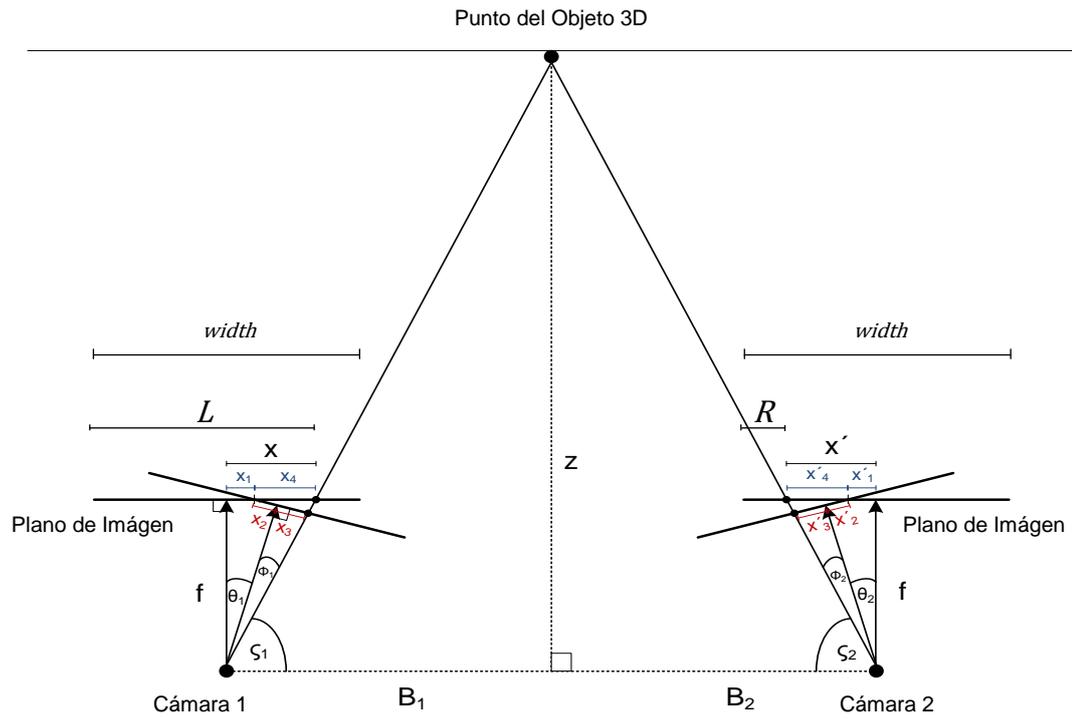


Figura 17. Modelo de Triangulación tomando en cuenta la Rotación de las Cámaras.

$$x_1 = x_2$$

$$x_2 = f \tan \frac{\theta_1}{2}$$

$$\phi_1 = \tan^{-1} \frac{x_3}{f}$$

$$\rho_1 = 90 - \theta_1 - \phi_1$$

$$x_3 = L - \frac{\text{width}}{2}$$

$$x_1' = x_2'$$

$$x_2' = f \tan \frac{\theta_2}{2}$$

$$\phi_2 = \tan^{-1} \frac{x_3'}{f}$$

$$\rho_2 = 90 - \theta_2 - \phi_2$$

$$x_3' = \frac{\text{width}}{2} - R$$

$$x_4 = \frac{(x_2 + x_3) \operatorname{sen}(180 - \theta_1 - \rho_1)}{\operatorname{sen}(\rho_1)} \quad x_4' = \frac{(x_2' + x_3') \operatorname{sen}(180 - \theta_2 - \rho_2)}{\operatorname{sen}(\rho_2)}$$

$$x = x_1 + x_4$$

$$x' = x_1' + x_4'$$

Las imágenes obtenidas al momento de adquisición son representadas por el plano de imagen. Los valores L y R son los valores que se obtienen en el plano de imagen de la posición del punto según el punto de vista, mientras que el 'width' es el ancho de la imagen.

En la Figura 18 se tiene una vista de planta (superior) de cómo van variando los ángulos según el arreglo de puntos en una línea que está siendo calculada.

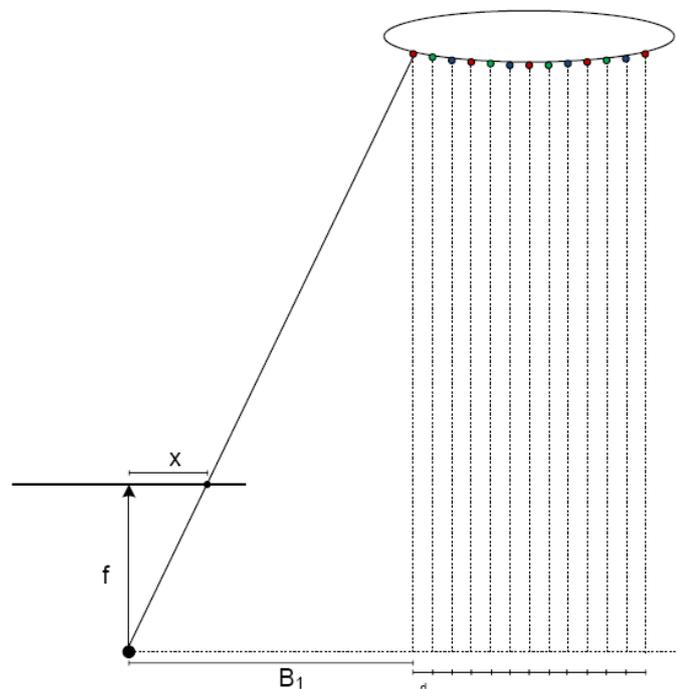


Figura 18. Distancias de cada Línea.

3.1.2. ARQUITECTURA DEL SISTEMA

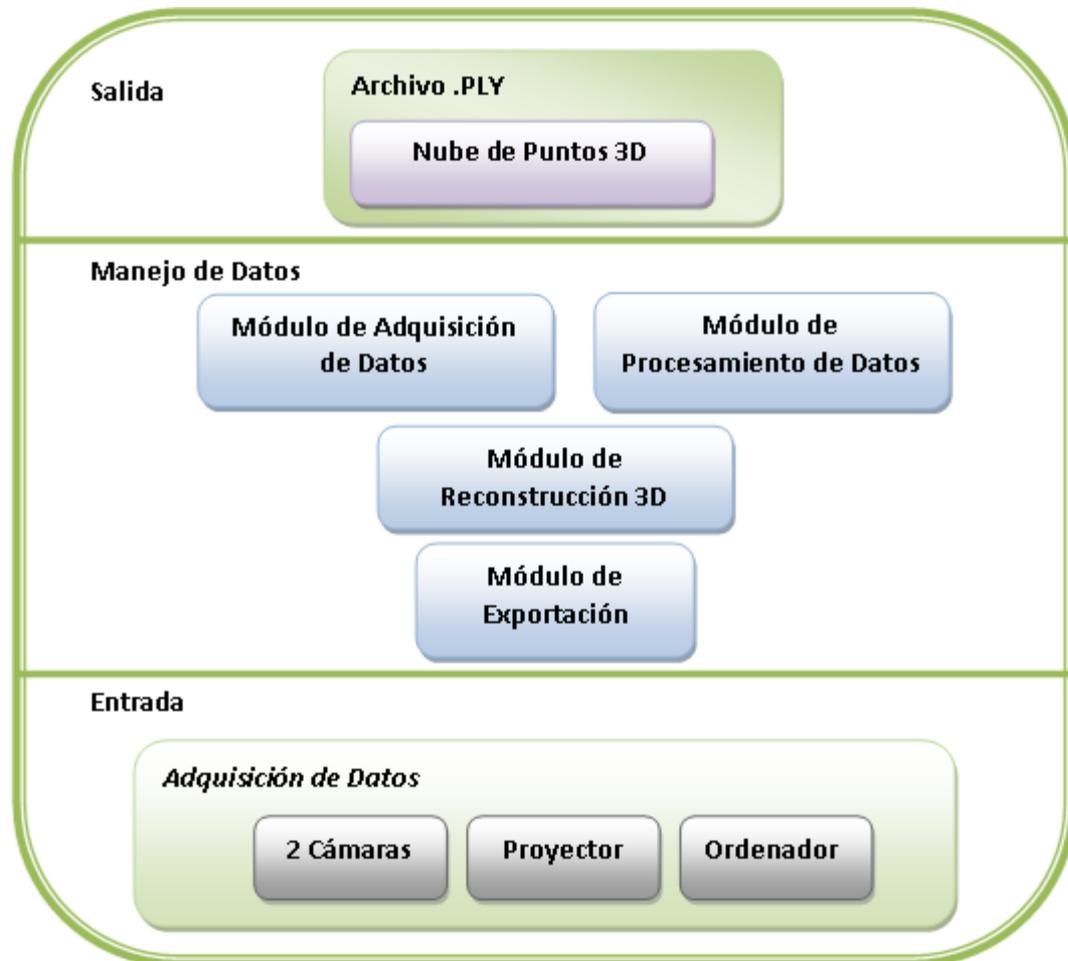


Figura 19. Esquema de la Arquitectura del Sistema.

En la Figura 19 se puede observar la conformación de las distintas capas de la arquitectura del sistema. En la capa de entrada se tienen los equipos necesarios para la adquisición de datos, que involucran los 2 sensores

ópticos, el proyector LCD y el ordenador que va a manejar los equipos y procesar la información.

En la capa de manejo de datos es donde ocurre todo el análisis y procesamiento de la información que ha sido obtenida. Esta capa posee cuatro módulos importantes que representan la estructura de software que fue implementado y que se detalla en la sección 3.1.3.

En la capa de salida, lo que el sistema busca es proveer un archivo con una extensión PLY que contiene toda la información 3D (nube de puntos) y las definiciones de la reconstrucción de polígonos.

3.1.3. COMPONENTES DEL SISTEMA

Los componentes del sistema más importantes se encuentran en la capa de manejo de datos que está dividido en 4 módulos, que son el Módulo de Adquisición de Datos, Módulo de Procesamiento de Datos, Módulo de Reconstrucción 3D y el módulo de Exportación.

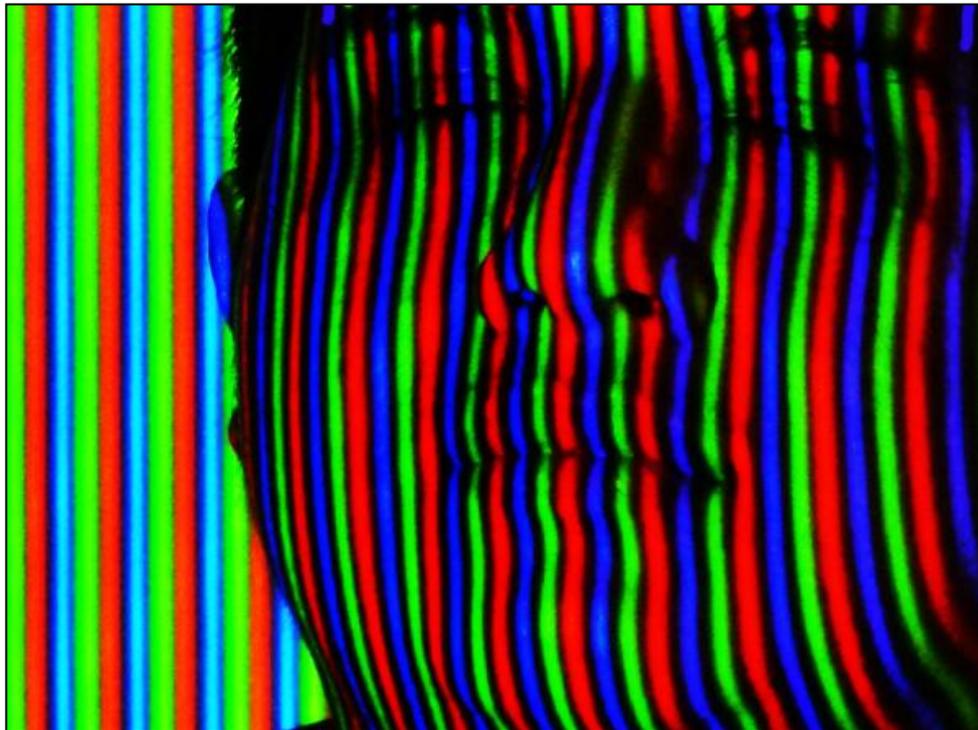


Figura 21. Proyección del Patrón sobre el Usuario.

3.1.3.1. MÓDULO ADQUISICIÓN DE DATOS



Figura 20. Esquema E/S del Módulo de Adquisición de Datos.

Como se puede observar en la Figura 20, este módulo tiene como entrada las imágenes obtenidas por las cámaras. A ambas imágenes se les aplica un proceso de filtrado para resaltar las líneas que han sido proyectadas sobre el rostro del usuario.

La información de cada pixel de las imágenes es guardada en las matrices L y R correspondientes a datos de la imagen izquierda y derecha respectivamente.

Filtros Importantes en este módulo:

- Int CountLines(IplImage*frame, intLoR)
- Void RealzarColoresBajos(IplImage*frame)
- Void ResaltarLineas (IplImage *frame1 , intLoR)

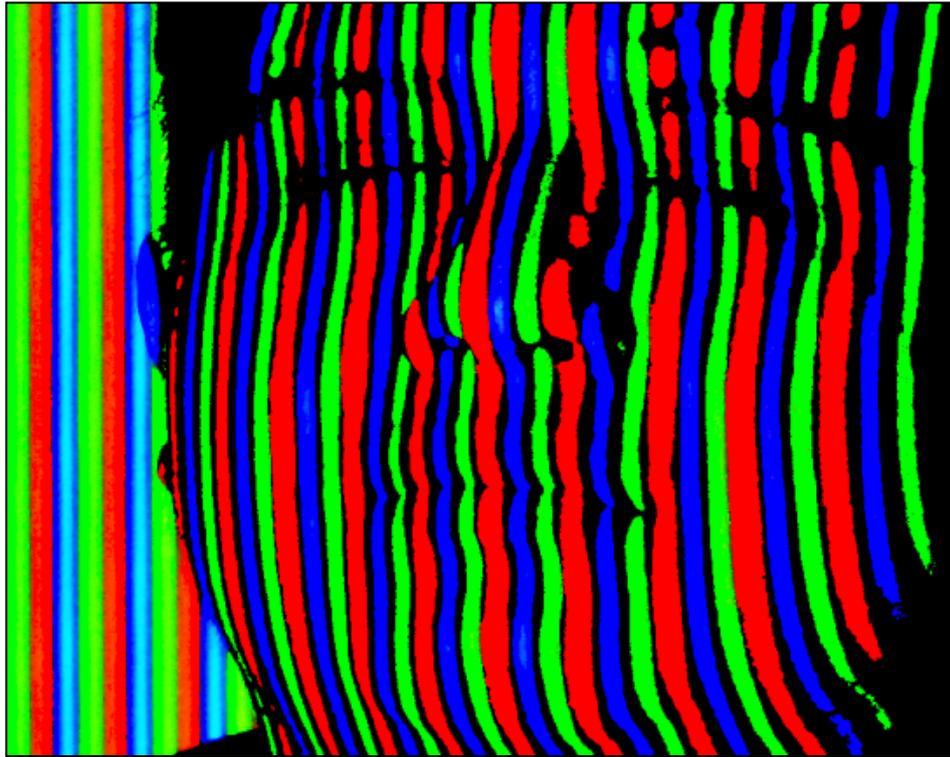


Figura 21. Filtro que detecta niveles de colores para Clasificarlos.

Una vez aplicada la función `CountLines`, esta se encarga de hacer un barrido por todos los pixeles censando su color. Si la función tiene certeza de que un color pertenece al Rojo, Verde o Azul le asigna una saturación máxima de ese color. Después de esto se pasa la imagen por un filtro en la función `RealzarColoresBajos` que busca los colores cuyo nivel es muy bajo, y si estos no pueden ser clasificados se los pinta con Negro. Los demás colores, en los que sus niveles son bajos pero tienen suficiente información para reconocerlos, se los pinta con una saturación máxima dando como resultado

colores RGB con un solo nivel de color; todo lo demás no es considerado tal como se observa en la Figura 22.

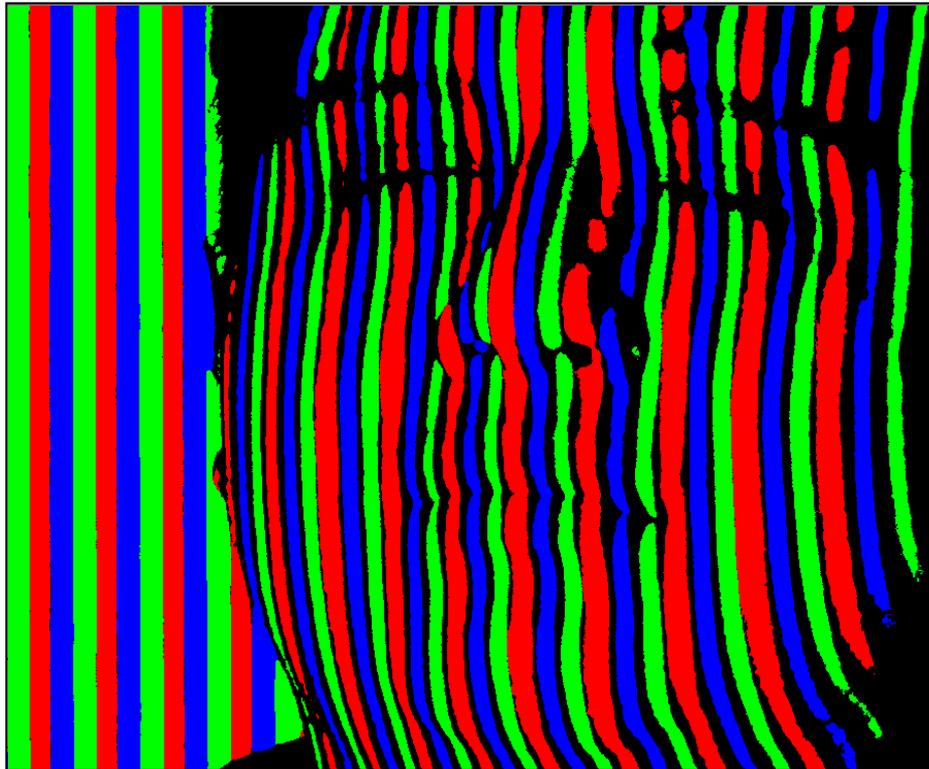


Figura 22. Eliminación de Colores que no sean RGB y se infiere niveles cercanos a estos.

Con los últimos filtros mencionados se obtiene solo la información más importante de las imágenes, en la cual el patrón es claramente resaltado.

Finalmente, se pasa por el filtro ResaltarLineas que deja las líneas con un ancho de un pixel tal como se indica en la Figura 23. Estos pixeles son los vértices que van a ser tomados en cuenta en el proceso de reconstrucción.

Este filtro se aplica a la imagen de izquierda a derecha en el caso de imágenes tomadas con la cámara izquierda. Al encontrar una línea, su propósito consiste en asignarle ese índice de línea a todos los vértices de esta. Esto lo logra con un algoritmo que va aplicándose por la línea desde el inicio hasta el final. El algoritmo se enfrenta a casos de sombras u oclusiones en los que se pierde la continuidad de las líneas. Debido a que se empieza de izquierda a derecha y que se considera que estas funciones tienen un elevado grado de éxito, se asume que todas las líneas a la izquierda de las líneas en cuestión ya han sido catalogadas y por ende no son tomadas en cuenta. A su vez por la perspectiva que se tiene (Izquierda en este caso), las continuaciones de las líneas tienden a ubicarse en una sección hacia el lado izquierdo de donde se perdió la continuidad. Una función `checkLeft`, cuyo propósito es buscar líneas a la izquierda, busca la continuación de la pérdida de continuidad. Por ende el algoritmo que viaja por las líneas retoma la ruta y continúa clasificando cada vértice con la línea que está buscando. Este método posee una tasa de éxito considerable, siempre y cuando no se presenten casos extremos de sombras y oclusiones, ya que estos ocasionan confusión en las funciones de búsqueda.

Una vez realizado este proceso con la perspectiva izquierda, este se repite para la perspectiva derecha cambiando el orden en que las líneas son consultadas. Se empieza la búsqueda de líneas de derecha a izquierda, y la sección donde se encuentra continuidad de líneas en casos de oclusión y

sombras será hacia la derecha de la pérdida de continuidad. La función que busca la continuidad en este caso se llama checkRight.

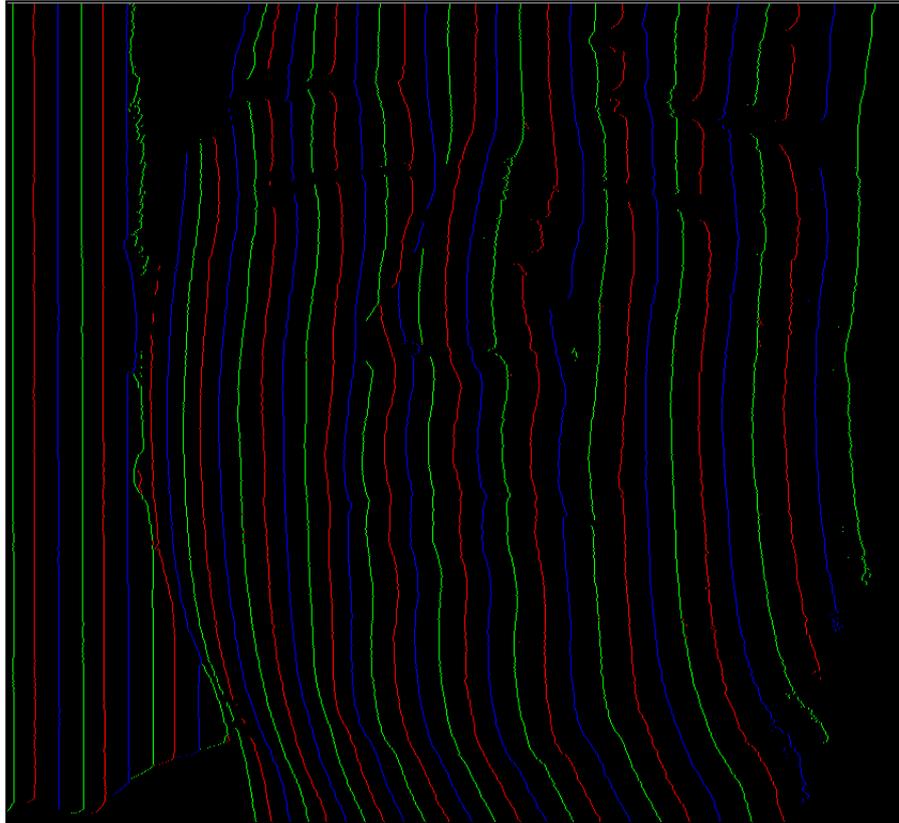


Figura 23. Se resaltan líneas a nivel de pixel.

3.1.3.2. MÓDULO DE PROCESAMIENTO DE DATOS

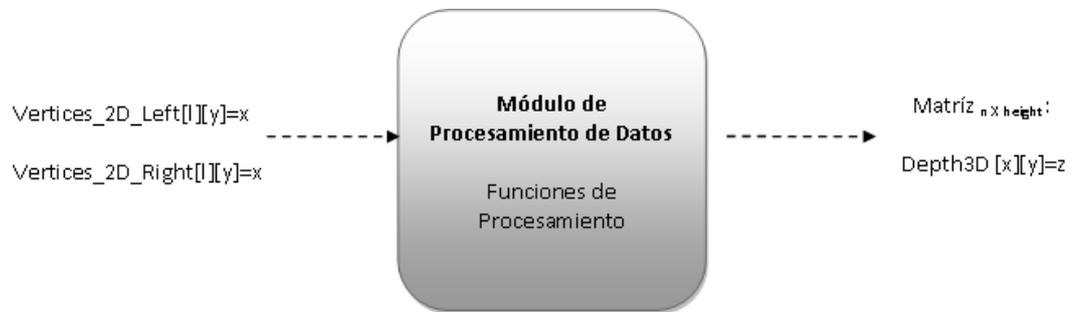


Figura 23. Esquema E/S del Módulo de Procesamiento de Datos.

Como se aprecia en la figura 23, este módulo tiene como entrada 2 matrices con las posiciones de cada vértice ordenados según la línea. Una columna de la matriz representa una línea vertical donde cada fila representa cada vértice perteneciente a esa línea. Al ir al índice de una línea teniendo su posición en 'y' se obtiene su desplazamiento en 'x' que es el deseado para compararlo con el mismo vértice en la otra imagen y así medir la distorsión y poder triangular su posición en forma efectiva.

Funciones importantes en este módulo son las siguientes:

- Void ProcesarLineas (IplImage *frame , intLoR)
- Void LlenarCasillasVacias (void)

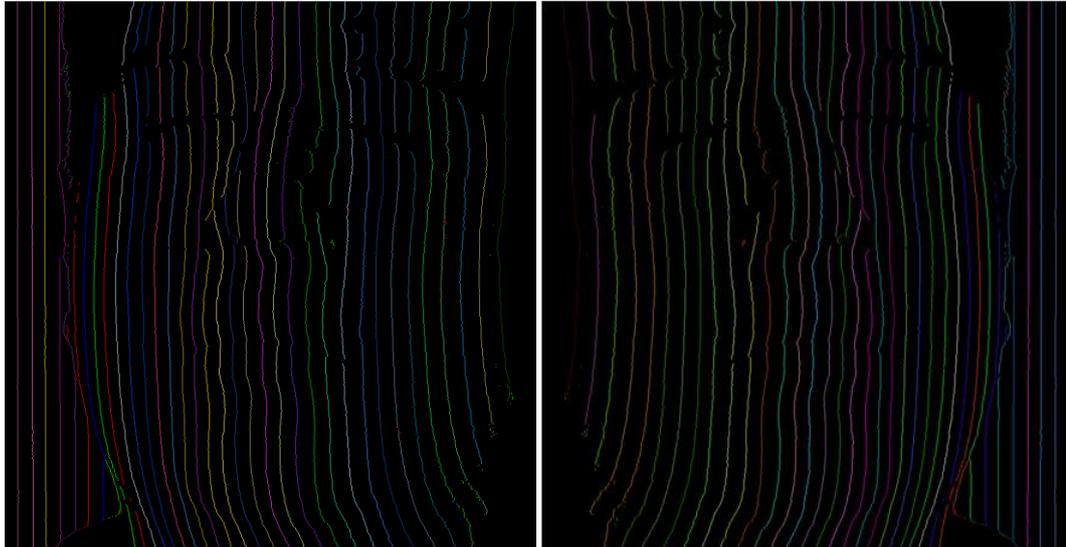


Figura 24. Función de Procesamiento Clasifica las líneas.

En la Figura 24, puede visualizarse la clasificación de cada una de las líneas una vez que estas han sido correctamente indexadas en ambas imágenes. Se compara en orden la primera línea de la imagen izquierda con la de la derecha. Cabe recalcar que en realidad esto corresponde a un mismo punto del rostro que aparece en ambas imágenes. La aplicación de ambas mediciones en la nueva fórmula matemática nos da como resultado la profundidad del punto. Una vez terminada la primera línea se avanza a la siguiente en ambas imágenes para repetir el proceso. Esto se realiza hasta haber comparado todas las líneas que fueron proyectadas.

Al finalizar este proceso se obtiene una Matriz $Depth3D[x][y]$, en la que 'x' contiene la información de las líneas, 'y' la cantidad de vértices para la línea

que se está consultando, coordenadas para la cual la matriz guardó una medida de profundidad, completándose la información 'x', 'y' y 'z'.

La función LlenarCasillasVacias como su nombre lo indica busca en la matriz datos vacios que aparecen debido a falta de información que no pudieron ser detectados por las cámaras y que normalmente son causadas por oclusiones, sombras y que también podrían corresponder a casos de confusión por parte del algoritmo. Este inconveniente se resuelve llenando estas casillas críticas de la matriz con información similar de las casillas vecinas.

3.1.3.3. MÓDULO DE RECONSTRUCCIÓN 3D

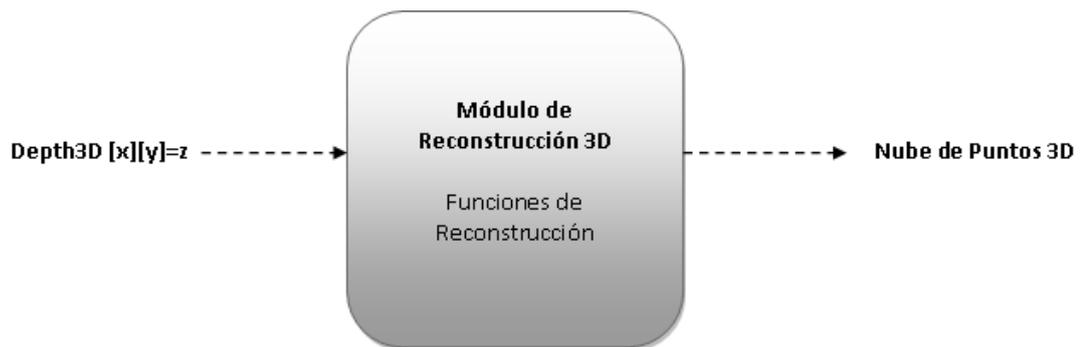


Figura 25. Esquema E/S del Módulo de Reconstrucción 3D.

En este módulo, el objetivo principal es usar la información 3D de todos los vértices calculados como se observa en la Figura 25. Según la posición 'x' y 'y' de cada vértice, la matriz devuelve el valor de 'z' que representa el valor de profundidad obteniéndose de esta manera una nube de puntos en el espacio. Además, este módulo se encarga de reconstruir los polígonos a partir de los nodos que tiene a su disposición. El método de reconstrucción se lo implementó para propósitos de visualización, por lo que la triangulación de mesh es no adaptativa.

La función que se encarga de hacer la reconstrucción es:

- `Void ManejarDepth3D(Depth3D[x][y])`

Esta función se encarga de mostrar la reconstrucción que fue implementada con el uso de las herramientas provistas por OpenGL. Para definir un polígono, que para este caso se eligieron triángulos, es necesario definir la posición espacial de 3 vértices para construirlo. Debido a que se tiene una matriz de puntos de X por Y, se pueden definir 2 triángulos con 4 vértices como se observa en la figura 26.

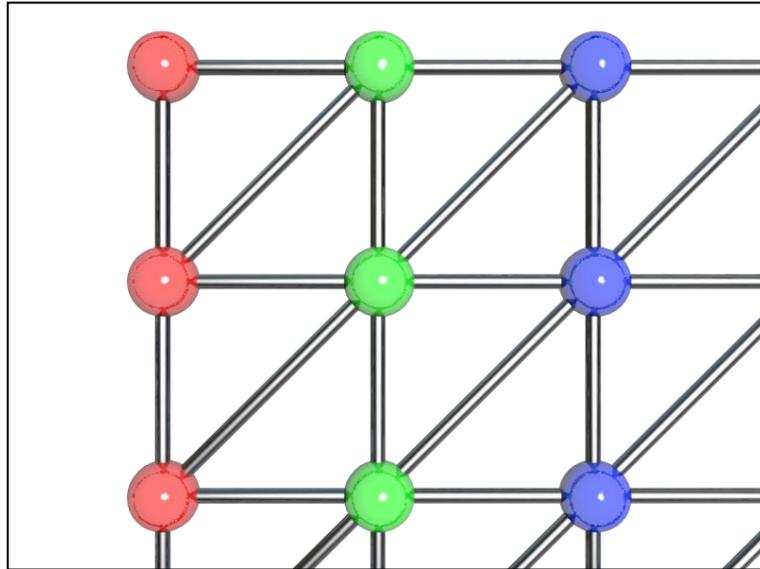


Figura26. La Reconstrucción de Polígonos una cada vértice con los vértices vecinos.

En el código que muestra a continuación se detalla cómo se definen los 2 triángulos generados por 4 vértices. La función se encarga de recorrer toda la matriz e ir generando todos los polígonos con la información 'X', 'Y' y 'Z'. Una vez terminado se tendrá una malla (mesh) en 3D de la reconstrucción del rostro. La cantidad de vértices de la nube de puntos resultantes varía según la cantidad de líneas que fueron detectadas.

```
glBegin(GL_TRIANGLES);
    glVertex3f(x,y,Depth3D[x][y]);
    glVertex3f(x+1,y,Depth3D[x+1][y]);
    glVertex3f(x,y+1,Depth3D[x][y+1]);
glEnd();
glBegin(GL_TRIANGLES);
    glVertex3f(x+1,y,Depth3D[x+1][y]);
    glVertex3f(x+1,y+1,Depth3D[x+1][y+1]);
    glVertex3f(x,y+1,Depth3D[x][y+1]);
glEnd();
```

El mismo método de reconstrucción se usa para el modulo de exportación de archivo que se detalla en la siguiente sección.

3.1.3.4. MÓDULO DE EXPORTACIÓN

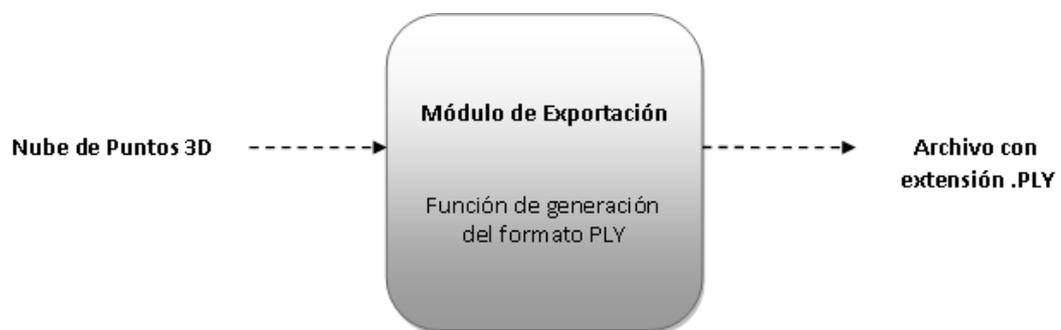


Figura 27. Esquema del Módulo de Exportación.

El módulo de exportación como su nombre lo indica es llevar la información ya reconstruida a un formato que pueda ser leído por los visualizadores de malla (meshes). En la etapa de diseño se eligió el software MeshLab debido a que es abierto y posee varias herramientas de visión que ofrecen complementos en el manejo de modelos 3D además de la funcionalidad de visualización.

La función que se encarga de hacer la exportación es la siguiente:

- Void ExportFunction(void)

Esta función trabaja en la escritura de un archivo al que le asigna una extensión PLY que podrá ser leída por MeshLab. El detalle de la estructura necesaria para que el archivo sea correctamente creado se muestra en la siguiente sección.

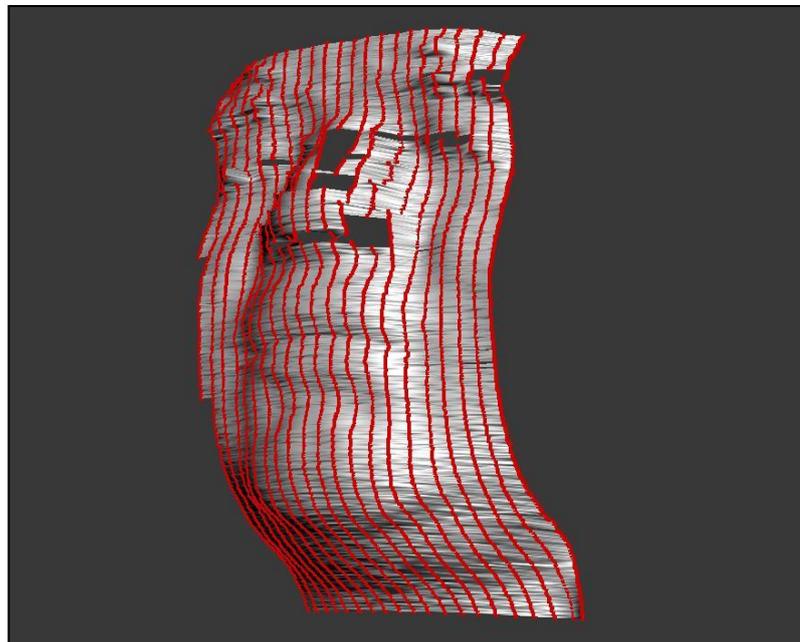


Figura 28. Reconstrucción de polígonos usando la Nube de Puntos 3D.

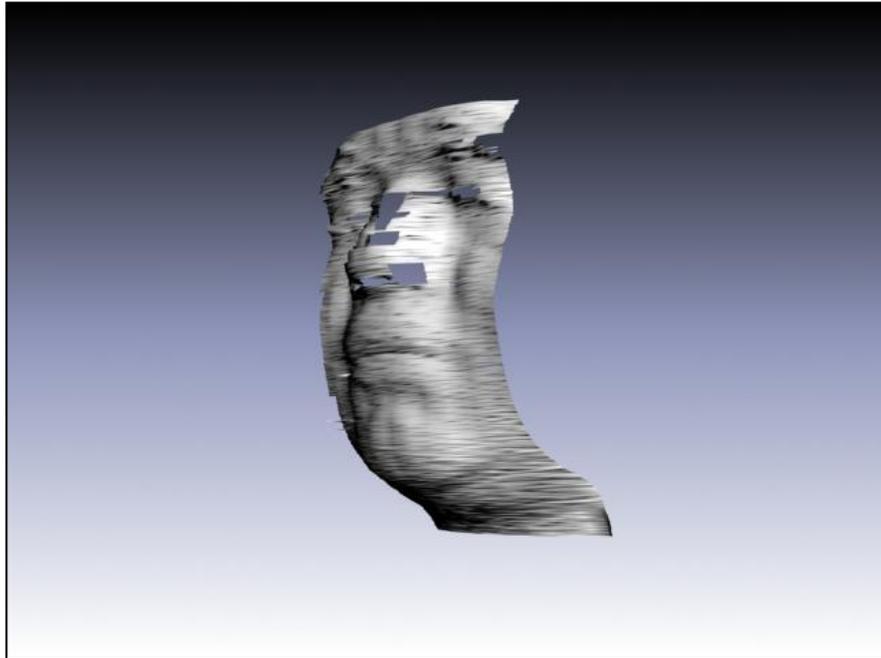


Figura 29. Modelo 3D del Usuario.

3.1.4. FORMATO DEL ARCHIVO CON EXTENSIÓN PLY PARA EXPORTACIÓN

La extensión PLY es un formato de archivo conocido como Polygon File Format el cual define la nube de puntos en el espacio. Este formato de archivo posee una estructura específica donde se pueden ir definiendo algunas propiedades del modelo 3D que se está exportando. Para propósitos de definir la información 3D, las principales propiedades consisten en ir listando las coordenadas 'X', 'Y' y 'Z' de cada vértice; haciendo esto, se define una nube de puntos en el espacio. Luego de haber definido los puntos

es necesario definir todos los polígonos de la reconstrucción, por lo que se crea otro listado especificando el número de los 3 vértices que conforma todos los polígonos de la malla (que en este caso son triángulos). Por tanto, para crear el archivo es necesario la escritura de un formato de encabezado que define la cantidad de vértices que van a ser listados, la cantidad de caras o polígonos, y el tipo de polígonos que se está manejando. A continuación se muestra el formato de encabezado.

Formato de Encabezado:

```
ply
formatbinary_little_endian 1.0
comment VCGLIB generated
element vertex 60749
property float x
property float y
property float z
property float nx
property float ny
property float nz
propertyint flags
element face 118126
property list ucharintvertex_indices
propertyint flags
end_header
```

3.2. LIMITACIONES DEL DISEÑO

La concepción del presente diseño tuvo el objetivo principal de escanear rostros, por lo que fue implementado para seguir las líneas de la superficie del rostro en el que se espera que las oclusiones que se obtengan en el proceso de adquisición sean muy pequeñas.

Una condición bajo la cual esta aplicación funciona de manera óptima corresponde a aquella en la que la iluminación ambiental no afecte el proceso de adquisición ya que esto afecta el brillo de las líneas y su color.

Especificaciones:

- Las cámaras deben enfocar a la cara.
- Las cámaras deben tener el mismo ángulo de rotación.
- Los lentes deben estar a la misma distancia del proyector
- La luz ambiental debe disminuirse al máximo. Los escaneos son realizados en cuartos oscuros para que el patrón se resalte y no haya ruido en la adquisición.

3.3. ALCANCE DEL DISEÑO



Figura 30. Proyección del Prototipo.

El prototipo y software que se desarrollaron fueron construidos con el propósito de demostrar la técnica de triangulación de puntos a partir de visión estereoscópica. Para este propósito, se implementaron algoritmos que analizan la profundidad de vértices que son detectados en rostros de personas.

Dado que el proyecto se enfoca específicamente en el objetivo de digitalizar rostros de personas, la geometría fuera de la zona de interés no es un objetivo de escaneo al momento de digitalizar a un usuario. Se debe indicar

que el cabello no puede ser digitalizado debido a la complejidad de la textura y por su propiedad de absorción de la luz que es proyectada.

3.4. DISEÑO DE LA PLATAFORMA PARA LOS EQUIPOS

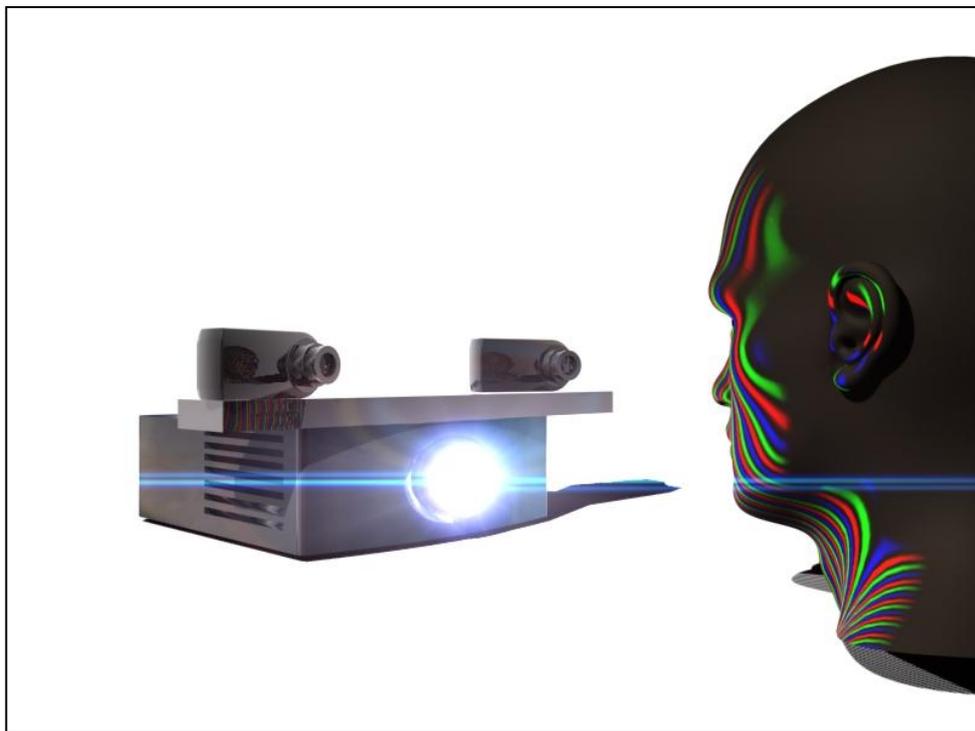


Figura 31. Ilustración del funcionamiento del equipo.

El prototipo está conformado por 2 cámaras con iguales especificaciones y un proyector. Parte esencial del prototipo lo constituye la correcta alineación y separación de las cámaras según la calidad deseada.

Cabe recalcar que cuando la separación de las cámaras supera los 30 cm aumenta la probabilidad que existan sombras de significancia y con ello se presentarán errores de reconocimiento e indexado de las líneas.

El sistema de proyección funciona a través de un proyector LCD común (off the shelf) muestra la proyección generada por un computador. La implementación permite proyectar en pantalla completa el patrón de manera automática para que el computador tome ambas fotos simultáneamente, o estas puedan ser tomadas de manera manual. Es necesario colocar el proyector sobre una plataforma que se eleve lo suficiente para proyectar el patrón al nivel del rostro. De igual manera, ambas cámaras deben estar aproximadamente a la misma altura. Para propósitos de probar la implementación con el prototipo, el proyector fue ubicado a una altura específica, y ambas cámaras fueron soportadas con 2 trípodes como se observa en las figuras 32, 33. Estas cámaras estuvieron posicionadas a la misma distancia del lente del proyector que fue de aproximadamente 20 cm, alejando ambas cámaras a 40 cm de distancia entre sí. La rotación de cada cámara fue de 14° respecto al eje de la distancia del dispositivo y el rostro. La distancia aproximada del lente de proyección hacia el rostro se encontraba en un rango entre 80 y 90 cm.



Figura 32. Prototipo usando un Proyector LCD y 2 cámaras.

Para un prototipo más preciso será necesario construir una plataforma que pueda acoplarse al proyector LCD y puedan colocarse las cámaras a la distancia adecuada entre sí. Una propuesta de diseño se visualiza en las figuras 30 y 31. Una vez colocado de manera correcta, se colocaría un prototipo sobre un trípode para llevar el sistema a una altura adecuada para el escaneo.



Figura 33. Prototipo proyectando Luz Estructurada.

En la tabla 1 se indican los dispositivos y materiales usados en la implementación realizada. Todos estos dispositivos no se catalogan como fuera de alcance económico para instituciones como centros de investigación, universidades o distintos centros de estudio que tengan la necesidad de una funcionalidad de escaneo a bajo costo.

CANTIDAD	MATERIAL	COSTO
1	Proyector LCD EPSON	\$500
2	Cámaras Panasonic LUMIX 14 MP	\$200
2	Trípodes	\$60 (los dos)
1	Soporte horizontal para proyector ⁵	\$20
TOTAL		\$780

Tabla 1. Dispositivos usados en el prototipo

Los equipos necesarios y costos aproximados para poder crear un prototipo se detallan en la Tabla 1. Para la implementación y pruebas, se usó un proyector LCD de marca EPSON con 2600 lúmenes, dos cámaras Panasonic LUMIX modelo DMC-FH22 de 14 MP a color con una resolución de 4320x3240, y el resto de materiales en la tabla. La resolución provista por el proyector es de 1280x800.

Las imágenes capturadas inicialmente por las cámaras LUMIX fueron procesadas para disminuir su resolución a un tamaño de 1280x800 con el propósito de tener un menor volumen de datos. El formato de imagen con el que se trabajó fue JPG.

⁵ Un soporte simple puede ser una mesa pequeña o una caja de una altura específica

3.5. DISEÑO DE PRUEBAS

Las pruebas permiten evaluar la eficiencia de la implementación y para ello deben ser diseñadas apropiadamente.

Las dos partes de las pruebas se orientan los siguientes objetivos:

- Observar el resultado de la aplicación de cámaras en un arreglo canónico (cámaras alineadas en forma completamente paralela), y cómo una rotación convergente con igual ángulo de dichas cámaras en un plano horizontal afecta a la calidad e integridad de los modelos. Este resultado permitirá elegir la alineación final de las cámaras en el resto del trabajo.
- Analizar resultados a partir de un aumento o disminución en la cantidad de colores proyectados, que permita asistir en el desempeño óptimo del algoritmo de clasificación de colores para el correcto indexado de líneas.
- Medir la cantidad de vértices no detectados en la implementación final con la finalidad de analizar el algoritmo de detección de líneas. Con esto se realizará un contraste de la cantidad de vértices detectados para tener una medición de error en la detección.
- Analizar los resultados a partir de un aumento o disminución de la cantidad de líneas en el patrón proyectado para encontrar un rango de la cantidad de líneas óptimas en la implementación.

Las pruebas son divididas en 2 partes, pruebas en condiciones ideales⁶ y pruebas en condiciones reales. El propósito de esto es poder tener una referencia de resultados bajo condiciones ideales de las cámaras, de iluminación y de reflexión. En las pruebas bajo condiciones reales entran una serie de factores externos que afectan el resultado.

La primera parte consiste en pruebas en condiciones ideales, que tienen que ser simuladas con el uso de herramientas como modeladores 3D para poder excluir los factores externos que afectan las pruebas en condiciones reales.

En la segunda parte se harán pruebas con varios usuarios con el prototipo e implementación final. En estas pruebas se espera que factores externos afecten el resultado final. Para este propósito se escanearán varios rostros con rasgos físicos diferentes.

⁶ Condiciones en las cuales el escaneo no es afectado por iluminación externa, ruido de las cámaras, desenfoque de las cámaras, y problemas causados por el balance de blanco de las cámaras que se presentaron en las pruebas reales.

CAPITULO 4

4. IMPLEMENTACIÓN Y PRUEBAS

4.1. IMPLEMENTACIÓN

La implementación del software empezó con un esquema distinto al utilizado en su posterior desarrollo. En sus inicios se empezó con una serie de pares de fotografías (stereo-pair) sobre un barrido de puntos que eran almacenadas en el computador en un proceso de adquisición de datos. A partir de esta serie de imágenes, cada par estéreo (stereo-pair) que pasado por varios filtros que resaltaban la información de los patrones proyectados para luego calcular la profundidad de cada uno de los puntos capturados. Una vez obtenidos todos los puntos de todos los stereo-pairs tomados, se los pasaba por una función de reconstrucción que creaba la nube de puntos en el espacio y se lo visualizaba con el uso de la librería de OpenGL.

Se observó en varias simulaciones que la técnica de proyección de puntos funcionaba de manera adecuada solo en condiciones muy ideales de iluminación y de captura de imágenes. En la práctica, se presentaron muchos problemas causados principalmente por los balances de blanco de las cámaras que afectaban el diámetro de los puntos y por ende causaba

mediciones incorrectas sobre las posiciones de los puntos en cada par estéreo (stereo-pair) por lo que cualquier reconstrucción basada en esa información resultaría en un escaneo distorsionado del modelo.

Como solución se optó una proyección de luz estructurada en forma de líneas verticales que permite tener:

- Un solo par estéreo (stereo-pair).
- Múltiples líneas en una sola toma.
- Una gran resolución en el eje “Y”⁷.

4.1.1. MANEJO DE PÍXELES

El manejo de píxeles es esencial en los módulos que manejan la información de las imágenes obtenidas. Para esto se usan tipos de datos y funciones especiales que son provistas por la librería openCV. A continuación describimos algunos tipos de datos y funciones utilizadas:

- Tipos de datos
 - IplImage
 - Este identificador es el que almacena las imágenes obtenidas en la memoria RAM mientras el programa es

⁷ La cantidad de puntos en el eje ‘Y’ será como máximo la altura de la resolución de las imágenes tomadas por las cámaras.

ejecutado. Básicamente es una matriz con el alto y ancho de la resolución de la imagen que se está manejando. Al acceder a cualquiera de sus elementos se obtiene información de cada píxel.

- Capture
 - Se lo usa para adquirir un flujo de imágenes; a este flujo se le define un dispositivo de adquisición para extracción de imágenes.
- Uchar
 - Tipo de dato usado para consultar los valores RGB de cada píxel que es leído por los filtros y funciones.
- Funciones
 - cvPtr2D(IplImage,int idx0 ,int idx1)
 - Es la función más importante en el manejo de píxeles de este proyecto. Esta devuelve un arreglo de tipo uchar con 3 valores, correspondientes a los niveles de RGB del píxel consultado. La función tiene 3 entradas, la imagen que se está manejando definida en un IplImage, y 2 valores enteros idx0 e idx1. Estos valores representan la posición en 'y' y en 'x' de la matriz respectivamente. Para leer información de una imagen, esta función debe ser usada por cada uno de los

píxeles. Esta función es ampliamente usada por todos los filtros y algunas funciones.

- `cvWaitKey(int milliseconds)`
 - Función usada para que detenga la ejecución de funciones por un tiempo designado. Es usada extensivamente para que la captura de imágenes sea después de que la cámara ha sido apropiadamente configurada y también para la espera de control por teclado del inicio del proceso de adquisición o procesamiento.
- `cvShowImage(IplImage)`
 - Esta función muestra la imagen que es consultada en una ventana. Es mayormente usado para poder ver el progreso de los filtros paso a paso.
- `cvDestroyImage(IplImage)`
 - Las imágenes `IplImage` reservan memoria RAM para el almacenamiento temporal de las imágenes. Esta función libera ese espacio una vez que una imagen ha cumplido su propósito.
- `cvCapture(CAM)`
 - Función que devuelve un identificador tipo `capture`. Asignando un valor 0, devuelve la entrada de video del

dispositivo de adquisición por default. Aumentando el valor a 1 obtendrá la entrada de un segundo dispositivo. Si existen más dispositivos solo se deberá seguir aumentando el valor.

- cvQueryFrame(Capture)
 - Esta función toma una foto de la entrada de video de una cámara. Esto permite almacenarla en un dato `IplImage` para luego ser procesada.
- cvLoadImage(File Location)
 - Función que permite leer archivos de imágenes almacenados en el ordenador especificando la ruta de la localización del archivo de imagen.

4.2. PRUEBAS

4.2.1. PRUEBAS EN AMBIENTE CONTROLADO

Se realizaron pruebas simuladas. Estas consistieron en el uso de imágenes obtenidas en simulaciones realizadas con un modelador 3D, específicamente la herramienta de 3D Studio MAX. En la simulación se usó un modelo tridimensional de un rostro sobre el cual se colocó una proyección del mismo

patrón que estaba siendo usado en las pruebas reales; esto con el fin de igualar las condiciones de un escaneo normal sin los inconvenientes de problemas causados por factores externos. Estas pruebas fueron consideradas “Pruebas en una ambiente controlado”.

Este tipo de pruebas se realizaron en una etapa temprana del proceso de desarrollo de la implementación, y que resultaron ser a su vez de gran ayuda al mostrar también el funcionamiento de los algoritmos desarrollados en condiciones ideales. Este tipo de pruebas representan una referencia para contrastar los resultados obtenidos en ambientes reales.

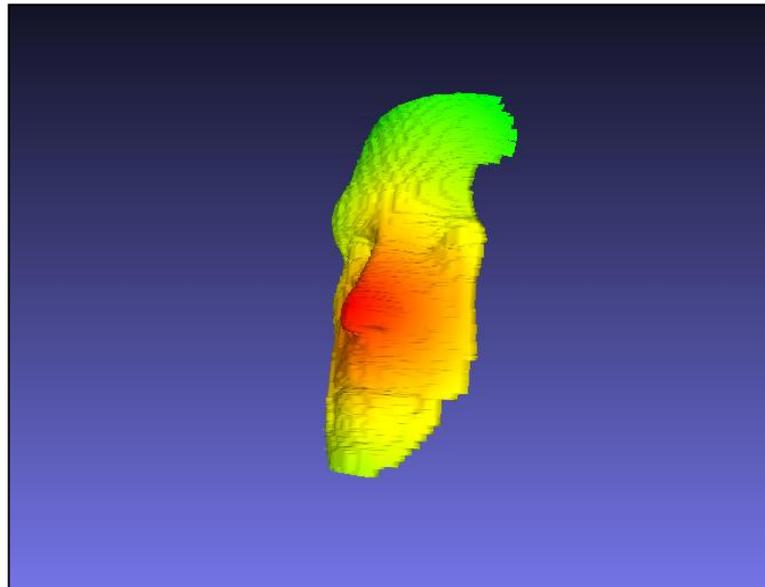


Figura34. Pruebas Iniciales.

En la figura 34, se puede observar la reconstrucción de un rostro en condiciones simuladas. En este caso se uso el modelo canónico, en la que las cámaras no se les aplicó ningún ángulo de rotación. La información de la

superficie del rostro es muy similar al modelo original, pero a su vez se puede visualizar unas leves deformaciones en la superficie del rostro.

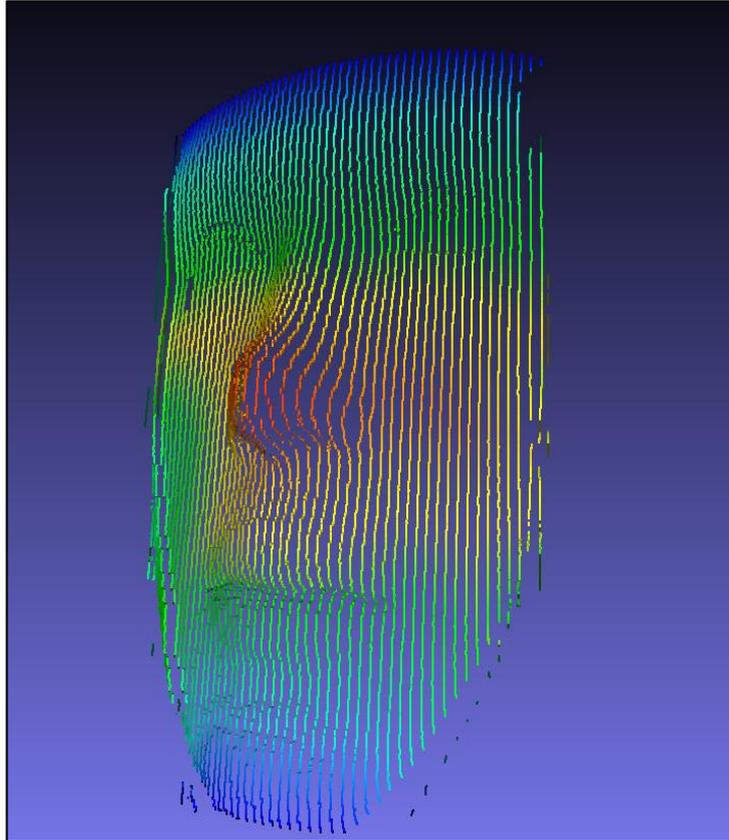


Figura 35. Problema al usar el modelo ideal cuando las cámaras han sido rotadas.

El problema se observa con mayor facilidad en la figura 35, las deformaciones ocurren debido a que no hay una continuidad en los vértices, las líneas tienen como un efecto de pixelado. Esto fue causado al simular el escaneo sin separar mucho las cámaras, lo que genera poca información de deformación de las líneas; este problema en el escaneo puede observarse

con facilidad en la Figura 10. Se debe mencionar que el algoritmo de seguimiento de líneas, a pesar de que no hay mucha información 3D, tiene una mayor tasa de éxito en el indexado de líneas debido a que las deformaciones no son muy grandes, por ende hay una menor tasa de confusión.

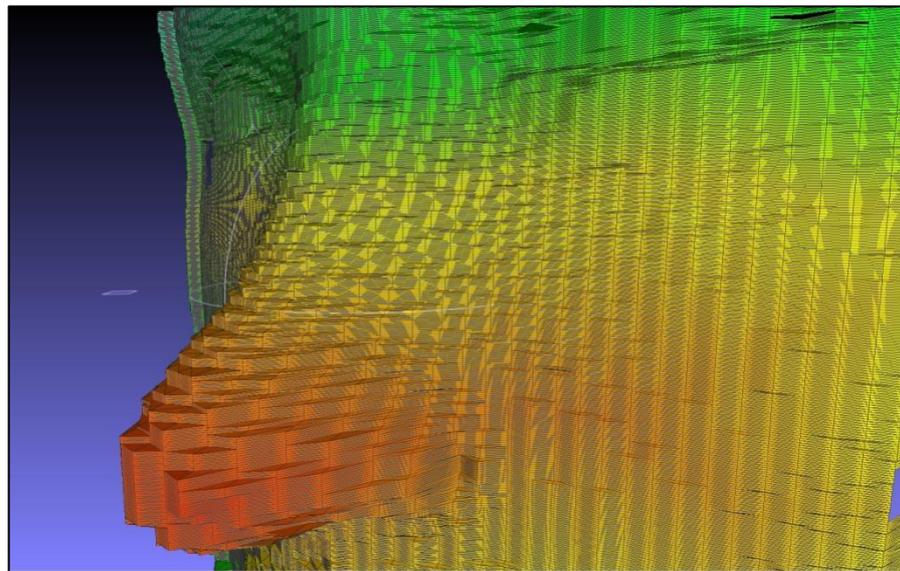


Figura 36. El problema ocasiona un estiramiento en la zona de la nariz y una contracción al nivel de los ojos.

Con el fin de observar una mayor resolución, se optó por separar ligeramente las cámaras, lo cual resultó ser una buena solución pero se observó un riesgo mayor de que el rostro salga de la zona de convergencia de las 2 cámaras, por lo cual se decidió rotar las cámaras sin alterar el modelo matemático. Esto resultó en una deformación a nivel de profundidad de los

datos. En la figura 36 y 37 puede observarse que esto causa que la zona de la nariz se estire y ocurra lo contrario en los puntos al nivel de los ojos.

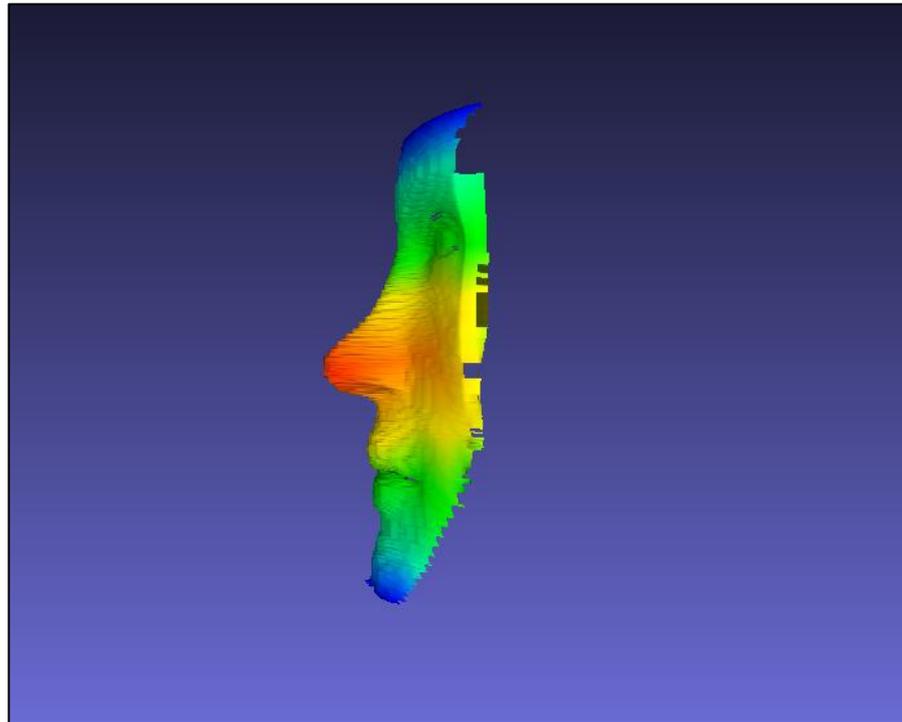


Figura 37. Perspectiva de los efectos causados por el problema que aparece al usar el modelo ideal cuando las cámaras han sido rotadas y no se ha tomado en cuenta el nuevo modelo matemático.

Una vez considerado e integrado el nuevo modelo del arreglo de las cámaras se generó un nuevo escaneo que puede observarse en la figura 38. Debido a que se aumentaron los ángulos, no se presentó el problema de deformación por lo que hubo mayor libertad en separar las cámaras. La variación angular facilitó la obtención de mejores resultados en cuanto a calidad y fidelidad del modelo original.

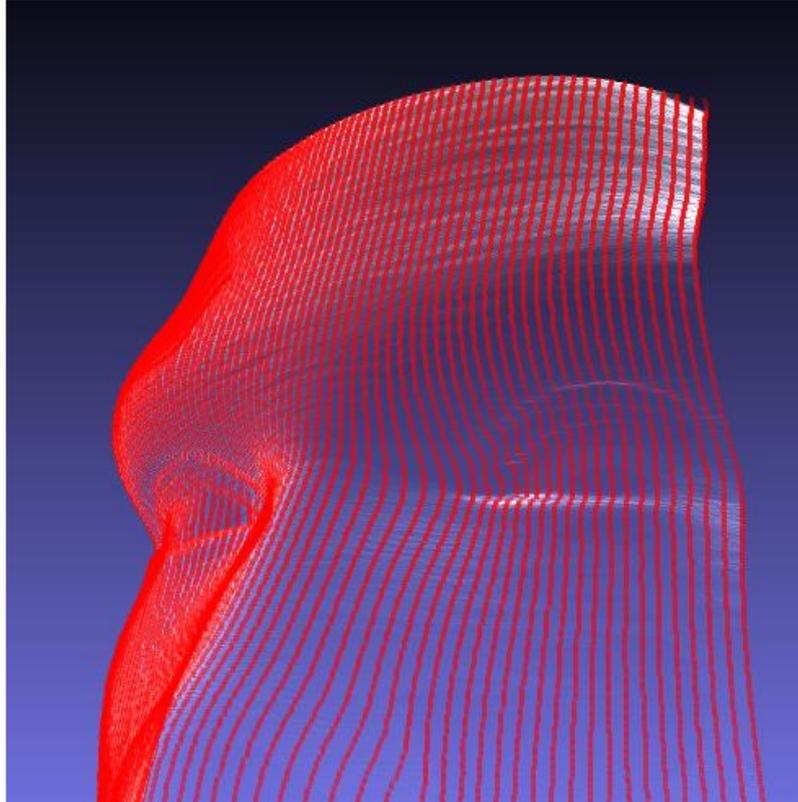


Figura 38. Pruebas usando el modelo matemático que considera la rotación de cámaras.

4.2.2. PRUEBAS EN AMBIENTE REAL



Figura 39. Proyección del Patrón RGB sobre un usuario en condiciones reales.

Las pruebas realizadas en la práctica deben ser catalogadas de manera distinta debido a que se presentan distintos factores externos como por ejemplo, la luz ambiental, el brillo que genera el proyector sobre el rostro, el color de piel que afecta el tono de los colores, y el enfoque de las cámaras y del proyector.

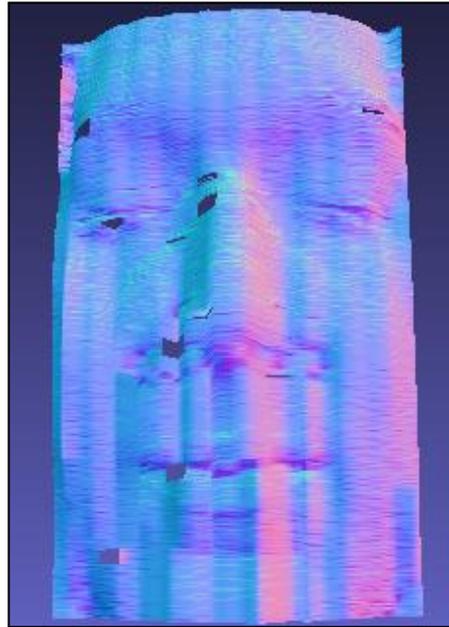


Figura 40. Problemas de mal indexado de líneas.

En la Figura 40 puede observarse el problema de un mal indexado. Esto sucedió en una etapa inicial del desarrollo del algoritmo de seguimiento de líneas. El problema consistió en las funciones de seguimiento al toparse con una oclusión en donde el riesgo de no seguir por la línea correcta aumentaba. La solución a esto fue mejorar el algoritmo de seguimiento con las funciones `checkLeft` y `checkRight`.

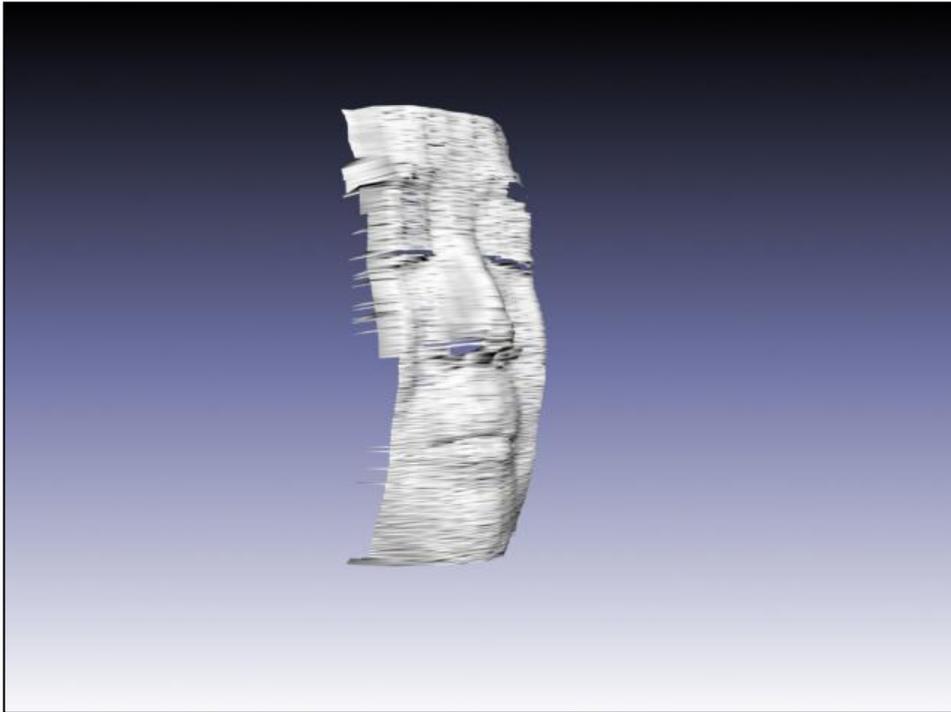


Figura 41. Reconstrucción en Baja Resolución.

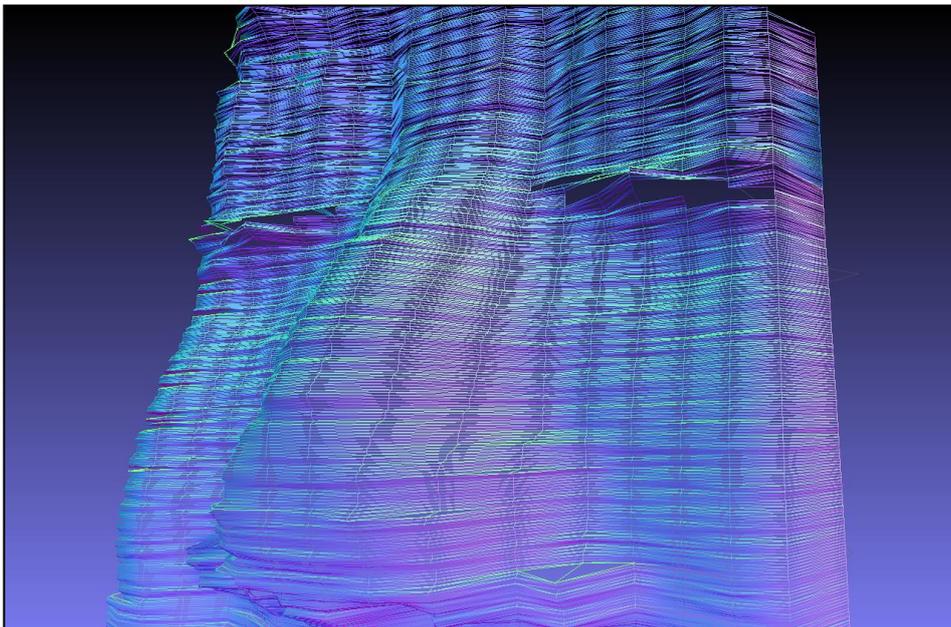


Figura 42. Malla que muestra la cantidad de polígonos generados.

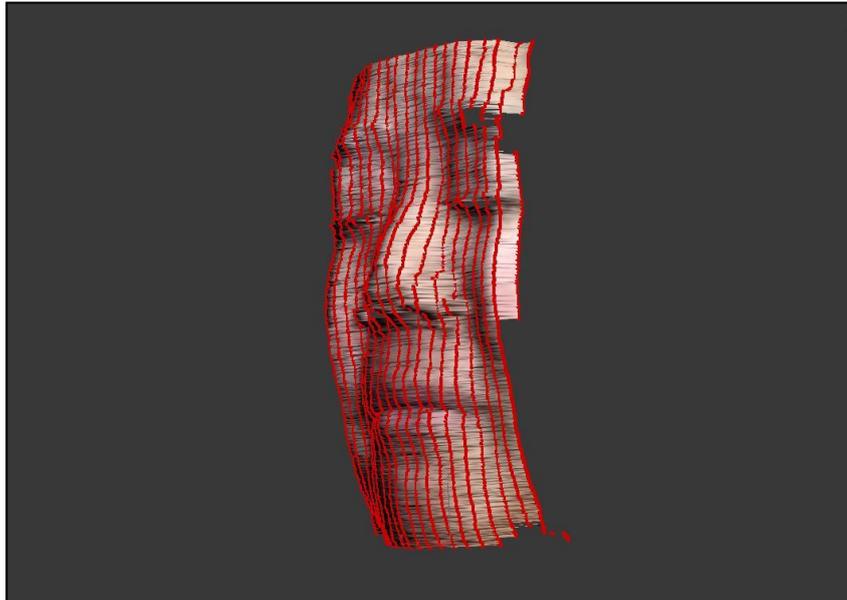


Figura 43. Modelo 3D del Usuario resaltando las líneas detectadas.

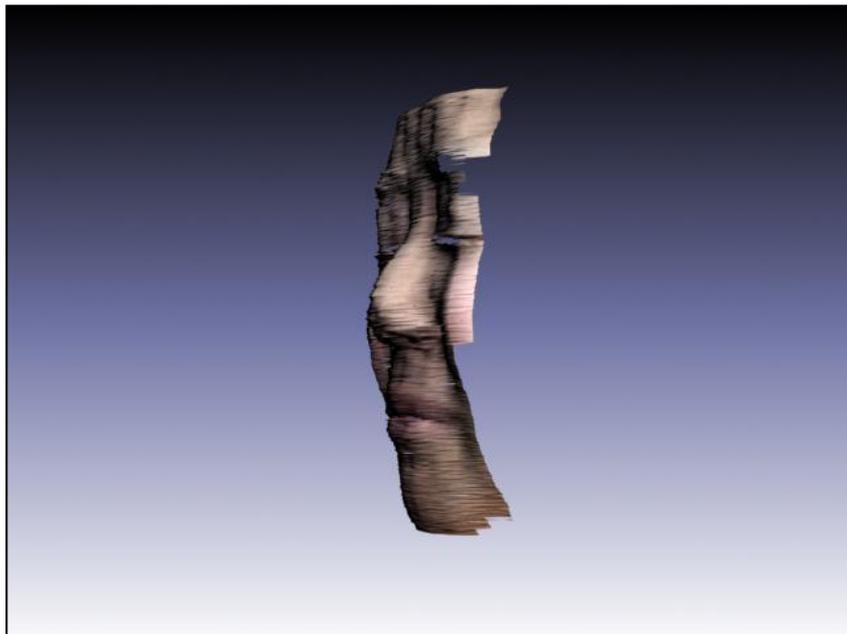


Figura 44. Modelo 3D de Resolución Baja con Textura.

En las Figuras 41, 42, 43 y 44 fue el resultado de una prueba con un bajo número de líneas proyectadas. En la proyección del patrón sólo 20 líneas fueron detectadas sobre el rostro del usuario.

Total de Líneas	Puntos Proyectados	Puntos Detectados	Porcentaje de Error
20	16000	15390	3.81%

Tabla 2. Datos obtenidos con proyección de 20 líneas.

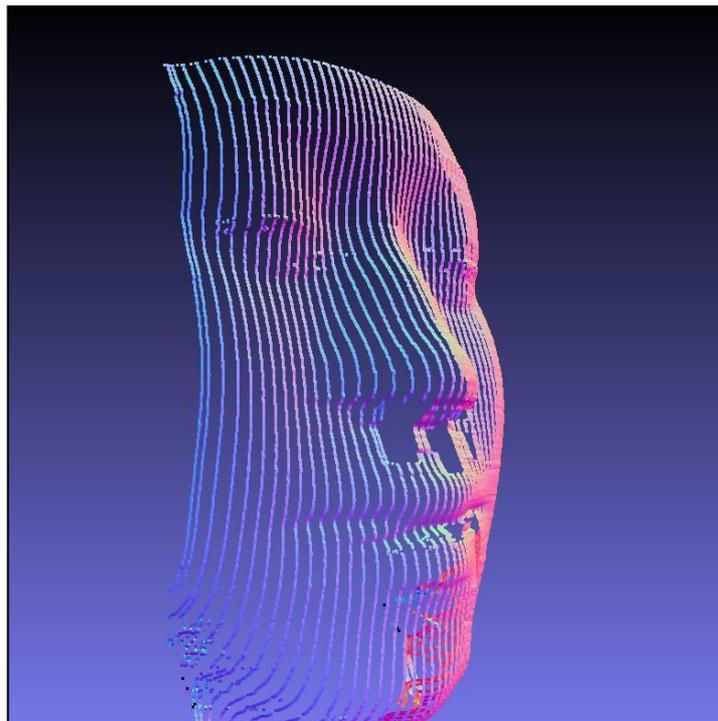


Figura 45. Nube de Puntos 3D de un escaneo con mayor resolución.

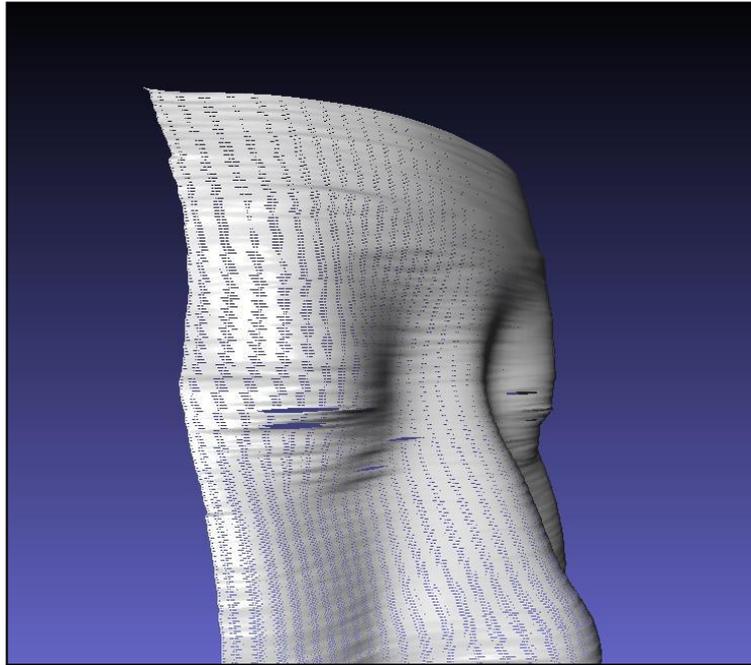


Figura 46. Malla del modelo 3D mostrando cantidad de polígonos sobre el rostro.

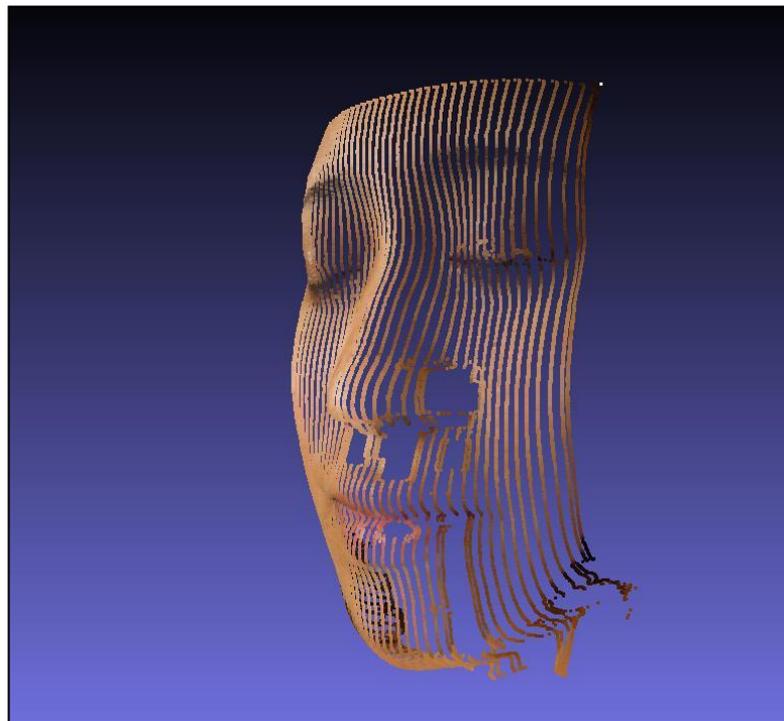


Figura 47. Nube de Puntos 3D.



Figura 48. Modelo 3D de Usuario en mayor resolución con una proyección de 41 líneas.

Total de Líneas	Puntos Proyectados	Puntos Detectados	Porcentaje de Error
41	32800	30930	5.70%

Tabla 3. Datos obtenidos con proyección de 41 líneas.

A continuación se fue aumentando el número de líneas sobre el rostro. En las Figuras desde la 45 a la 48. Se detectaron exitosamente 41 líneas sobre el rostro. Con un 5.70% de puntos que no fueron detectados. Sin embargo, el modelo adquirido fue el de mayor calidad que fue se pudo obtener.



Figura 49. Modelo 3D al que se le proyectó una cantidad de líneas superior a 45.

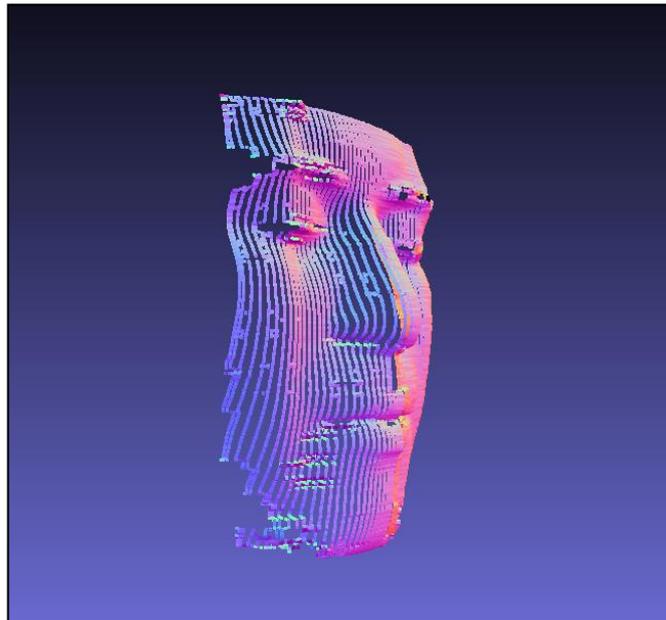


Figura 50. Malla del usuario al que se le proyectó una cantidad de líneas superior a 45.

Total de Líneas	Puntos Proyectados	Puntos Detectados	Porcentaje de Error
49	39200	32077	18.17%

Tabla 4. Datos obtenidos con proyección de 49 líneas.

Al aumentar el número de líneas proyectadas a 49, el porcentaje de error subió a 18.17% y la calidad del modelo se vio afectada como se muestra en las Figuras 49 y 50.



Figura 51. Usuario usado en la explicación de los módulos.

Total de Líneas	Puntos Proyectados	Puntos Detectados	Porcentaje de Error
22	17600	16478	6.38%

Tabla 5. Datos obtenidos con proyección de 22 líneas.

En la Figura 51 se tiene el resultado final del modelo obtenido en la explicación del Capítulo 3.

4.3. ANÁLISIS DE RESULTADOS

A partir de la información obtenida en los 2 tipos de pruebas se pudieron examinar los siguientes puntos:

4.3.1. PRUEBAS REALES Y SIMULADAS

Se usó el mismo algoritmo tanto en las pruebas simuladas como en las reales. Se observa la existencia de un porcentaje de error en cada escaneo realizado a distintas personas del CTI. En estas pruebas se midieron la cantidad de líneas que eran proyectadas sobre el rostro. Cada una de estas líneas está conformada por 800 vértices por lo que puede calcularse la cantidad de puntos totales proyectados sobre el rostro en un escaneo específico. Una vez realizado un escaneo y digitalizado en el ordenador, con el uso de la herramienta de MeshLab,

se obtiene el valor de puntos que fueron usados en la reconstrucción del modelo 3D. Este mismo valor es también usado para contrastarlo con la cantidad de puntos que debieron ser detectados. Por tanto, para cada prueba (real) implementada se calculó el porcentaje de pérdida de información.

En las pruebas simuladas no se presentan factores externos que afecten directa o indirectamente al rostro en el momento del escaneo, por lo que trabajar con simulaciones permite desarrollar y mejorar los algoritmos en etapas iniciales.

4.3.2. USO DE UN ARREGLO DE CÁMARAS GIRADAS CON UN ÁNGULO

Cuando se realizaron pruebas iniciales simuladas, los primeros resultados mostraban una correcta detección de las distancias relativas entre los puntos como se muestra en la figura 35, pero a su vez se notaba una baja resolución. Esto se debe a poca información tridimensional existente y por ende la solución es aumentar la distancia entre las cámaras. Esta solución presenta el inconveniente de que no se obtendrá suficiente información sin comprometer que el rostro se mantenga dentro de la zona de convergencia.

Al separar los dispositivos de adquisición, se hace necesario también rotar las cámaras ya que la rotación de las cámaras ofrecerá una mejor calidad en el escaneo al detectar las líneas con un mayor desplazamiento. Se debe recalcar que si se exagera la separación de las cámaras, hay un mayor riesgo de que existan oclusiones y sombras. Estas suelen ser ocasionadas por las mejillas y nariz.

4.3.3. USO DE COLORES EN EL PATRÓN

El poder indexar correctamente todos los vértices de las líneas en cada imagen es la clave para un buen resultado de un escaneo. Para esto, es necesario que los filtros y algoritmos puedan inferir con seguridad a qué línea pertenece cada punto. Esto se lo logra encontrando el inicio de una línea y moviéndose por ésta hasta que llega a su fin.

En pruebas iniciales del uso del patrón de líneas solo se usó el color blanco, el cual funcionaba adecuadamente en casos ideales de iluminación y de proyección de pocas líneas. Esto no era adecuado debido a que se adquiría un modelo de baja resolución y que era muy susceptible de confundirse en la clasificación de las líneas. A medida que se aumentaba el número de líneas había una mayor confusión con líneas vecinas.

Al agregar los colores RGB se disminuyó éste problema debido a que son los colores con mayor contraste entre sí. Esto permitió agregar una mayor cantidad de líneas al patrón y así obtener una mayor calidad.

Se intentó agregar los colores cyan, amarillo, y magenta al patrón para disminuir aún más la posible confusión del algoritmo. Pero esto en realidad ocasionó una mayor confusión debido al parecido que tienen estos colores entre sí. El magenta es similar al rojo, el cyan es confundido con azul, y el amarillo podía ser confundido en ocasiones con el rojo. Además, los colores variaban según el color de la piel de la persona que estaba siendo escaneada. Este problema no se presenta con el patrón RGB.

Los resultados del uso de más colores fueron de muy baja calidad ya que los modelos no eran representativos del rostro que fue escaneado.

4.3.4. CANTIDAD DE LÍNEAS

La cantidad de líneas proyectadas es un factor de la calidad de modelo que se va a obtener. Cuando se proyectaron patrones con una cantidad baja de líneas había un menor riesgo de confusión por los algoritmos. Esto se debe a que las líneas proyectadas están más separadas y el riesgo de confusión con las líneas vecinas disminuye.

Un ejemplo de esto puede verse en la Figura 45. En este modelo se detectaron 20 líneas, por lo que en el mejor de los casos se pueden detectar 16000 vértices. En la prueba se detectaron 15390 vértices, esto es, el 3.81% no fue detectado.

Al aumentar el número de líneas puede observarse una mejora en la calidad. Esto aumenta la resolución horizontal, y aumenta la calidad del modelo. Este mejor resultado en cuanto a calidad y fidelidad se aprecia en la figura 48. En este escaneo se detectaron 41 líneas, y se detectaron 30930 vértices con un 5.70% de error.

Cuando se agregaron más líneas al patrón la calidad comenzó a disminuir ya que las líneas se acercan más a sus líneas vecinas y aumenta la cantidad de errores.

La experiencia obtenida indica que es recomendable que se proyecten de manera correcta entre unas 30 y 40 líneas en cada escaneo. Menos de 30 serán correctamente detectadas, pero la resolución final será baja. Cabe indicar que hacer escaneos a baja resolución puede ser recomendable si el modelo es aplicado con herramientas con subdivisión de polígonos y suavizado.

Los porcentajes de error obtenidos en las pruebas varían debido a distintos factores que fueron esperados al momento de hacer pruebas reales y que son los siguientes:

- Iluminación al momento del escaneo.
- Color de la piel.
- Cantidad de líneas detectadas sobre el rostro.
- Sombras generadas por el proyector.

La iluminación durante los escaneos fue complicado de controlar debido a que en el lugar escogido existía luz ambiental que afectaba la calidad de imágenes obtenidas. Los distintos escaneos realizados variaban según el nivel de luz existente y con ello se afectaba cada adquisición.

El color de piel de los usuarios afectó ligeramente a los escaneos. En razón del alto contraste de los colores escogidos en el patrón, el riesgo de que este factor afecte el resultado disminuyó considerablemente.

La cantidad de líneas es un factor que visiblemente afecta los resultados. Esto pudo notarse al agregar demasiadas líneas. Como se mencionó anteriormente, esto aumenta el riesgo de confusión por parte del algoritmo debido a la mayor proximidad de las líneas.

Finalmente se debe indicar que si el proyector no está correctamente alineado con el rostro, se generará sombras más grandes, las mismas que representan vacíos de información en las imágenes y que generan secciones que no podrán ser reconstruidas.

4.3.5. LUZ ESTRUCTURADA MODULADA

La técnica conocida como luz estructurada modulada se basa en la proyección de varios patrones estructurados [10], con el fin de codificar cada vértice con un código binario que es generado con todos los valores en cada una de las imágenes para un mismo pixel. Con ese código es posible identificar la línea a la que pertenece cada punto sin la necesidad de implementar un algoritmo de seguimiento de líneas. El uso de este nuevo método ofrecerá mejores escaneos en cuanto a calidad debido a la posibilidad de hacer un barrido del patrón de líneas verticales, sin embargo la complejidad de las funciones de indexado serán ligeramente mayor. A su vez, los costos de prototipos con este tipo de implementaciones involucrarán el uso de dispositivos de adquisición más avanzados debido al grado de control que debe ser realizado por un ordenador en el momento de la captura de imágenes.

CONCLUSIONES

- 1) En este proyecto se pudo implementar un escáner de rostros con una calidad media, digitalizando el resultado en una nube de puntos 3D que puede ser exportada a otras herramientas de manejo de mallas (meshes) para el uso y mejora de los modelos obtenidos.
- 2) A medida que se separan las cámaras se obtiene una mayor información tridimensional pero con la desventaja de que se presentan oclusiones y sombras con más facilidad y por ende el seguimiento de patrones (algoritmo) estará sujeto a una mayor tasa de error.
- 3) El aumento en el número de líneas mejora la resolución a nivel horizontal, pero a su vez acerca a cada una de las líneas a sus líneas vecinas aumentando el riesgo de confusión por parte del algoritmo. Se encuentra apropiado que se proyecten de manera correcta entre unas 30 y 40 líneas en cada escaneo ya que menos de 30 serán correctamente detectadas, pero la resolución final será baja.
- 4) El aumento en el número de colores que tienen un alto contraste entre sí mejora la detección y el correcto indexado de todos los vértices. Aumentar colores con características cercanas aumenta el riesgo de que exista confusión al momento de la clasificación de cada color. El uso de mejores dispositivos de adquisición sería de gran ayuda para

prevenir este problema, sin embargo se debe notar que esto representa un costo superior.

- 5) El uso de un patrón de líneas verticales fue muy efectivo para obtener una buena resolución de manera vertical. La resolución horizontal depende del número de líneas detectadas sin exceder de 45 (líneas). La implementación de un barrido del patrón ya seleccionado podrá aumentar aún más la resolución horizontal pero a su vez aumenta la complejidad de los cálculos que deban realizarse.
- 6) Los escaneos usando luz estructurada tienen el potencial de proveer modelos de rostro tridimensionales de una calidad suficiente para muchos tipos de aplicaciones pero es necesario controlar varios factores externos como la iluminación ambiental, y el arreglo de los dispositivos para su correcto funcionamiento.

RECOMENDACIONES

- 1) Para futuras implementaciones de escáneres a bajo costo se recomienda el uso de luz estructurada modulada; esta técnica es un poco más compleja y requiere de dispositivos de adquisición con un costo mayor, pero ofrece mejores resultados debido a que elimina la necesidad de tener algoritmos de seguimiento y los problemas con el indexado son casi inexistentes.
- 2) La decisión del lugar en donde se realicen los escaneos con luz estructurada debería realizarse evitando en lo posible cualquier tipo de luz ambiental que afecte el proceso de adquisición de los datos. Preferiblemente se debe escoger un lugar bien oscuro.
- 3) Controlar los factores de iluminación y cantidad de líneas proyectadas además de usar el arreglo de cámaras con una rotación para que simultáneamente permita obtener mayor información 3D y evite salir de la región de convergencia de ambas cámaras.
- 4) Aplicar, en los modelos obtenidos, funcionalidades extras tales como la subdivisión de polígonos o las funciones de suavizado. Estas pueden obtenerse mediante herramientas técnicas como MeshLab. A su vez, existen herramientas de diseño 3D que permiten manipular directamente los polígonos para rellenar cualquier vacío de información y mejorar el resultado final.

- 5) Aumentar la resolución horizontal en las futuras implementaciones, implementando la interpolación entre las líneas con el fin de proveer mayor información 3D. Se debe notar que dicha implementación aumentará la cantidad de vértices y por ende la complejidad de la triangulación además del peso en el almacenamiento de los modelos. Es por esto que se recomienda a su vez un análisis de la carga computacional en el proceso de adquisición, procesamiento y reconstrucción para poder medir la eficiencia de los algoritmos a implementar.
- 6) Se sugiere el uso de resoluciones nativas provistas por las cámaras en vez del uso de un proceso de disminución de resolución hecho de manera digital, debido a que no se aprovecha la máxima calidad ofrecida por los equipos de adquisición.

BIBLIOGRAFÍA

- [1] Chan, M., Delmas, P., Gimel'farb, G. and Leclercq, P., "Comparative Study of 3D Face Acquisition Techniques", Conference in Advances in Image Processing, <http://dl.acm.org/citation.cfm?id=2094348>, 2005
- [2] Rocchini, C., Cignoni, P., Montani, C., Pingi, P. and Scopigno, R. , "A Low Cost 3D Scanner Based on Structured Light", EUROGRAPHICS, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.5393>, 2001
- [3] Robinson, A., Alboul, L. and Rodrigues, M., "Methods for Indexing Stripes in Uncoded Structured Light Scanning Systems", Journal of WSCG, http://www3.shu.ac.uk/c3ri/gmpr/assets/pdf/gmpr-wscg_paper.pdf, 2003
- [4] Redert, A., Reinders, M., Van Den Eelaar, I., and Hendriks, E., "Extraction of Semantic 3D Models of Human Faces from Stereoscopic Image Sequences", <http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.36.2311&type=cc>, 1998.
- [5] Georgopoulos, A., Ioannidis, Ch., and Valanis ,A., "Assesing the performance of a structured light scanner", International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII, <http://www.isprs.org/proceedings/XXXVIII/part5/papers/177.pdf>, 2010
- [6] Bogusław, C. and Siebert, J.P., *An Introduction to 3D Computer Vision Techniques and Algorithms*, J. Wiley & Sons, 2009
- [7] Wright, R., Lipchak, B. and Haemel, N., *OpenGL Superbible: Comprehensive Tutorial and Reference*, Upper Saddle River, NJ: Addison-Wesley, 2007
- [8] Fechteler, P., Eisert, P., and Rurainsky, J., "Fast and High Resolution 3D Face Scanning", ICIP 2007, IEEE International Conference, www.informatik.hu-berlin.de/forschung/gebiete/viscom/papers/icip07b.pdf, 2007
- [9] Chen, C., Hung, Y., Chiang, Chiann, C., Wu, J., "Range data acquisition using color structured lightning and stereo vision", Image and Vision Computing, www.iis.sinica.edu.tw/papers/song/794-F.pdf, 1997

- [10] Chen, J., Mouaddibà, E., and Salvi, J., "Recent Progress in Coded Structured Light as a Technique to solve the Correspondence Problem: A Survey". *Pattern Recognition*, Vol. 31, No. 7, pp. 963-982, http://classes.soe.ucsc.edu/cmcs290b/Fall05/readings/Battle_PatRec98_RecentProgressCodedLight.pdf, 1998
- [11] University of W. Australia, "Stereo and Structured Lightning", Computer Vision CITS4240, School of Computer Science & Software Engineering, <http://undergraduate.csse.uwa.edu.au/units/CITS4240/Lectures/stereo.pdf>
- [12] Mattocchia, S., "Stereo Vision: Algorithms and Applications", University of Bologna, <http://www.vision.deis.unibo.it/smatt/Seminars/StereoVision.pdf>
- [13] Scharstein, D., and Szeliski, R., "High-Accuracy Stereo Depth Maps Using Structured Light", *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 195–202, <http://community.middlebury.edu/~schar/papers/structlight/structlight.pdf>, 2003
- [14] Zhang, S. and Yau, S. "High-speed Three-dimensional Shape Measurement System Using a Modified Two-plus-one Phase-shifting Algorithm", *Optical Engineering*, <http://structured-light.googlecode.com/svn-history/r41/trunk/zhang-two-plus-one.pdf>, 2007
- [15] Zhang, S. and Huang, P., "Fast three-step phase-shifting algorithm". *Applied Optics*, <http://www.opticsinfobase.org/ao/abstract.cfm?uri=ao-45-21-5086>, 2006.
- [16] Bradski, G. and Kaehler, A., "*Learning OpenCV: Computer Vision with the OpenCV Library*". Sebastopol, CA: O'Reilly, 2008.
- [17] Wikimedia Foundation, "Point Cloud", Wikipedia, http://en.wikipedia.org/wiki/Point_cloud, 2012
- [18] Gerrits, D., Gabriëls, R. and Kooijmans, P., "A Survey of Mesh Generation Techniques", *Advanced Algorithms 2005/2006*, Technische Universiteit Eindhoven, Department of Mathematics & Computer Science, <http://www.dirkgerrits.com/wp-content/uploads/mesh-generation-survey.pdf>, 2006
- [19] Bern, M. and Plassmann, P., "*Mesh Generation*", *Xerox Palo Alto Research Center and Department of Computer Science and Engineering at*

The Pennsylvania State University,
<http://www.cs.berkeley.edu/~jrs/meshpapers/BernPlassmann.pdf>, 2000

[20] Carey, G.F. and Oden, J.T., "Finite Elements: Computational Aspects", Prentice-Hall, 1984

[21] P.L. George, F. Hetch, and E. Saltel, "Fully automatic mesh generator for 3D domains of any shape", Impact of Com. in Sci. and Eng, 1990

[22] Bern M., Edelsbrunner H., Eppstein D., Mitchell S. and Tan, T.S., "Mesh generation and optimal triangulation", Disc. And Comp. Geometry. 2nd Edition. World Scientific, Singapore, 1995

[23] Shewchuk, Jonathan Richard, "Triangle: A two-dimensional quality mesh generator and Delaunay triangulator", Computer Science Division, University of California at Berkeley,
<http://www.cs.cmu.edu/%7Equake/triangle.html>

[24] *CGAL Project*, "Mesh Generation", *CGAL User and Reference Manual*,
www.cgal.org/Manual/latest/doc_html/cgal_manual/packages.html