

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**SOFTWARE PARA DETECTAR Y CORREGIR ERRORES EN
EL MAPA DIFUSO FOTOGRAMÉTRICO, OBTENIDO A
TRAVÉS DE CÁMARAS DIGITALES DE USO COMÚN**

para optar el Título de Licenciatura en Informática

Mención: Ingeniería de Sistemas Informáticos

POR: IVAN MAMANI FUENTES

TUTOR METODOLÓGICO: M.SC. ALDO RAMIRO VALDEZ ALVARADO

ASESOR: LIC. JHONNY ROBERTO FELIPEZ ANDRADE

LA PAZ – BOLIVIA
2020



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**

LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria

*A dios por ayudarme a cumplir un objetivo más
en mi vida*

A toda mi familia por el apoyo incondicional

AGRADECIMIENTOS

A Dios por acompañarme en los momentos buenos y en los momentos más difíciles de mi vida y por ayudarme con mi salud y la de mi familia.

A toda mi familia, en especial a mi Madre por su constante apoyo, ayudándome a creer que todo se puede con la ayuda de Dios.

A mi asesor Lic. Jhonny Roberto Felipez Andrade por su predisposición e interés durante el desarrollo de la presente investigación, y durante las revisiones por las observaciones, consejos que me fueron de mucha ayuda.

A mi tutor M.Sc. Aldo Ramiro Valdez Alvarado por guiarme durante el desarrollo de la presente investigación, por sus valiosas sugerencias y consejos.

RESUMEN

La fotogrametría, en los últimos años ha provocado un gran impacto en el cine y los videojuegos, esto gracias al avance de la tecnología y la aparición de nuevas tecnologías que complementan a la fotogrametría, y haciendo el análisis del impacto de la fotogrametría en la actualidad y viendo el futuro del mismo, se puede decir que la fotogrametría es el futuro de los videojuegos.

Pero por falta de información, una guía exacta para aplicar la fotogrametría de forma correcta, eficiente, en los videojuegos, muchos desarrolladores independientes no pueden aplicar esta tecnología en sus proyectos, y además vale la pena aclarar que la falta de tecnologías que ayuden a incorporar la fotogrametría en los videojuegos es un obstáculo para muchos desarrolladores independientes, además es necesario resaltar que la mayoría de los desarrolladores independientes no cuentan con una inversión económica, lo que significa que esto les impide a acceder a tecnologías fotogramétricas.

La presente investigación tiene como objetivo, ayudar a los desarrolladores independientes a incorporar la fotogrametría en los videojuegos de forma eficiente, tomando en cuenta que en el proceso de fotogrametría se usan cámaras digitales de uso común, y que a causa de esto, la imagen resultante tiene áreas erróneas en esta imagen resultante, dicho de otra forma, se plantea desarrollar un software que pueda ayudar a incorporar la fotogrametría en los videojuegos, ayudando a mejorar la imagen resultante (mapa difuso fotogramétrico) de la fotogrametría, detectando y corrigiendo áreas erróneas en esta imagen.

El software prototipo que pueda ayudar a mejorar la imagen resultante de la fotogrametría, fue desarrollado bajo el lenguaje de programación Python, haciendo uso de la librería de visión artificial, la morfología de imágenes, la librería blend modes, y la tecnología qt designer.

Durante los análisis y prueba de los resultados obtenidos con la presente investigación, se concluyó que los objetivos, y la hipótesis se cumplen.

Palabras Clave: áreas erróneas, detección, corrección, morfología de imágenes.

ABSTRACT

Photogrammetry, in recent years has caused a great impact on film and video games, this thanks to the advancement of technology and the emergence of new technologies that complement photogrammetry, and making the analysis of the impact of photogrammetry today and seeing the future of it, you can say that photogrammetry is the future of video games.

But for lack of information, an exact guide to apply photogrammetry correctly, efficiently, in video games, many independent developers cannot apply this technology in their projects, and it is also worth clarifying that the lack of technologies that help to incorporate photogrammetry in video games is an obstacle for many independent developers, it is also necessary to highlight that most independent developers do not have a monetary investment, which means that this prevents them from accessing photogrammetric technologies.

The purpose of this research is to help independent developers to incorporate photogrammetry into video games efficiently, taking into account that digital cameras are commonly used in the photogrammetry process, and because of this, the resulting image It has erroneous areas in this resulting image, in other words, it is proposed to develop software that can help incorporate photogrammetry in video games, helping to improve the resulting image (diffuse photogrammetric map) of photogrammetry, detecting and correcting erroneous areas in this picture.

The prototype software that can help improve the image resulting from photogrammetry was developed under the Python programming language, using the artificial vision library, image morphology, blend modes library, and qt designer technology.

During the analysis and testing of the results obtained with the present investigation, it was concluded that the objectives and the hypothesis are met.

Keywords: erroneous areas, detection, correction, morphology of images.

INDICE

CAPITULO I MARCO REFERENCIAL

1.3. Introducción	1
1.4. Antecedentes	2
1.5. Planteamiento de problema.....	4
1.5.1. Problema central	4
1.5.2. Problemas secundarios	5
1.6. Definición de objetivos.....	5
1.6.1. Objetivo general.....	5
1.6.2. Objetivos específicos	5
1.7. Hipótesis	5
1.7.1. Operacionalizacion de variables	6
1.8. Justificación	6
1.8.1. Justificación económica	6
1.8.2. Justificación social	6
1.8.3. Justificación científica.....	6
1.9. Alcances y limites	6
1.9.1. Alcances	6
1.9.2. Limites.....	7
1.10. Aportes.....	7
1.10.1. Practico	7
1.10.2. Teórico.....	7
1.11. Metodología	7

CAPITULO II MARCO TEÓRICO

2.1. Introducción	8
2.2. Metodología de desarrollo rad.....	8
2.1.1. Ventajas	9
2.1.2. Desventajas.....	10

2.3.La fotogrametría	10
2.2.1. Historia de la fotogrametría	11
2.2.2. Tipos de la fotogrametría.....	15
2.2.3. Método.....	15
2.4.La fotogrametría y su aplicación	16
2.5.La fotogrametría en los video juegos.....	19
2.6.Procesamiento de imágenes	25
2.7.Texturas.....	29
5.1.1. Tipos de texturas	30
2.8.Morfología de imágenes	31
2.9.Liberia de visión artificial	35
2.10. Qt creator	34
2.11. Qt creator en tecnologías 3d	36

CAPITULO III MARCO APLICATIVO

3.1.Introducción	38
3.2.Metodología de desarrollo rad.....	38
3.3.Faces de la metodología de desarrollo rad.....	38
3.3.1. Modelado de gestión.....	38
3.3.2. Modelado de datos	38
3.3.3. Modelado de procesos.....	41
3.3.3.1.Proceso de fotogrametría.....	41
3.3.3.2.Proceso de modelado 3d.....	41
3.3.3.3.Proceso de detección y corrección de errores	44
3.3.3.4.Proceso de pruebas y resultados	44
3.3.4. Generación de la aplicación.....	44
3.3.4.1.Diseño de interfaz gráfica.....	44
3.3.4.2.Compilación de la interfaz gráfica a lenguaje Python.....	46

3.3.4.3. Algoritmo para la detección de errores.....	47
3.3.4.4. Algoritmo para la corrección de errores.....	54
3.3.5. Pruebas y entrega	59
3.3.5.1. Pruebas.....	59
3.3.5.2. Resultados	62

CAPITULO IV PRUEBA DE HIPÓTESIS

4.1. Análisis de los resultados.....	66
4.2. Prueba de hipótesis	71

CAPITULO V CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones	73
5.2. Recomendaciones	73

BIBLIOGRAFÍA	74
---------------------------	----

ANEXOS	76
---------------------	----

• ANEXO A – TABLA T-STUDENT.....	77
----------------------------------	----

ÍNDICE DE FIGURAS

Figura 2.1: Fases de la metodología RAD.....	9
Figura 2.2: UAV civil para fotografía aérea y mapeo fotográfico	18
Figura 2.3: Fotogrametría en dos videojuegos de los dos deportes más populares	20
Figura 2.4: Escaneo facial.....	21
Figura 2.5: Escaneo facial y la captura de movimientos en los videojuegos	21
Figura 2.6: Fotogrametría y realismo en videojuegos, Assassins Creed.....	23
Figura 2.7: Fotogrametría y realismo en videojuegos, Star Wars BattleFront	23
Figura 2.8: La fotogrametría en Unreal Engine 4	24
Figura 2.9: Ejemplos de filtro sobre una imagen.....	25
Figura 2.10: Filtros en el dominio de la frecuencia	27
Figura 2.11: Ejemplo del método dilatación	31
Figura 2.12: Ejemplo del método erosión	32
Figura 2.13: Ejemplo del método apertura	34
Figura 3.1: Conjunto de datos para la fotogrametría.....	39
Figura 3.2: Datos básicos obtenidos de la fotogrametría	42
Figura 3.3: Resultado de la fotogrametría en Cryengine.....	43
Figura 3.4: Realismo con el sistema de vegetación de Cryengine.....	43
Figura 3.5: Software qt designer	45
Figura 3.6: Interfaz gráfica del software en su primera versión	45
Figura 3.7: Interfaz gráfica del software en su versión final	46
Figura 3.8: Área errónea en un mapa difuso fotogramétrico.....	47
Figura 3.9: Los casos más comunes entre áreas erróneas y no erróneas.....	48
Figura 3.10: Ejemplo de Umbralizacion de una imagen	51
Figura 3.11: Ejemplo del método morfológico Open	52
Figura 3.12: Ejemplo del método morfológico Close	52
Figura 3.13: Ejemplo del método morfológico Dilatacion.....	53
Figura 3.14: Ejemplo del método Inpainting telea	53

Figura 3.15: Ejemplo del método Hard light	56
Figura 3.16: Ejemplo del método Sreem.....	57
Figura 3.17: Software en su versión final.....	59
Figura 3.18: Resultado de la detección de errores	60
Figura 3.19: Resultado del algoritmo de reconstrucción Inpainting Telea	61
Figura 3.20: Resultado de la corrección de errores.....	62
Figura 3.21: El modelo 3d “hk9”, con el mapa difuso corregido	63
Figura 3.22: El modelo 3d “hk9”, con el mapa difuso corregido	63
Figura 3.23: El modelo 3d “hk1”, con el mapa difuso original	64
Figura 3.24: El modelo 3d “hk1”, con el mapa difuso corregido	65
Figura 4.1: El mapa difuso original y el resultado obtenido.....	67
Figura 4.2: El mapa difuso original y el resultado esperado	68
Figura 4.3: Región de aceptación de la hipótesis nula	72

ÍNDICE DE TABLAS

tabla 3.1 Características mínimas de las imágenes a procesar.....	40
tabla 3.2 Análisis de los píxeles en el caso fácil.....	49
tabla 3.3 Análisis de los píxeles en el caso intermedio	49
tabla 3.4 Análisis de los píxeles en el caso complicado	50
tabla 4.1 Muestra para la prueba de la hipótesis.	70

CAPITULO I

MARCO INTRODUCTORIO

1.1.INTRODUCCIÓN

La fotogrametría es definida como una ciencia o arte de realizar mediciones en base a fotografías a fin de determinar características métricas y geométricas de los objetos fotografiados, como por ejemplo tamaño, forma y posición (Routin, 1978).

La fotogrametría fue evolucionando con el tiempo, y con la aparición de nuevas tecnologías de procesamiento de imágenes, la fotogrametría llegó al mundo de los videojuegos, y gracias a esta tecnología en los últimos años los videojuegos han evolucionado de forma muy acelerada.

Viendo la actualidad de los videojuegos, claramente la fotogrametría es el futuro de los videojuegos, ya que, con la constante mejora de tarjetas gráficas, la llegada PS 5, la nueva API Vulkan, y la llegada de la tecnología de trazado de rayos a los motores gráficos, la fotogrametría será el pilar central. por los resultados que ofrece.

Viendo los resultados de la fotogrametría en los videojuegos y el cine podemos reafirmar, que la fotogrametría es una de las herramientas más potentes en el desarrollo de videojuegos y además la mayoría de los desarrolladores buscan explotar esta tecnología al 100%.

En la comunidad de desarrolladores independientes de videojuegos, la mayoría no cuenta con lo necesario para aplicar la fotogrametría en sus proyectos de forma eficiente, muchos de estos desarrolladores son limitados por la falta de inversión económica, y a causa de esto no cuentan con tecnologías que complementan a la fotogrametría para su eficiente aplicación.

En el presente trabajo se investigará las principales causas que hacen que un software de escaneo 3d no pueda procesar correctamente imágenes, imágenes obtenidas a través de cámaras digitales de uso común, y posteriormente desarrollar un software para detectar y corregir errores en las texturas resultantes del escaneo 3d, de tal forma que todos los

resultados del escaneo 3d se utilicen sin tener ningún problema con las diferentes técnicas de iluminación en un motor gráfico.

1.2.ANTECEDENTES

Al principio la fotogrametría fue utilizada como una herramienta a través de la cual se podía documentar grandes espacios de tierra desde el aire, posteriormente se adentró en grandes espacios pero desde tierra, en los últimos años se ha podido ver una gran participación en diferentes ramas entre las cuales podemos ver a la arqueología, las posibilidades que la fotogrametría brinda a la hora de documentar piezas y espacios arqueológicos no han sido vistas de forma indiferente por los profesionales de esta disciplina, fruto de esta extensión por distintas ramas, podemos hoy día ver cómo se ha difundido hasta el cine y los videojuegos (Luengo, 2016).

Desde sus inicios los desarrolladores han querido conseguir videojuegos cada vez más reales. Dentro de la fotogrametría y el realismo en videojuegos, uno de los ejemplos es el Doom, en 1993 Id Software creó un juego, que se podía jugar a través de disquete y cargando a través de un PC con MS-DOS, en el cual un marine de los EE. UU. inspecciona una de las lunas de marte y en un fallo de tele transportación quedan abiertas las puertas del infierno.

Veintitrés años más tarde, en 2016 salió Doom 4 para PC, Xbox One y PS4 con una realidad sorprendente. Un cambio radical en busca de escenario y detalles cada vez más reales.

Lo mismo ha ocurrido en muchos videojuegos con nombres conocidos y con muchos años en la industria. Metal Gear, Final Fantasy, Resident Evil, GTA desde sus respectivas creaciones para PC, han pasado muchos años y muchos avances tecnológicos, incluyendo la Fotogrametría.

La realidad se ha querido conseguir en diferentes estilos de juego. En especial, se ha notado en gran medida el cambio hacia la búsqueda de lo real en juegos de coches (Gran Turismo), Fútbol (FIFA), Baloncesto (NBALive2k17), etc. Auténticos juegos de simulación en los que el realismo juega un papel fundamental y es lo que espera año tras año el usuario, además de la constante mejora de tarjetas gráficas.

A lo largo de la historia, una de las cosas más difíciles era conseguir realidad en las caras humanas, pero a través de la fotogrametría se ha conseguido de una manera increíble.

Uno de los ejemplos más recientes es el Resident Evil VII, una creación de Capcom. En Capcom, han invertido en Investigación y Desarrollo para desarrollar un método avanzado de fotogrametría. Con esta técnica se generan texturas y modelados de la más alta calidad a través de fotografías de objetos y personas, las cuales se toman desde diferentes perspectivas, gracias a esta tecnología, su último videojuego de la saga de zombis más exitosa, en el Resident Evil VII, se han podido crear protagonistas y criaturas con una realidad que da miedo. Junto a la fotogrametría, se ha utilizado el modelado 3D y actores con maquillaje para poder crear monstruos y zombis con deformidades de gran realismo, además, Capcom, ha creado un estudio completo con más de 100 cámaras para seguir trabajando con la técnica de fotogrametría y el escaneo en 3D para futuros proyectos, al cual han llamado RE Engine.

Desde los años 90, en las industrias del cine y publicidad, se ha utilizado la exploración facial y corporal 3D para la creación de dobles digitales en actores. En la industria del videojuego no hacía falta, ya que la tecnología existente en baja resolución y el hardware no permitían crear formas tan realistas (GlobalMediterranea, 2017).

Quiixel Mixer es un software para mezclar texturas fotogramétricas, pero también es muy útil para la rápida corrección de texturas, pero está limitada con las texturas de modelos 3d, ya que lo dificulta el mapa UV del modelo 3d.

La herramienta De-Light es una herramienta para eliminar información de la iluminación en una textura fotogramétrica, la herramienta de De-light de Unity3d requiere ciertos mapas de textura para funcionar, entre las cuales están el: Mapa Difuso, el Mapa de Oclusión Ambiental, Mapa Normal del Espacio de Objetos, Bent Normal Map. La mayoría del software de fotogrametría como, PhotoScan, RealityCapture, ReCap Photo, 3DF Zephyr, entre otros, le permitirá exportar un mapa difuso y posiblemente otros mapas.

Houdini fx es una de las tecnologías más potentes y más utilizadas por grandes compañías para la creación de contenido para videojuegos y películas, Delight es uno de los plugins de Houdini Fx, la cual sirve para eliminar las sombras en un modelo 3d escaneado, basándose

en los colores de los vértices del modelo 3d, para colorear los vértices de un modelo escaneado de forma automática, Reality Capture es uno de los softwares que tienen la posibilidad de obtener colores de los vértices a partir de las imágenes en proceso de escaneo.

1.3.PLANTEAMIENTO DEL PROBLEMA

1.3.1. PROBLEMA CENTRAL

Durante los últimos años, con la aparición de nuevas tecnologías de fotogrametría, los videojuegos han evolucionado de forma muy acelerada, cada vez se parecen más al mundo real, tanto en los gráficos, inteligencia artificial, animaciones, entre otros.

Durante las conferencias GDC (2016), Andrew Hamilton Artista principal del medio ambiente y Kenneth Brown Director técnico de arte de la empresa EA Digital Illusions CE (EA DICE) mostraron el uso de las tecnologías de fotogrametría en su video juego Star Wars Battlefront, desde los objetos más pequeños o simples hasta muy complejos, llegando a escanear mundos completos, y como resultado obteniendo escenarios realistas, de mucha calidad gráfica a 60 fps.

Aplicar la fotogrametría en los videojuegos puede acelerar de forma notable el desarrollo de un proyecto y se puede obtener resultados de mucha calidad y realismo, pero aprovechar esta tecnología al máximo puede llegar a ser muy complejo, por ejemplo, en escaneo 3d la iluminación y el brillo de un objeto juegan un papel de gran importancia y no siempre se obtienen los resultados esperados, además se debe tener a la disposición una cámara digital de muy buena calidad, calibrador de colores, aparatos de iluminación, softwares profesionales, entre otros.

Cuando se hace el uso de cámaras digitales de uso común en la fotogrametría, los resultados obtenidos como ser el mapa difuso, el modelo 3d, el mapa normal, no son adecuados para ser utilizados de forma directa en un motor gráfico, ya que normalmente tienen errores notorios, sobre todo en el mapa difuso, y esto se debe a que solo se utilizó un celular, para realizar la fotografías, por ende, se plantea:

¿Cómo aprovechar al máximo todas las texturas fotogramétricas generados a partir de fotografías tomadas con una cámara digital de uso común?

1.3.2. PROBLEMAS SECUNDARIOS

- Una Clima inadecuada al momento de fotografiar un objeto, provoca que los resultados finales aplicados al modelo 3d tengan problemas con las diferentes técnicas de iluminación en un motor gráfico.
- Las malas características del objeto al momento de fotografiarlo, provoca que un software de escaneo 3d no pueda reconocer algunas áreas de forma correcta, a causa de esto el mapa difuso pierde su calidad.
- La falta de herramientas que puedan ayudar a mejorar texturas fotogramétricas, obtenidas a través de cámaras digitales de uso común, provoca que algunos de estas texturas no puedan ser utilizados en un motor gráfico.

1.4.DEFINICIÓN DE OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un software para detectar y corregir errores en el mapa difuso fotogramétrico, obtenido a través de cámaras digitales de uso común.

1.4.2. OBJETIVOS ESPECÍFICOS

- Fotografiar todo objeto en clima adecuado, de tal forma que las imágenes resultantes del proceso de fotogrametría tengan la menor cantidad de errores posibles, y para que estos puedan ser utilizados de forma eficiente.
- Antes de fotografiar un objeto verificar que los objetos a su alrededor no lleguen a afectarlo, así para tener la menor cantidad de errores y además con el fin de evitar que aparezcan áreas anormales (un área se considera anormal, cuando este tiene un color muy diferente con respecto a la imagen que pertenece).
- Desarrollar un software que pueda ser capaz de detectar todas las áreas erróneas (un área se considera errónea, cuando las diferencias entre los pixeles en esta área son muy bajos) en el mapa difuso fotogramétrico y corregirlos.

1.5.HIPÓTESIS

La librería de visión artificial, la morfología de imágenes, y la librería blend modes, permitirá al software detectar y corregir errores en el mapa difuso fotogramétrico con un acertamiento mayor al 85%.

1.5.1. OPERACIONALIZACION DE VARIABLES

Variables independientes:

- Software para detectar y corregir errores en el mapa difuso fotogramétrico.

Variables dependientes:

- Morfología de imágenes
- Librería de visión artificial
- Librería blend modes

1.6. JUSTIFICACIÓN

1.6.1. JUSTIFICACIÓN ECONÓMICA

La presente investigación ayudará a los desarrolladores independientes de videojuegos a ahorrar dinero en comprar equipos necesarios para aplicar la fotogrametría de forma eficiente, ya estos equipos son costosos, además muchos de estos no se encuentran disponibles en todos los países, dicho de otra forma, en la presente investigación solo se utilizará cámaras digitales de uso común, junto a un software de escaneo 3d.

1.6.2. JUSTIFICACIÓN SOCIAL

La presente investigación ayudara a los desarrolladores independientes de videojuegos a incorporar las tecnologías de fotogrametría en sus proyectos de forma eficiente, aprovechando la mayor cantidad de texturas que se pueden obtener de un escaneo 3d.

1.6.3. JUSTIFICACIÓN CIENTÍFICA

El software a desarrollarse durante la presente investigación estará implementado en el lenguaje de programación Python, y qt designer, y gracias a que muchos softwares 3d tienen soporte para Python, en el futuro esta investigación podrá formar parte de ellos.

1.7.ALCANCES Y LIMITES

1.7.1. ALCANCES

- El software a desarrollarse corregirá errores en un mapa difuso fotogramétrico de forma automática.

- El software a desarrollarse ayudara a que un modelo 3d escaneado sea utilizado de forma adecuado en un motor gráfico.

1.7.2. LIMITES

- El software a desarrollarse no podrá corregir errores en el mapa normal correspondiente a un mapa difuso fotogramétrico.
- El software a desarrollarse no podrá corregir errores de forma adecuada en el mapa difuso fotogramétrico si este tiene un área errónea con más del 50% de pixeles.

1.8.APORTES

1.8.1. PRACTICO

El aporte principal de la presente investigación y desarrollo es el uso de las cámaras digitales de uso común para tomar fotografías a objetos del mundo real para luego procesarlos en un software de fotogrametría y usar estos resultados en proyectos de videojuego de forma eficiente.

1.8.2. TEÓRICO

El aporte principal es el análisis, detección y corrección de anomalías en un matriz de pixeles, utilizando métodos de la morfología de imágenes, la librería de visión artificial, y otras librerías disponibles para el lenguaje de programación Python.

1.9.METODOLOGÍA

La presente investigación se procederá con la metodología de enfoque cuantitativo descriptiva, dado que esta metodóloga permitirá el análisis y pruebas de los factores relevantes a utilizar cámaras digitales de uso común en la fotogrametría.

Par el desarrollo del software se empleará la metodología de desarrollo rápido de aplicaciones (RAD), que implica el desarrollo iterativo y la construcción de prototipos, esta metodología permitirá a la presente investigación una correcta manipulación de las imágenes obtenidas a través de cámaras digitales de uso común, su posterior procesamiento, pruebas con los resultados finales en el motor gráfico en tiempo real y de esta forma poder verificar los alcances y límites.

CAPITULO II

MARCO TEÓRICO

2.1. INTRODUCCIÓN

En el presente capítulo se explica la definición, funcionamiento y aplicaciones de la fotogrametría en la actualidad, además de la morfología de imágenes y la librería de visión artificial, y la tecnología qt.

2.2.METODOLOGÍA DE DESARROLLO RÁPIDO DE APLICACIONES RAD

El desarrollo rápido de aplicaciones o RAD (acrónimo en inglés de rapid application development) es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución (Mauricio, 2017).

El Modelo RAD comprende las siguientes etapas:

- **Modelado de gestión.** Este modelo se basa en dar respuesta a las siguientes preguntas: – ¿Qué información conduce el proceso de gestión? – ¿Qué información genera? – ¿A dónde va la información? – ¿Quién la procesa?
- **Modelado de datos.** En este modelo se definen los almacenes de datos y cómo se relacionan los almacenes entre sí.
- **Modelado del proceso.** Se utiliza para añadir, modificar, suprimir o recuperar un objeto de datos.
- **Generación de aplicaciones.** Para esto se utiliza una herramienta de cuarta (o quinta) generación que permite crear el software y facilitar la construcción del programa.
- **Pruebas y entrega.** El proceso de desarrollo finaliza realizando pruebas de calidad del software diseñado con la herramienta RAD, posteriormente se realiza la implementación de la aplicación.

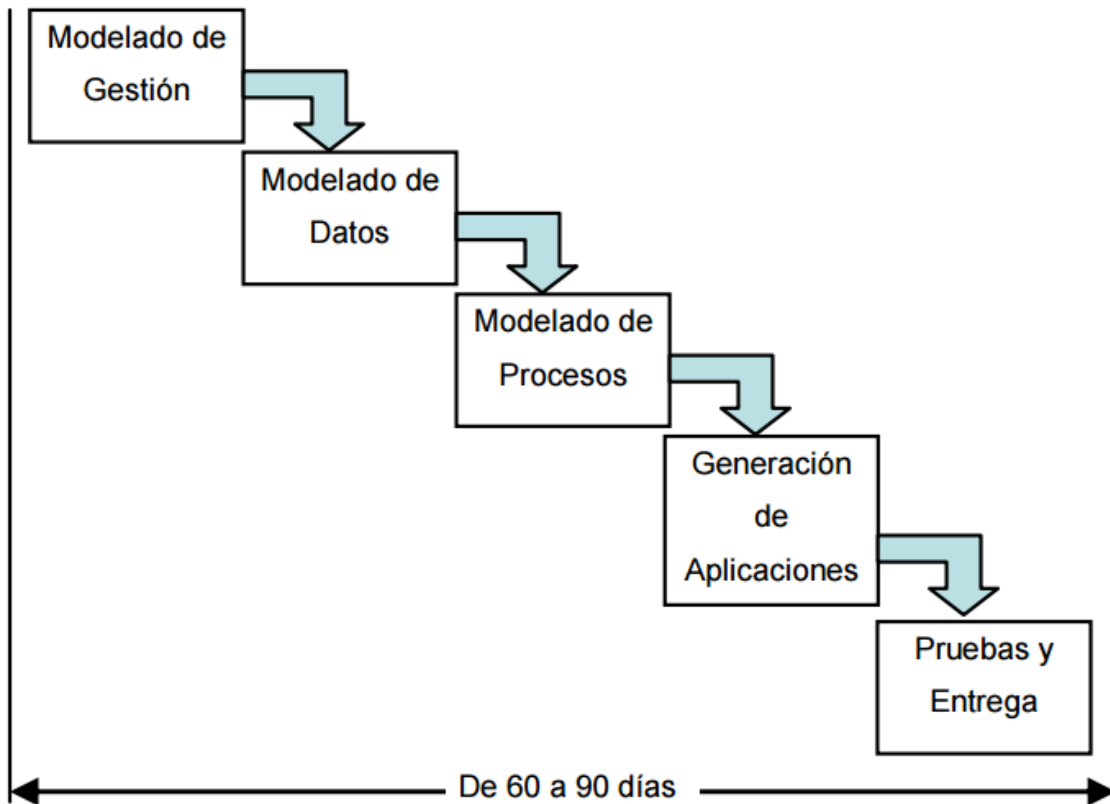


Figura 2.1: Etapas de la metodología rad.

Fuente: Mauricio, 2017

2.2.1. VENTAJAS

- Comprar puede ahorrar dinero en comparación con construir.
- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Visibilidad temprana.
- Mayor flexibilidad.
- Menor codificación manual.
- Mayor involucramiento de los usuarios.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.

- Interfaz gráfica estándar.

2.2.2. DESVENTAJAS

- Comprar puede ser más caro que construir.
- Costo de herramientas integradas y equipo necesario.
- Progreso más difícil de medir.
- Menos eficiente.
- Menor precisión científica.
- Riesgo de revertirse a las prácticas sin control de antaño.
- Más fallas (por síndrome de “codificar a lo bestia”).
- Prototipos pueden no escalar, un problema mayúsculo.
- Funciones reducidas (por “timeboxing”).
- Dependencia en componentes de terceros: funcionalidad de más o de menos, problemas legales.

2.3. FOTOGRAMETRÍA

La fotogrametría es una técnica que tiene como objetivo el estudio y la definición de forma precisa de la forma, las dimensiones y/o posición en el espacio de un objeto, a través de los datos obtenidos de una o varias fotografías (GlobalMediterráneaGeomática, 2020).

Según su etimología, la palabra fotogrametría significa la medida de lo escrito con luz, deriva del vocablo “fotograma” (de “phos”, “photós”, luz, y “gramma”, trazado, dibujo) y “metrón”, medir.

• Ventajas

Son muchas las ventajas frente a la topografía clásica. Es la agilidad y rapidez en la toma de datos. También se puede acceder a terrenos de difícil acceso y más en la actualidad, en la que el uso de drones en la fotogrametría permite llegar a todo tipo de lugares. Además, no le influyen las condiciones climatológicas. En las fotografías obtenidas queda registrados todos los detalles del terreno de forma continua, mientras que en la topografía clásica solo se obtienen puntos medidos en las coordenadas (Global Mediterránea Geomatica,2020).

2.3.1. HISTORIA DE LA FOTOGRAMETRÍA

- **Fotogrametría terrestre**

En 1725 se tiene la primera noticia del empleo de perspectivas con fines cartográficos. M.A. Capeller cartografía el monte Pilatus con un procedimiento similar al método de las intersecciones de la topografía. Las perspectivas habían sido construidas con un procedimiento ideado por el pintor alemán Alberto Durero en el siglo XVI. Esas perspectivas presentaban una serie de problemas para usarlas en cartografía. Capeller no obtuvo las precisiones adecuadas y su idea no se generalizó.

En 1839 François Arago inventa la fotografía que en 1859 fue utilizada por el coronel francés Aimé Laussedat para la confección de planos topográficos, que ya en 1846 había empezado a emplear perspectivas dibujadas por medio de una cámara clara o cámara lúcida para el mismo propósito. En 1852 Laussedat comienza a trabajar para reemplazar la cámara clara por la cámara oscura hasta construir en 1859 un prototipo de lo que después sería un fototeodolito. El procedimiento empleado por Laussedat era lo que se conoce como fotogrametría de intersección.

En 1858 Albrecht Meydenbauer utiliza el procedimiento de intersecciones a partir de fotografías para el levantamiento de obras arquitectónicas y lo denomina fotogrametría. Los procesos de Laussedat y Meydenbauer tenían problemas en cuanto a la identificación de un mismo punto en dos fotografías.

En 1901 Carl Pulfrich elimina el problema de la identificación de puntos homólogos con el nacimiento del estereocomparador de Pulfrich, que también permite la medición de coordenadas y paralajes con alta precisión.

En 1914 aparece el estereoautógrafo de von Orel, construido sobre la base de estereocomparador de Pulfrich al que se acopló un dispositivo de regletas mecánicas que transmiten los valores de las coordenadas de los puntos de los fotogramas, permitiendo el trazado continuo de los rasgos cartográficos.

En 1920, en Argentina, se construye el estereógrafo, perfeccionado en 1926. Distintos modelos de este aparato han prestado servicio durante muchos años en el Instituto

Geográfico Nacional y en la empresa privada “Instituto Fototopográfico Argentino”, propiedad de los inventores.

- **Fotogrametría aérea**

En 1897, el austríaco T. Scheimpflug comenzó sus importantes trabajos sobre rectificación, técnica que adquiriría gran difusión y, en 1898 sentó las bases de la idea de la proyección doble.

En 1899 S. Finsterwalder propone resolver el problema de la orientación en tres pasos: orientación interior, orientación relativa y orientación absoluta.

El 1915 Gasser, con las ideas de Scheimpflug, construyó su Proyector Doble, primer aparato apto para fotografías aéreas con buen funcionamiento. La observación de las imágenes proyectadas en la mesa medidora se hacía por el método anaglífico.

En 1920 Nistri construyó su Fotocartógrafo, también con el sistema anaglífico.

En 1921 W. Bauersfeld propone una elegante solución para trabajar con el principio Porro-Kope, con la cual Carl Zeiss construye el estereoplanógrafo. A partir de 1921, Ermenegildo Santoni en Italia retomó la proyección mecánica con su autoreductor, al que siguieron toda una serie de diseños.

En 1924 Otto von Gruber resuelve el problema de las orientaciones de manera más sistemática y completa. Desde 1936, Wild, es Suiza produjo solamente instrumentos de proyección mecánica y en 1960 Zeiss Oberkochen también cambió a la proyección mecánica con instrumentos como el Planimap y el Planicart.

Gasser y von Gruber también propusieron y diseñaron procedimientos para la concatenación de modelos dando nacimiento a la aerotriangulación.

La Fotogrametría Aérea adquiere con los aparatos restituidores analógicos una rápida y formidable difusión. El camino abierto por el estereoautógrafo de von Orel en la Fotogrametría Terrestre se ensancha considerablemente y la producción de mapas y planos de todo tipo se incrementa drásticamente.

Con el advenimiento de la computación, los cálculos pudieron hacerse a altas velocidades. Durante muchos años coexistieron procedimientos analógicos y analíticos. En los años '60 el Restituidor Analítico, creado por el finlandés Uki Helava, tenía un alto coste. Sólo en los años '80 su precio comenzó a parecerse al de los analógicos. El restituidor analítico trajo una sensible mejora en la precisión y, además, posibilitó el empleo de cualquier tipo de fotografía o aún de imágenes no fotográficas. Otras ventajas importantes fueron la facilidad con la que se podían corregir errores sistemáticos (distorsión de la lente, variaciones dimensionales de la película y efectos de refracción atmosférica y curvatura terrestre) y el empleo de altas redundancias con un tratamiento por mínimos cuadrados. Sin embargo, los grandes beneficios de la Fotogrametría Analítica no estuvieron en el restituidor analítico, sino en el cambio en las técnicas de Aerotriangulación.

La aerotriangulación analógica concatenaba los modelos de una pasada en los aparatos analógicos. La muy desfavorable propagación de errores tenía una cierta ventaja: el efecto de la doble sumatoria en los errores de transferencia hacía que, aun considerando que los errores fueran accidentales, las deformaciones finales de la pasada tenían una apariencia sistemática, pudiendo evaluarse mediante el empleo de puntos de control terrestre al comienzo, en el medio y al final de cada pasada. La precisión de esos procedimientos era muy limitada porque no respondían a una verdadera compensación por mínimos cuadrados.

El motor del cambio de la evolución de aerotriangulación, en este caso la computadora, era algo que evolucionaba día a día. Soluciones que no requerían computadoras onerosas fueron empleadas en la década del '70 y aún en la del '80, a pesar de que no brindaban una solución rigurosa de acuerdo a mínimos cuadrados. En esto tuvo mucho que ver la generosidad de G.H.Schut, que puso sus programas gratuitamente a disposición de la comunidad fotogramétrica internacional. A mediados de la década de los '80, y con el advenimiento de las computadoras personales, los programas de compensación en bloque por mínimos cuadrados adquirieron una importante difusión. Su precio había descendido considerablemente y pasaron a formar parte del software opcional provisto al adquirir un restituidor analítico. Los desarrollos de software para la compensación de la Triangulación Aérea marcan todo un hito en la historia del diseño y elaboración de técnicas fotogramétricas.

En este tiempo se asiste a la transición que lleva de los procedimientos analíticos a los digitales. Los procedimientos digitales eran de uso corriente en la Teledetección desde el '70, pero la Fotogrametría se había mantenido casi impermeable a ellos hasta bien entrada la década del '80. La barrera existente entre los procedimientos digitales y la fotogrametría comenzó a perforarse a partir de los '90. En los últimos años se ha conseguido la correspondencia de imágenes con precisiones de subpixel.

Ante la dificultad de montar cámaras digitales en aeronaves, la Fotogrametría ha venido empleando hasta ahora un procedimiento híbrido: obtención de fotografías con la cámara analógica tradicional y posterior digitalización empleando escáneres de alta precisión. Sin embargo, en estos últimos años se han producido grandes avances en la construcción de tales cámaras digitales con una calidad de imagen y precisión similar a la de las cámaras aéreas tradicionales. Los primeros modelos de esas cámaras ya están en el mercado. En cuanto al tratamiento fotogramétrico de imágenes satelitales, se dispone desde hace varios años de imágenes con tamaño de píxel compatible con las precisiones cartográficas de las pequeñas escalas.

- **Fotogrametría de largo alcance**

El objetivo de la fotogrametría de largo alcance es el conocimiento de las dimensiones y la posición de objetos en el espacio, a través de la medida o medidas realizadas a partir de la intersección de dos o más fotografías, o de una fotografía y el modelo digital del terreno correspondiente al lugar representado, el cual ha de ser realizado anteriormente por intersección de dos o más fotografías.

Esta técnica es básica para la elaboración de toda la cartografía, ya sea topográfica, temática, catastral, entre otros.

Puede ayudarse de información espectral y radiométrica de una imagen digital apoyada en la teledetección.

La fotogrametría puede ser terrestre o aérea dependiendo desde donde son obtenidas las imágenes.

2.3.2. Tipos de fotogrametría

Existen varias formas de hacer fotogrametría:

- **Fotogrametría analógica:** Son los modelos matemáticos utilizados. Evidentemente, fue la primera parte de la fotogrametría en desarrollarse.
- **Fotogrametría analítica:** Se encarga de aplicar los modelos matemáticos a objetos físicos. Fue la segunda parte en desarrollarse.
- **Fotogrametría digital:** Con la aparición de los ordenadores, se sustituye la imagen analógica por la imagen digital, del mismo modo que se empiezan a utilizar programas informáticos. En la actualidad la fotogrametría digital convive con la analítica.
- **Fotogrametría aérea:** Es cuando las estaciones se encuentran en el aire. Se aplica para la elaboración de planos y/o mapas para el desarrollo de proyectos de ingeniería.
- **Fotogrametría terrestre:** En este caso las estaciones se encuentran a nivel del suelo.

2.3.3. MÉTODO

El método de reconstrucción de objetos o terreno (cartografía) mediante fotogrametría:

- **Fotografiar los objetos:** Será necesario una previa Planificación del vuelo y de las tomas de fotografías (se hace en la fase de Proyecto de vuelo), tras la planificación se procede a la Obtención de imágenes (Vuelo), y a un posterior Procesado.
- **Orientación de las imágenes:** Colocación de los fotogramas en la posición adecuada con sus marcas fiducias (orientación interna); Colocar los fotogramas en la misma posición que ocupaban entre ellos en el momento de las tomas (orientación relativa);
- **Formación del modelo por restitución** para después aplicarle giros, una traslación y un factor de escala (orientación absoluta) para tener el modelo (objeto) en coordenadas terreno. Incluye también el escalado del objeto para obtener y realizar medidas en las magnitudes reales.

- Formación del modelo por rectificación, consistente en, una vez aplicados la orientación tanto interna como externa del haz de luz, hallar la intersección entre dicho haz orientado y el modelo digital del terreno correspondiente al espacio que se quiere determinar.

2.4.FOTOGRAMETRÍA Y SU APLICACIÓN

Sus aplicaciones son numerosas, es una técnica ampliamente utilizada en muchas disciplinas como: Agronomía, Cartografía, Ortofotografía, Arquitectura, Planeamiento y ordenación del territorio, Medio ambiente, Arqueología, Control de estructuras, Mediciones, Topografía, Biomecánica en diversos campos como la Medicina, Ergonomía, Deporte, Investigación policial (reconstrucción de accidentes), Zoología, Ortofoto, Aplicaciones arquitectónicas, Arqueología, Control de deformaciones, Evaluación de vegetación e impacto ambiental, Cine, Videojuegos, entre otras.

La fotogrametría se utiliza en campos como el mapeo topográfico , la arquitectura , la ingeniería , la fabricación , el control de calidad , la investigación policial , el patrimonio cultural y la geología . Los arqueólogos lo usan para producir rápidamente planos de sitios grandes o complejos, y los meteorólogos lo usan para determinar la velocidad del viento de los tornados cuando no se pueden obtener datos climáticos objetivos.

También se utiliza para combinar la acción en vivo con imágenes generadas por computadora en la postproducción de películas ; Matrix es un buen ejemplo del uso de la fotogrametría en la película (los detalles se proporcionan en los extras del DVD). La fotogrametría se utiliza ampliamente para crear activos ambientales fotorrealistas para los juegos de vídeo, incluyendo la desaparición de Ethan Carter , así como EA DICE 's Star Wars Battlefront . El personaje principal del juego Hellblade: Senua's Sacrifice se derivó de modelos fotogramétricos de captura de movimiento tomados de la actriz Melina Juergens.

La fotogrametría también se emplea comúnmente en la ingeniería de colisiones, especialmente con automóviles. Cuando ocurre un litigio por accidentes y los ingenieros necesitan determinar la deformación exacta presente en el vehículo, es común que hayan pasado varios años y la única evidencia que queda son las fotografías tomadas por la policía en la escena del accidente. La fotogrametría se usa para determinar cuánto se deformó el

automóvil en cuestión, lo que se relaciona con la cantidad de energía requerida para producir esa deformación. La energía se puede usar para determinar información importante sobre el choque (como la velocidad en el momento del impacto).

Photomapping es el proceso de hacer un mapa con "mejoras cartográficas" que se han extraído de un fotomosaico que es "una imagen fotográfica compuesta del suelo" o más precisamente como un fotomosaico controlado donde "las fotografías individuales se rectifican para inclinación y llevado a una escala común (al menos en ciertos puntos de control) ".

La rectificación de las imágenes generalmente se logra "ajustando las imágenes proyectadas de cada fotografía a un conjunto de cuatro puntos de control cuyas posiciones se han derivado de un mapa existente o de mediciones en el suelo. Cuando estas fotografías rectificadas y escaladas se colocan en una cuadrícula de puntos de control, se puede lograr una buena correspondencia entre ellos a través de un ajuste y ajuste habilidosos y el uso de las áreas alrededor del punto principal donde los desplazamientos de alivio (que no se pueden eliminar) son mínimos ".

"Es bastante razonable concluir que alguna forma de fotomapa se convertirá en el mapa general estándar del futuro". continúa sugiriendo que "el mapeo fotográfico parecería ser la única forma de aprovechar razonablemente" las futuras fuentes de datos, como los aviones de gran altitud y las imágenes satelitales. Los mapas fotográficos aéreos de mayor resolución en GoogleEarth son imágenes de resolución espacial de aproximadamente 2,5 cm (0,98 pulgadas). La fotomapa de mayor resolución de imágenes ortográficas se realizó en Hungría en 2012 con una resolución espacial de 0,5 cm (0,20 pulgadas).

La industria del videojuego se mueve y se está adaptando a esta tecnología cada vez más realista. Un ejemplo de ello es el desarrollador de juegos The Astronauts, Esta empresa de Polonia ha sido siempre defensora de la fotogrametría, esta técnica 'permite tener activos verdaderamente fotorrealistas' según dicen.

Desde hace unos pocos años se ha capturado el movimiento a través de cámaras que se colocaban en las caras y cuerpo de los actores. Ahora junto a la fotogrametría, títulos como Heavy Rain, LA Noire o Beyond: Two Souls, éste último con Williem Dafoe y Ellen

Page como protagonistas y de una caracterización excelente, han salido a la luz dejando boquiabiertos a muchos usuarios.



Figura 2.2: UAV civil para fotografía aérea y mapeo fotográfico

Fuente: Russell, 2019

La fotografía aérea se ha aplicado ampliamente para mapear restos de superficie y exposiciones de excavación en sitios arqueológicos. Las plataformas sugeridas para capturar estas fotografías incluyen: Globos de guerra de la Primera Guerra Mundial; globos meteorológicos de goma; cometas ; plataformas de madera, estructuras metálicas, construidas sobre una exposición de excavación; escaleras, solas y unidas con postes o tablas; escaleras de tres patas; postes simples y de múltiples secciones; bípodes; trípodés; tetrápodos, camiones cisterna aéreos ("recolectores de cerezas").

Se han utilizado fotografías digitales de mano cerca del nadir sobre la cabeza con sistemas de información geográfica (SIG) para registrar las exposiciones de excavación.

La fotogrametría se usa cada vez más en la arqueología marítima debido a la relativa facilidad de los sitios de mapeo en comparación con los métodos tradicionales, lo que permite la creación de mapas 3D que se pueden representar en realidad virtual.

2.5.FOTOGRAMETRÍA EN LOS VIDEOJUEGOS

En 2014, los desarrolladores de The Vanishing of Ethan Carter revelaron que las asombrosas configuraciones en el juego se crearon utilizando la fotogrametría, una técnica de modelado que hasta ese momento no se consideraba demasiado compleja por las limitaciones técnicas de los motores de juego. Poco después, EA DICE anunció que su título insignia de 2015, Star Wars Battlefront, adoptaría la fotogrametría ampliamente, no solo para capturar accesorios y disfraces clave, sino también para recrear los lugares clave de Star Wars.aficionados. Desde entonces, la industria de los juegos ha sido inundada con artículos, tutoriales y nuevos softwares dedicados al uso de la fotogrametría para crear activos de juegos en 3D, mostrando que la fotogrametría solucionaba muchos problemas en la actualidad sobre el desarrollo de videojuegos.

La creación 3D en tiempo real ha evolucionado de forma notable y sorprendente a lo largo de los años. El mundo tiene detalles que hacen muy complicada su representación como por ejemplo la luminosidad de los paisajes, a pesar de ello poco a poco se está consiguiendo con técnicas como la fotogrametría.

Desde sus inicios los desarrolladores han querido conseguir videojuegos cada vez más reales. Dentro de la fotogrametría y el realismo en videojuegos, uno de los ejemplos es el Doom, en 1993 id Software creó un juego, que se podía jugar a través de disquete y cargando a través de un PC con MS-DOS, en el cual un marine de los EE. UU. inspecciona una de las lunas de marte y en un fallo de teletransportación quedan abiertas las puertas del infierno.

Veintitrés años más tarde, en 2016 salió Doom 4 para PC, Xbox One y PS4 con una realidad sorprendente. Un cambio radical en busca de escenario y detalles cada vez más reales.

Lo mismo ha ocurrido en muchos videojuegos con nombres conocidos y con muchos años en la industria. Metal Gear, Final Fantasy, Resident Evil, GTA... desde sus respectivas creaciones para Nes o PC, han pasado muchos años y muchos avances tecnológicos, incluyendo la Fotogrametría

La realidad se ha querido conseguir en diferentes estilos de juego. En especial, se ha notado en gran medida el cambio hacia la búsqueda de lo real en juegos de coches (Gran Turismo), Fútbol (FIFA), Baloncesto (NBALive2k17), etc. Auténticos juegos de simulación en los que el realismo juega un papel fundamental y es lo que espera año tras año el usuario.

En el caso de los juegos de simulación con jugadores reales, utilizan una serie de cámaras calibradas de gama alta para capturar cada detalle de la persona instantánea. La fotogrametría se utiliza para reconstruir una versión digital de la persona. Esta técnica utiliza fotografías tomadas de todos los ángulos de un objeto como entrada.



Figura 2.3: Fotogrametría en dos videojuegos de los dos deportes más populares

Fuente: Elaboración propia

A lo largo de la historia, una de las cosas más difíciles era conseguir realidad en las caras humanas, pero a través de la fotogrametría se ha conseguido de una manera increíble.

La captura de movimiento (del inglés motion capture, o motion tracking, también abreviada mocap) es una técnica de grabación de movimiento, en general de actores y de animales vivos, y el traslado de dicho movimiento a un modelo digital, realizado en imágenes de computadora. Se basa en las técnicas de fotogrametría y se utiliza principalmente en la industria del cine de fantasía o de ciencia ficción, en la industria de los

videojuegos o también en los deportes, con fines médicos. En el contexto de la producción de una película, se refiere a la técnica de almacenar las acciones de actores humanos, y usar esa información para animar modelos digitales de personajes en animación 3D.



Figura 2.4: Escaneo Facial

Fuente: Brendan, 2017



Figura 2.5: Escaneo facial y la captura de movimientos en los videojuegos

Fuente: Elaboración propia

Uno de los ejemplos más recientes es el Resident Evil VII, una creación de Capcom. En Capcom, han invertido en Investigación y Desarrollo para desarrollar un método avanzado de fotogrametría. Con esta técnica se generan texturas y modelados de la más alta

calidad a través de fotografías de objetos y personas, las cuales se toman desde diferentes perspectivas.

Gracias a esta tecnología, su último videojuego de la saga de zombis más exitosa, en el Resident Evil VII, se han podido crear protagonistas y criaturas con una realidad que da miedo. Junto a la fotogrametría, se ha utilizado el modelado 3D y actores con maquillaje para poder crear monstruos y zombis con deformidades de gran realismo.

Además, Capcom, ha creado un estudio completo con más de 100 cámaras para seguir trabajando con la técnica de fotogrametría y el escaneo en 3D para futuros proyectos, al cual han llamado RE Engine.

Desde los años 90, en las industrias del cine y publicidad, se ha utilizado la exploración facial y corporal 3D para la creación de dobles digitales en actores. En la industria del videojuego no hacía falta, ya que la tecnología existente en baja resolución y el hardware no permitían crear formas tan realistas. Todo esto ha cambiado radicalmente en la actualidad.

Desde hace unos pocos años se ha capturado el movimiento a través de cámaras que se colocaban en las caras y cuerpo de los actores. Ahora junto a la fotogrametría, títulos como Heavy Rain, LA Noire o Beyond: Two Souls, éste último con Williem Dafoe y Ellen Page como protagonistas y de una caracterización excelente, han salido a la luz dejando boquiabiertos a muchos usuarios.

Por otro lado, compañías como Ubisoft, Bandai Namco o Activision entre otras, están utilizando la fotogrametría para la creación de los trailers de presentación de sus juegos o para la recreación de escenarios. Títulos como Assassin's Creed Origins o el último Call of Duty World War II, son ejemplos de ello.

Para el desarrollo del videojuego Assassin's Creed Unity por Ubisoft, el equipo encargado del apartado gráfico, tuvo que recopilar y estudiar gran cantidad de fotos y vídeos hiperrealistas de la catedral tanto por dentro como por fuera, desde la estructura más grande hasta el detalle de un ladrillo para compilarlo en un detallado mapa interactivo 3D de la catedral.



Figura 2.6: Fotogrametría y realismo en videojuegos, Assassin's Creed Origins

Fuente: Elaboración propia

Uno de los logros más importantes en el videojuego Assassin's Creed Origins es sin duda el apartado gráfico, además vale la pena aclarar que la iluminación, el reflejo del sol en el agua, la vegetación, son increíbles, pero además los personajes, las animaciones también están muy bien logrados.



Figura 2.7: Fotogrametría y realismo en videojuegos, Star Wars BattleFront

Fuente: Elaboración propia

A parte de los títulos destacados anteriormente, otro de los más importantes y sonados será el Star Wars Battlefront II. Los de la compañía de EA y DICE, se trasladaron a cada uno de los lugares en los cuales se rodaron las películas para retratarlos a la perfección.

La fotogrametría ha sido la herramienta principal para conseguir unos escenarios muy reales para crearle al jugador un ambiente Star Wars perfecto.

Según relatan los chicos de DICE, que fueron los encargados de llevar a cabo el trabajo fotogramétrico, se hizo un escaneo total de todo, es decir, desde los cascos de los Stormtrooper hasta los árboles gigantes de Redwood.

Muchas de las naves y armaduras se trataron con sumo cuidado y se pudieron representar sin problemas, como la de Boba Fett. La máscara de Darth Vader fue un gran problema por ser negra y brillante

Con esta técnica han conseguido unos niveles de realismo y ambientación del mundo Star Wars sin precedentes, capturando árboles, rocas y lados enteros de acantilados. Una gran obra maestra que llegará a finales de año.

Desde hace unos años, muchas empresas de creación de videojuegos están optando por utilizar la fotogrametría para la creación de paisajes, personajes y objetos cada vez más reales.

La industria del videojuego se mueve y se está adaptando a esta tecnología cada vez más realista. Un ejemplo de ello es el desarrollador de juegos The Astronauts, Esta empresa de Polonia ha sido siempre defensora de la fotogrametría, esta técnica ‘permite tener activos verdaderamente fotorrealistas’ según dicen.

Lo que defienden desde The Astronauts es que hay muchos detalles realistas que se deben tener en cuenta. ‘El polvo se acumulará en lugares específicos según el viento, las erosiones de las superficies se notarán debido a la lluvia o tráfico humano, etc. El mundo real está lleno de pequeños detalles. Es la falta de este detalle en el mundo virtual que rompe inmediatamente la ilusión’. Y todo ello se consigue con la fotogrametría.

Por otro lado, Unreal Engine es el encargado de conseguir, gracias a la fotogrametría, videojuegos tan espectaculares como el Tekken 7, el remake del Final Fantasy VII, etc. Con su tecnología, Unreal Engine 4 crea mundos virtuales y fotorrealistas en 4K a través de la fotogrametría.



Figura 2.8: La fotogrametría en Unreal Engine 4

Fuentes: Elaboración propia

Además de Unreal Engine 4, vale la pena mencionar al motor gráfico Unity 3d, ya que este motor es una de las principales competencias del Unreal Engine, ya sea por su posición en el mercado o por su potencial gráfica.

Y es que con esta técnica de escaneo 3D, ha permitido a estas empresas a recortar tanto los costes como los plazos durante el desarrollo. De esta forma, permite a las empresas lanzar proyectos realistas en periodos más cortos de tiempo.

2.6.PROCESAMIENTO DE IMÁGENES

El procesamiento, el análisis y la interpretación de imágenes es un campo de especialización muy importante de la visión artificial, que hace posible que un ordenador procese imágenes o fotografías bidimensionales (aisladas o bien conectadas en secuencias dinámicas o temporales).

Se trata de una clase de procesamiento de señales que puede tomar fotografías o fotogramas de vídeo y someterlos a un procesado de perfil bajo con el fin de extraer determinadas características o parámetros, o bien para elaborar nuevas imágenes procesadas como material de salida. En un sentido más amplio, la visión artificial puede ser considerada como una forma más genérica del procesamiento de imágenes, más cercana a la interpretación de

imágenes, cuya finalidad es procesar, comprender y descifrar características o rastrear objetos, con un objetivo claro, en una imagen o una secuencia de imágenes de vídeo.

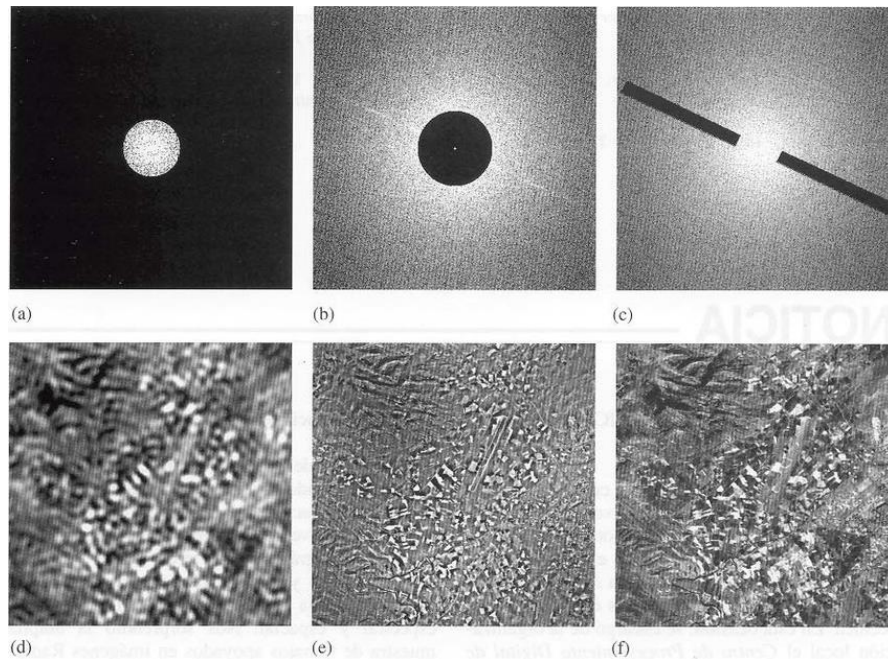


Figura 2.9: Ejemplos de filtro sobre una imagen

Fuente: Jan, 2015

El procesamiento de imágenes digitales es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

- **Proceso de filtrado**

Es el conjunto de técnicas englobadas dentro del pre procesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella.

Los principales objetivos que se persiguen con la aplicación de filtros son:

- Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.

- Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- Realzar bordes: destacar los bordes que se localizan en una imagen.
- Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad.

Por tanto, se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella.

El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio.

- **Filtrado en el dominio de la frecuencia**

Los filtros de frecuencia procesan una imagen trabajando sobre el dominio de la frecuencia en la Transformada de Fourier de la imagen. Para ello, ésta se modifica siguiendo el Teorema de la Convolución correspondiente:

- se aplica la Transformada de Fourier,
- se multiplica posteriormente por la función del filtro que ha sido escogido,
- para concluir re-transformándola al dominio espacial empleando la Transformada Inversa de Fourier.

Teorema de la Convolución (frecuencia):

$$G(u, v) = F(u, v) * H(u, v)$$

$F(u,v)$: transformada de Fourier de la imagen original.

$H(u,v)$: filtro atenuador de frecuencias.

Como la multiplicación en el espacio de Fourier es idéntica a la convolución en el dominio espacial, todos los filtros podrían, en teoría, ser implementados como un filtro espacial.

- **Tipos de filtro**

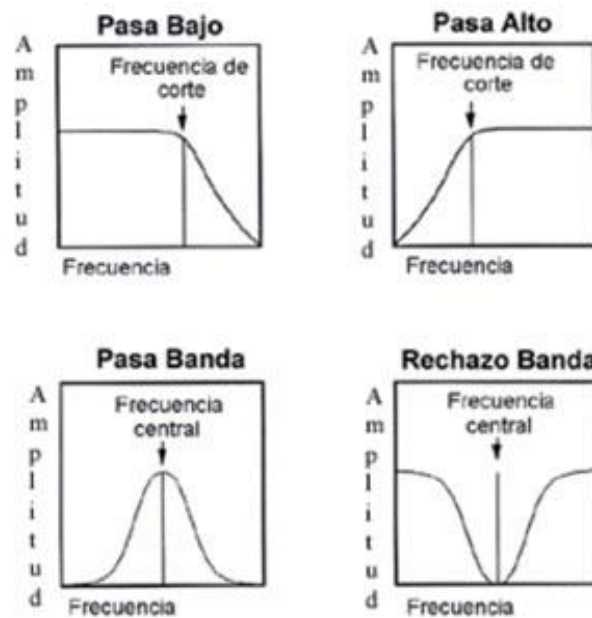


Figura 2.10: Filtros en el dominio de la frecuencia

Fuente: Jorge, 2014

Existen básicamente tres tipos distintos de filtros que pueden aplicarse:

- **Filtro paso bajo:** atenúa las frecuencias altas y mantiene sin variaciones las bajas. El resultado en el dominio espacial es equivalente al de un filtro de suavizado, donde las altas frecuencias que son filtradas se corresponden con los cambios fuertes de intensidad. Consigue reducir el ruido suavizando las transiciones existentes.
- **Filtro paso alto:** atenúa las frecuencias bajas manteniendo invariables las frecuencias altas. Puesto que las altas frecuencias corresponden en las imágenes a cambios bruscos de densidad, este tipo de filtros es usado, porque entre otras ventajas, ofrece mejoras en la detección de bordes en el dominio espacial, ya que estos contienen gran cantidad de dichas frecuencias. Refuerza los contrastes que se encuentran en la imagen.
- **Filtro paso banda:** atenúa frecuencias muy altas o muy bajas manteniendo una banda de rango medio.

- **Ventajas**

- Método simple y sencillo de implementar.
- Fácil asociación del concepto de frecuencia con ciertas características de la imagen; cambios de tonalidad suaves implican frecuencias bajas y cambios bruscos frecuencias altas.
- Proporciona flexibilidad en el diseño de soluciones de filtrado.
- Rapidez en el filtrado al utilizar el Teorema de la Convolución.

- **Desventajas**

- Se necesitan conocimientos en varios campos para desarrollar una aplicación para el procesamiento de imágenes.
- El ruido no puede ser eliminado completamente.

- **Filtrado en el dominio del espacio**

Las operaciones de filtrado se llevan a cabo directamente sobre los píxeles de la imagen. En este proceso se relaciona, para todos y cada uno de los puntos de la imagen, un conjunto de píxeles próximos al píxel objetivo con la finalidad de obtener una información útil, dependiente del tipo de filtro aplicado, que permita actuar sobre el píxel concreto en que se está llevando a cabo el proceso de filtrado para, de este modo, obtener mejoras sobre la imagen y/o datos que podrían ser utilizados en futuras acciones o procesos de trabajo sobre ella.

Los filtros en el dominio del espacio pueden clasificarse en:

- **Filtros lineales** (filtros basados en núcleos o máscaras de convolución).
- **Filtros no lineales.**

El concepto de núcleo se entiende como una matriz de coeficientes donde el entorno del punto (x,y) que se considera en la imagen para obtener $g(x,y)$ está determinado por el tamaño y forma del núcleo seleccionado. Aunque la forma y tamaño de esta matriz es variable y queda a elección de cada usuario, es común el uso de núcleos cuadrados $n \times n$.

Dependiendo de la implementación, en los límites de la imagen se aplica un tratamiento especial (se asume un marco exterior de ceros o se repiten los valores del borde) o no se aplica ninguno. Es por ello, que el tipo de filtrado queda establecido por el contenido de dicho núcleo utilizado.

Para realizar un filtrado en el dominio del espacio se realiza una convolución (barrido) del núcleo sobre la imagen. Para ello se sigue el Teorema de Convolución en el espacio:

$$g(x, y) = h(x, y) * f(x, y)$$

- Cada píxel de la nueva imagen se obtiene mediante el sumatorio de la multiplicación del núcleo por los píxeles contiguos.

$$g(x,y) = \sum \sum f(i,j) w(i,j)$$

- Generalmente se divide sobre cierto valor constante para normalizar que suele obtenerse de la suma de los valores del núcleo empleado.

2.7.TEXTURAS

Las texturas son bitmaps, mapas de información o mapas de bits. No dejan de ser información binaria que responde a un color representado en un píxel de la imagen. Cuanta más cantidad de píxeles más resolución y por ende más calidad, la calidad en los videojuegos actuales más modernos llegan a una resolución de 4096x4096 pixeles, aunque no a 60fps de forma estable (Rubi3d 2017).

A nivel general, la textura está vinculada a la superficie externa de un cuerpo. Se trata de una propiedad que es captada a través del sentido del tacto. La suavidad, la aspereza y la rugosidad son sensaciones que transmite la textura. Por ejemplo: “Amo la textura de esta camisa: es muy suave”, “La textura irregular del escritorio perjudicó mis trazos”, “Estas cremas contribuyen a suavizar la textura de mi piel”, “Los ciegos pueden leer gracias a la textura de las hojas escritas en Braille” (Pérez, J. Merino, M. 2013).

2.7.1. TIPOS DE TEXTURAS

Según el volumen se puede hablar de las siguientes texturas:

- **Visual:** estas texturas son las percibidas por medio de los órganos de la visión. Estas son representaciones gráficas, como fotografías, pinturas o dibujos, por lo que tienen sólo dos dimensiones, por tanto, carecen de relieve. Lo que buscan estas texturas es parecerse a objetos que existen y así transmitir realidad y viveza. Este tipo de textura puede ser logrado, en la pintura, por medio de la luz, el collage, la estampación o el frotado.
- **Táctil:** estas texturas, en cambio, son percibidas tanto con la vista como con el tacto y, a diferencia de las anteriores, sí poseen volumen, por lo que son tridimensionales. Las texturas táctiles de los objetos varían enormemente entre sí, pueden ser suaves, rugosas, lisas, etcétera.

Según el origen de las texturas, estas se clasifican en:

- **Naturales:** estas texturas son propias de las superficies y cuerpos que ofrece la naturaleza, por ejemplo, la corteza de un árbol, la piel del elefante.
- **Artificiales:** estas texturas, en cambio, provienen de aquellos objetos y superficies que fueron creados por las personas. Por ejemplo, la textura de una mesa o pared de ladrillos.

En **música**, el término textura alude a la manera en que se relacionan las distintas voces que forman parte de una pieza musical. Aquí se habla de las siguientes texturas: monódica, homofónica, heterofónica, polifónica, no melódica y acompañada.

2.8. MORFOLOGÍA DE IMÁGENES

La morfología matemática es una técnica de procesado no lineal de la imagen, interesada en la geometría de los objetos. Las operaciones morfológicas proporcionan información sobre la forma o estructura de una imagen (Serra, 1982).

- Las operaciones morfológicas simplifican imágenes y conservan las principales características de forma de los objetos.
- Un sistema de operadores de este tipo y su composición, permite que las formas subyacentes sean identificadas y reconstruidas de forma morfológica.

- Las formas subyacentes sean identificadas y reconstruidas de forma óptima a partir de sus formas distorsionadas y ruidosas.
- La morfología matemática se puede usar, entre otros, con los siguientes objetivos:
- Pre-procesamiento de imágenes (supresión de ruidos, simplificación de formas).
- Destacar la estructura de los objetos (extraer el esqueleto, detección de objetos, envolvente convexa, ampliación, reducción, entre otros)
- Descripción de objetos (área, perímetro, entre otros).

• **Dilatación**

Dada una imagen A, y un elemento estructural B, (ambas imágenes binarias con fondo blanco), la dilatación de A por B se define como:

$$A \oplus B = \{x | (\hat{B})_x \cap A \neq \emptyset\}$$

Tengamos en cuenta que, para la intersección sólo consideramos los píxeles negros de A y B.

El primer elemento de la dilatación, A, está asociado con la imagen que se está procesando y el segundo recibe el nombre de elemento estructural, la forma que actúa sobre A en la dilatación para producir.

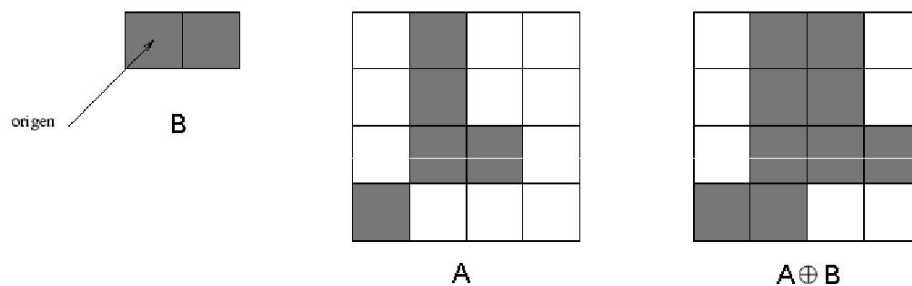


Figura 2.11: ejemplo del método dilatación

Fuente: Elaboración propia

observación: Es importante tener en cuenta que el sistema de coordenadas que se usará en este tema es (fila, columna).

- **Erosión**

Dada una imagen A, y un elemento estructural B, (ambas imágenes binarias con fondo blanco), la erosión de una imagen, A, por un elemento estructural, B, es el conjunto de todos los elementos x para los cuales B trasladado por x está contenido en A:

$$A \ominus B = \{x \mid B_x \subseteq A\}$$

Tengamos en cuenta que, para la condición $B_x \subseteq A$, sólo consideramos los píxeles negros de A y B.

La erosión es la operación morfológica dual de la dilatación.

La erosión se concibe usualmente como una reducción de la imagen original.

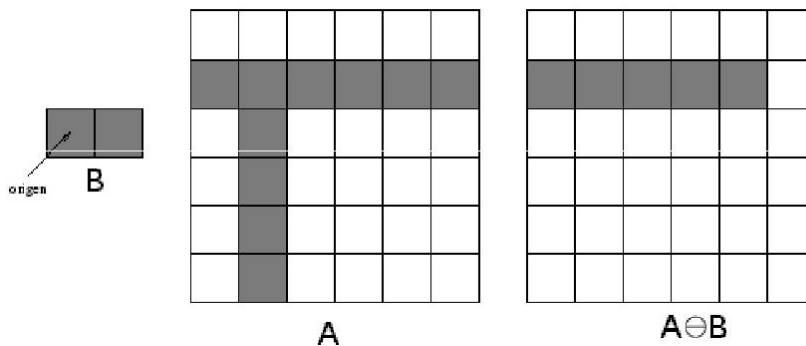


Figura 2.12: Ejemplo del método erosión

Fuente: Elaboración propia

- **Apertura**

La apertura de A por un elemento estructural K se define como

$$A \circ K = (A \ominus K) \oplus K$$

que, en palabras, establece que la apertura de A por K es simplemente la erosión de A por K, seguido de la dilatación del resultado por K.

Apertura Morfológica: Imágenes binarias

de A por K, seguido de la dilatación del resultado por K.

Si $A \circ K$ cambia con la apertura con K, diremos que A es abierto respecto a K.

Ejercicio: Da un ejemplo de un conjunto A y un elemento estructural K de más de un píxel de manera que A sea abierto respecto a K.

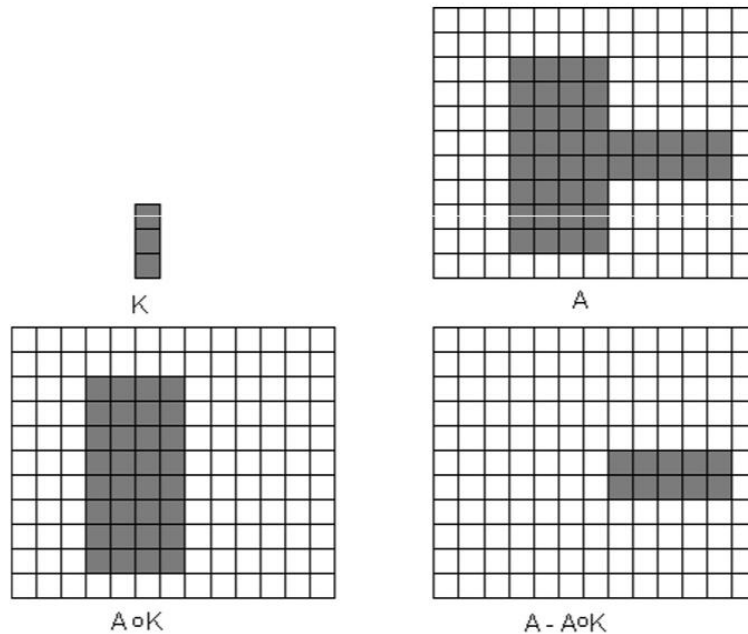


Figura 2.13: Ejemplo del método apertura

Fuente: Elaboración propia

- **Clausura**

La clausura de A por un elemento estructural K se define como

$$A \bullet K = (A \oplus K) \ominus K$$

que, en palabras, establece que la clausura de A por K es la dilatación de A por K

Morfología: Imágenes binarias Clausura

que, en palabras, establece que la clausura de A por K es la dilatación de A por K, seguido de la erosión del resultado por K.

2.9. LIBRERÍA DE VISIÓN ARTIFICIAL

OpenCV es una librería de computación visual creada por Intel, esta librería esta disponible para múltiples plataformas como: Windows, Linux, Mac, Android, además cuenta con soporte para diferentes lenguajes como: Python, Java, C/C++, .Net, entre otros, en esta serie de tutoriales aprenderás a manejar los conceptos básicos de la programación de aplicaciones enfocadas en el análisis y procesamiento de imágenes por computador. (Michael, E. 2017)

La biblioteca OpenCV puede ser usado bajo licencia BSD para proyectos escolares o comerciales, las aplicaciones de esta librería incluyen, la robótica, análisis y procesamiento de imágenes o vídeos, seguimiento y detección de objetos, detección y reconocimiento de rostros, reconocimiento de placas de vehículos, análisis de formas, reconstrucción 3D, realidad aumentada, y mucho más. (Michael, E. 2017)

2.10. QT CREATOR

Qt Creator es un IDE multi plataforma programado en C++, JavaScript y QML creado por Trolltech el cual es parte de SDK para el desarrollo de aplicaciones con Interfaces Gráficas de Usuario (GUI por sus siglas en inglés) con las bibliotecas Qt, Los sistemas operativos que soporta en forma oficial son:

- GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además, hay una versión para Linux con gcc 3.3.
- Mac OS X 10.4 o superior, requiriendo Qt 4.x.
- Windows XP y superiores, requiriendo el compilador MinGW y Qt 4.4.3 para MinGW.
- Qt Creator se centra en proporcionar características que ayudan a los nuevos usuarios de Qt a aprender, también aumentar la productividad.

• EDITOR AVANZADO DE CÓDIGO

Qt Creator se centra en proporcionar características que ayudan a los nuevos usuarios de Qt a aprender, también aumentar la productividad.

- Editor de código con soporte para C++, QML y ECMAScript

- Herramientas para la rápida navegación del código
- Resaltado de sintaxis y auto-completado de código
- Control estático de código y estilo a medida que se escribe
- Soporte para refactoring de código
- Ayuda sensitiva al contexto
- Plegado de código (code folding)
- Paréntesis coincidentes y modos de selección

- **DEPURADO VISUAL**

El depurador visual (visual debugger) para C++. Qt Creator muestra la información en bruto procedente de GDB de una manera clara y concisa.

- Interrupción de la ejecución del programa.
- Ejecución línea por línea o instrucción a instrucción.
- Puntos de interrupción (breakpoints).
- Examinar el contenido de llamadas a la pila (stack), los observadores y de la variables locales y globales.

2.10.1. QT CREATOR EN TECNOLOGÍAS 3D

Autodesk Maya (también conocido como Maya) es un programa informático dedicado al desarrollo de gráficos 3D por ordenador, efectos especiales y animación. Surgió a partir de la evolución de Power Animator y de la fusión de Alias y Wavefront, dos empresas canadienses dedicadas a los gráficos generados por ordenador. Más tarde Silicon Graphics (ahora SGI), el gigante informático, absorbió a Alias-Wavefront, que finalmente fue absorbida por Autodesk dueña de 3d Studio Max, por la cantidad de 182 millones de dólares.

Ejemplos de aplicaciones que utilizan Qt son: Autodesk Maya, Mathematica, Google Earth, el IDE RStudio, Spotify para Linux, la máquina virtual VirtualBox, VLC media player, y AMD Radeon Software Crimson Edition.

- Adobe Photoshop Album, aplicación para organizar imágenes.
- Avidemux, programa libre para la edición y procesamiento de video.

- Doxygen, API generadora de documentación.
- Gadu-Gadu, popular cliente polaco de mensajería instantánea.
- KDE, popular entorno de escritorio para sistemas operativos tipo-Unix.
- KDE Frameworks, una biblioteca base para gran cantidad de aplicaciones KDE, incluyendo Amarok, K3b, KDevelop, KOffice, entre otros.
- Last.fm Player, el cliente de escritorio para la popular comunidad web de música y radio por internet.
- Launchy, programa de código abierto para ejecutar aplicaciones para Windows.
- LMMS, secuenciador y sintetizador de música.
- LyX, un GUI para LaTeX.
- Mathematica, la versión de Linux usa Qt para el GUI.
- MythTV, grabador de vídeo digital de código abierto.
- Neural Designer, aplicación para la minería de datos.
- Psi, cliente de mensajería instantánea para XMPP.
- Quantum GIS, sistema de Información Geográfica.
- RealFlow, herramienta de simulación de fluidos.
- Scribus, aplicación para la publicación de escritorio.
- Skype, aplicación VOIP.
- TeamSpeak, aplicación multiplataforma para la comunicación con voz.
- Tlen.pl, popular cliente polaco de mensajería instantánea.
- TOra, herramienta para la administración de bases de datos.
- sMovieDB, catalogador/gestor de películas.
- Pandectas, aplicación para dibujo en pizarra digital.

CAPITULO III

MARCO APLICATIVO

3.1.INTRODUCCIÓN

En este capítulo se presentan los conceptos relevantes o necesarios para el desarrollo del software planteado en la presente investigación, con el fin de cumplir todos los objetivos propuestos en el Capítulo I, utilizando tecnologías de última generación, para que el software sea de calidad y que su desarrollo no tome mucho tiempo.

3.2.METODOLOGIA DE DESARROLLO RAD

Esta metodología por sus características que tiene, ayudará a la presente investigación, desarrollar un software de calidad, en corto tiempo, y utilizar tecnologías de última generación, permitiendo que sus futuros actualizaciones se lleven a cabo.

3.3.ETAPAS DE LA METODOLOGÍA DE DESARROLLO RAD

3.3.1. MODELADO DE GESTIÓN

Debido a las características que tiene la presente investigación, aplicar esta fase no se pudo llevar a cabo, pero es necesario aclarar que toda la información de entrada, los procesos de fotogrametría, modelado 3d, aplicar los métodos de detección y corrección de errores, la información de salida, las pruebas en distintos motores gráficos, y su posterior aceptación o rechazo, prácticamente dependen del usuario final.

3.3.2. MODELADO DE DATOS

A pesar de no contar con una base de datos, se tiene a disposición imágenes, imágenes que ayudaran al desarrollo del software planteado en la presente investigación, además con la ayuda de modelos 3d se podrá probar los resultados obtenidos con este software.

Las imágenes con la que se trabaja en la presente investigación ya sea que esta imagen sea resultado de la fotogrametría, o sea una imagen auxiliar obtenida de algún sitio del internet, en cualquier caso, es la principal información con la que se trabaja, y podría decirse que junto a los modelos 3d son los únicos datos que se tiene.

Para fines de probar la calidad e eficiencia del software planteado, se hace un largo proceso de fotografía a una variedad de objetos del mundo real, ya que en los mapas difusos resultantes el error en muchos casos será poco y en otros casos será bastante.

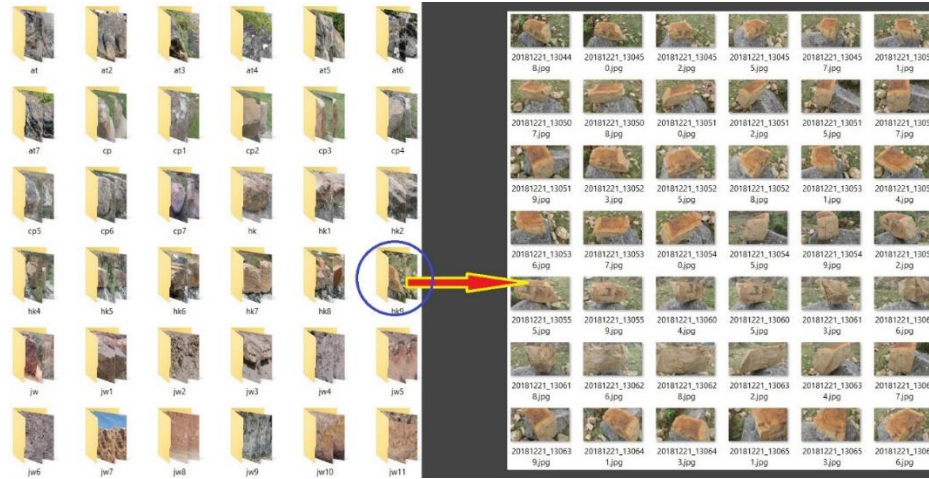


Figura 3.1: Conjunto de datos para la fotogrametría.

Fuente: Elaboración propia

El conjunto de datos de la figura 3.1, son fotografías de una variedad de objetos del mundo real, guardados en carpetas separados, estas imágenes son prácticamente la información base de la fotogrametría. Utilizando el conjunto de datos de la figura 3.1 en un software de fotogrametría, de cada objeto fotografiado, se tiene:

- Modelos 3d, de cada escaneo se pueden obtener desde modelos 3d de baja resolución hasta modelos de alta resolución.
- Mapa de colores correspondiente al modelo 3d, este mapa corresponde al modelo de alta resolución.

A partir de los modelos 3d, se pueden obtener las diferentes texturas que también son necesarias para representar un modelo 3d en un motor gráfico.

- **Texturas base:**

Estas texturas prácticamente son las que se necesitan para detectar y corregir errores en el mapa difuso fotogramétrico.

- **Mapa difuso**, el mapa difuso se obtiene con la técnica de transferencia del modelo de alta resolución proporcionado por Reality Capture y su posterior modelo de baja resolución.
 - **Mapa de opacidad**, el mapa de opacidad básicamente se obtiene de modelo de baja resolución.
 - **Textura Noise**, esta textura se puede generar en el software Photoshop, Substance Designer, entre otros, además de que también se puede descargar de forma gratuita del internet.
- **Texturas adicionales**

Estas texturas se consideran adicionales ya que no son relevantes en el proceso de detección y corrección de errores en el mapa difuso fotogramétrico, pero serán necesarios para probar los resultados obtenidos con la presente investigación, y comparar estos resultados con el mapa difuso original.

- **Mapa Normal**, el mapa normal, aunque no va a ser procesado, será necesario para probar los resultados finales en el motor gráfico y se puede obtener a través del modelo de alta resolución y el modelo de baja resolución.
- **Mapa Albedo**, esta textura es el mapa difuso original en su versión modificada, lo que significa que el mapa difuso fue retocado utilizando las técnicas de eliminación de sombra o simplemente que este haya sido calibrado con la tecnología cámara raw de Photoshop.
- **Mapa especular**, este mapa se puede obtener a partir del mapa difuso, y nos ayudara a visualizar mejor un modelo 3d en un motor gráfico.

Tabla 3.1: Características mínimas de las texturas a procesar

Imagen	formato	Tamaño mínimo	Tamaño máximo	Canales
Mapa difusa	tif	512x512	2048x2048	RGBA
Mapa de Opacidad	Jpg,png	512x512	2048x2048	RGB
Textura Noise	tif	512x512	2048x2048	RGBA

3.3.3. MODELADO DE PROCESOS

En esta fase se muestra todos los procesos involucrados con la presente investigación, pero es necesario resaltar que algunos de ellos son poco relevantes, por ejemplo, el proceso de fotogrametría, y el proceso de modelado 3d.

3.3.3.1.PROCESO DE FOTOGRAMETRÍA

El proceso de fotogrametría puede llegar a ser de fácil a muy complicado, y eso depende de los resultados que se esperan obtener; en este proceso se puede utilizar desde un simple celular, cámara profesional de alta resolución, color checker, entre otros.

Para la presente investigación se utilizó un simple celular como ya se mencionó anteriormente, los procesos necesarios son los siguientes:

- Realizar la captura desde los distintos ángulos del objeto que se está capturando.
- Procesar estas imágenes con el software Reality Capture, este proceso nos proporcionara modelos 3d, y un mapa de colores.

3.3.3.2.PROCESO DE FOTOGRAMETRÍA

El proceso de modelado 3d que se involucra con la presente investigación, se centra en la reducción de polígonos, en el que se utiliza el modelo de baja resolución, ósea que en este proceso se utiliza un software de modelado 3d para la reducción de polígonos y junto al modelo de alta resolución poder generar todas las texturas necesarias para representar un modelo 3d en un motor gráfico, ante todo esto, es necesario resaltar que este proceso se considera secundario, pero vale la pena aclarar que es necesario para generar algunas texturas auxiliares que son necesarias en la presente investigación, además es necesario aclarar que la presente investigación supone que, el usuario final del software a desarrollarse, tiene buena experiencia en modelado 3d y conocimientos básicos del software Photoshop.

Luego de aclarar todo lo anterior, se muestra todos los pasos involucrados en este proceso:

- Reducción de polígonos, la mayoría de los programas de modelado 3d ya vienen con algoritmos para reducir polígonos, pero a pesar de todo, prácticamente es necesario reducir más los polígonos manualmente.

- Obtención del mapa normal, el mapa normal se puede obtener utilizando el modelo 3d de alta resolución y el modelo 3d de baja resolución optimizado.
- Transferir el mapa de colores del modelo 3d de alta resolución al modelo 3d de baja resolución optimizado.
- Obtención del mapa de opacidad, este mapa se obtiene a partir del modelo optimizado y posteriormente utilizando Photoshop.
- Las texturas, mapa difuso, y el mapa noise deben tener el canal alpha, para el caso se utiliza Photoshop.

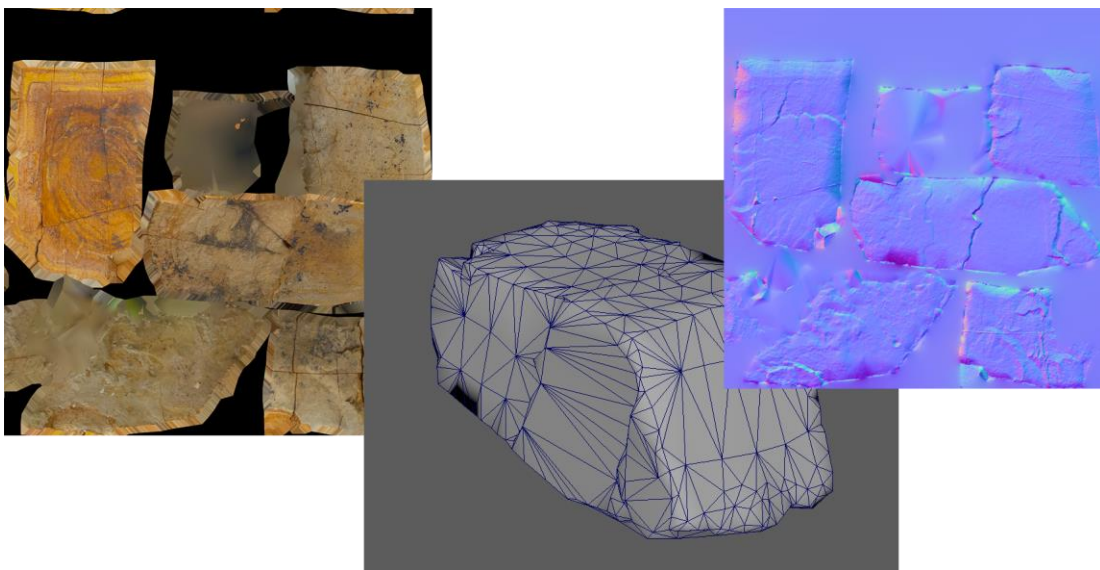


Figura 3.2: Resultados básicos de la fotogrametría

Fuente: Elaboración propia

Luego de obtener el modelo 3d optimizado y sus respectivas texturas, se hace una prueba directa de estos en un motor gráfico, lo cual significa que podrá ser posible ver la calidad de estos resultados y verificar si el mapa difuso tiene errores notorios.

Los errores en el mapa difuso y el mapa normal son las que más afectan a un modelo 3d representado en un motor gráfico, pero es preciso insistir que la presente investigación se centra en el mapa difuso fotogramétrico, pero vale la pena aclarar que los errores en el mapa normal se pueden corregir manualmente utilizando el modelo 3d de alta resolución en un software de modelado 3d.

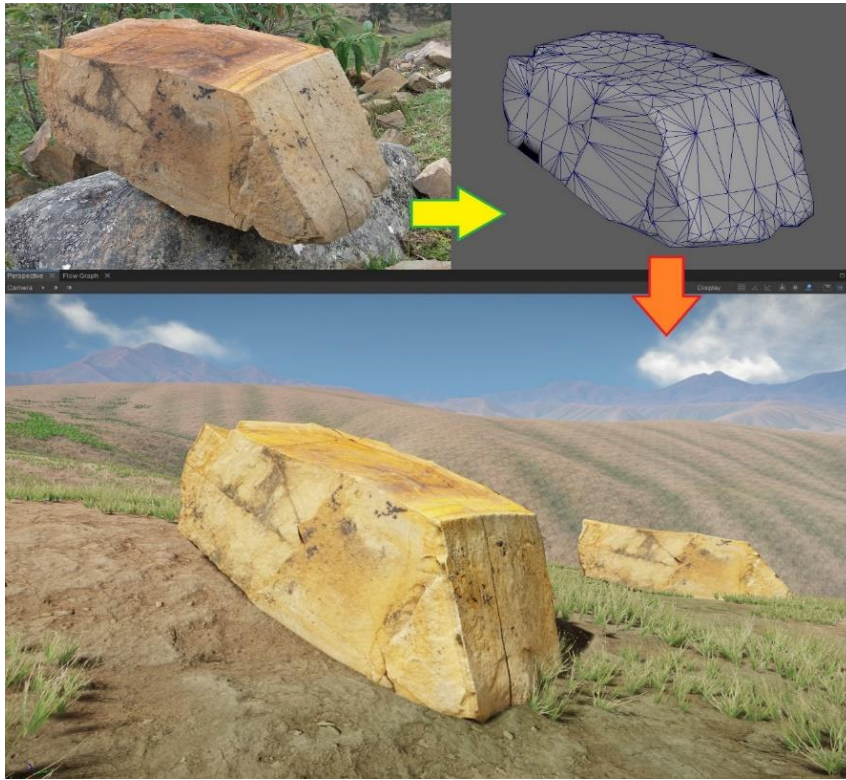


Figura 3.3: Resultado de la fotogrametría en cryengine

Fuente: Elaboración propia

Hay que resaltar que en la actualidad los motores gráficos brindan muchas herramientas para ayudar a los desarrolladores, y vale la pena aclarar que un de las mejores herramientas es el sistema de vegetación, este sistema permite distribuir un modelo 3d con tamaño, translación y rotación aleatoria, tal como se muestra a continuación.



Figura 3.4: Realismo con el sistema de vegetación de Cryengine.

Fuente: Elaboración propia

Viendo los resultados obtenidos en la figura 3.4, el sistema de vegetación es muy potente, pero ante todo, la desventaja de utilizar este sistema de vegetación es que, si un modelo 3d escaneado tiene algún error en una de sus mapas, ya sea el mapa difuso, el mapa normal, el mapa especular, entre otros, la probabilidad de que este error se vuelva muy notable es alto, lo que significa que esto afectara a gran escala a un escenario, dicho de otro modo, no será recomendable utilizar este modelo 3d con el sistema de vegetación.

3.3.3.3.PROCESO DE DETECCIÓN Y CORRECCIÓN DE ERRORES

Una vez que se tenga las texturas necesarias con las características necesarias, se procede a procesarlos con los métodos, detección y corrección de errores.

- **Método para detección de errores**, este método detectara los errores en el mapa difuso fotogramétrico y proporcionara este resultado de forma automática.
- **Método para corrección de errores**, este método corregirá los errores en el mapa difuso fotogramétrico, y exportará este resultado de forma automática.

3.3.3.4.PROCESO DE PRUEBAS Y RESULTADOS

Después de procesar las imágenes, se procede a ajustar estas imágenes resultantes en Photoshop según las necesidades, para posteriormente probarlos en el motor gráfico Cryengine V.

3.3.4. GENERACIÓN DE LA APLICACIÓN

Recordemos que la metodología de desarrollo RAD trata de minimizar la codificación en el desarrollo de un software, además de crear un software de calidad y minimizar el tiempo de desarrollo, por esta razón se optó por utilizar la tecnología qt designer, la librería de visión artificial, la librería blend modes y el lenguaje de programación Python, en otras palabras, las librerías mencionados minimizaran la codificación y el tiempo de desarrollo.

3.3.4.1.DISEÑO DE LA INTERFAZ GRÁFICA

Sabiendo que qt designer nos brinda herramientas dinámicas para la creación de interfaz gráfica y que esta interfaz se puede llevar a otros lenguajes de programación, se procede a crear la interfaz gráfica, de tal forma que este se ajuste a nuestros objetivos planteados en el capítulo I.

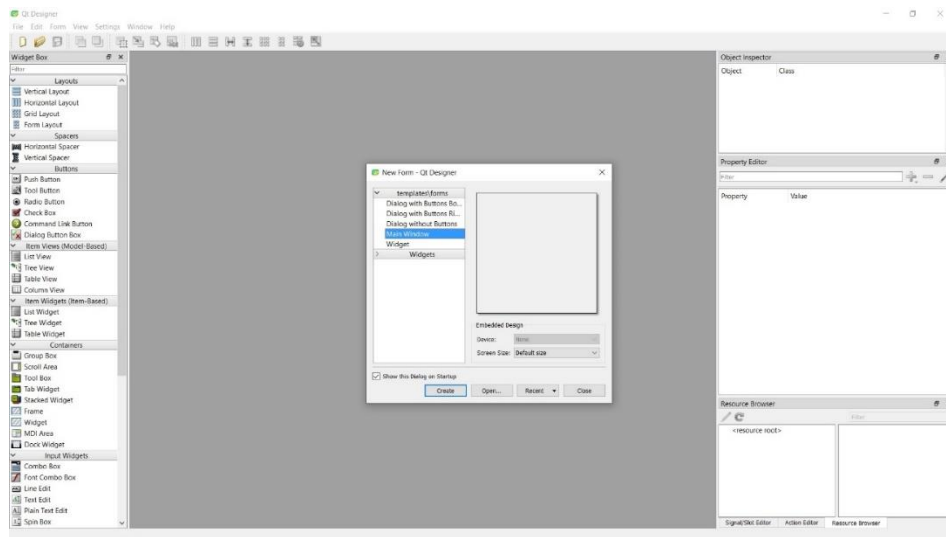


Figura 3.5: Software qt designer.

Fuente: Elaboración propia

Es necesario resaltar que qt designer se asemeja a visual estudio, dicho de otro modo, uno que sabe crear interfaz gráfica en visual estudio no tendrá problemas para crear un buen interfaz gráfica en qt designer.

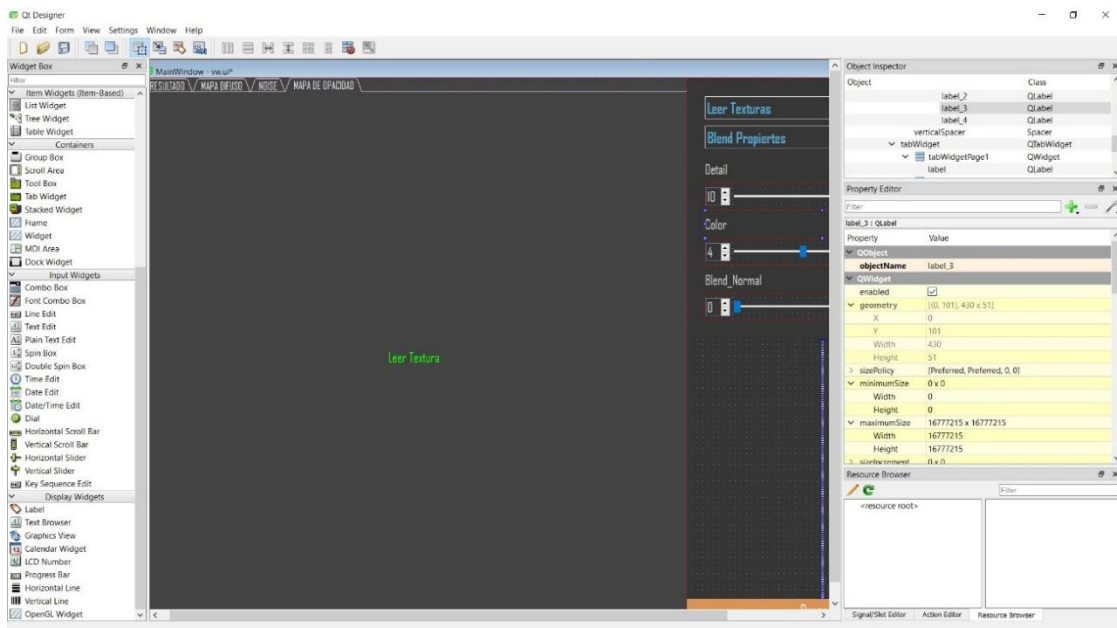


Figura 3.6: Interfaz gráfica del software en su primera versión

Fuente: Elaboración propia

La interfaz gráfica que se muestra en la figura 3.6 es el prototipo básico del software a desarrollarse, lo que significa que durante el desarrollo y pruebas se vio necesario modificar esta interfaz.

3.3.4.2.COMPILACION DE LA INTERFAZ GRÁFICA A LENGUAJE DE PROGRAMACIÓN PYTHON.

Una vez que se haya creado la interfaz gráfica en qt designer, es necesario convertir esta interfaz al lenguaje de programación python, para que esta interfaz sea interactiva, ya sea que pueda leer los datos de entrada, o procesar datos. Para compilar la ui generado en qt designer a lenguaje Python se utilizan el siguiente comando en cmd:

```
Pyuic5 -x vw.ui -o vw_ui.py
```

Luego el código debe ser ejecutado desde cmd utilizando el siguiente comando:

```
python vw_ui.py
```

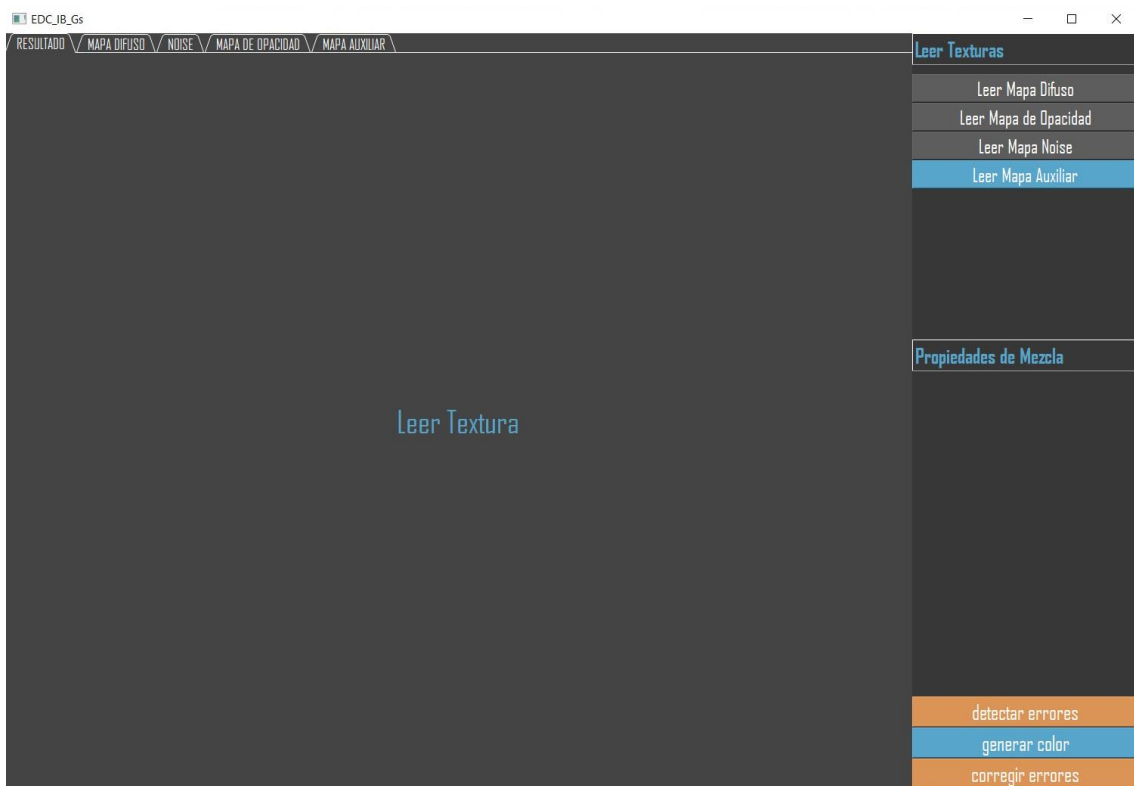


Figura 3.7: La interfaz gráfica del software en su versión final.

Una vez generado el código y además de probar que el software se ejecuta exitosamente se pueden adicionar los métodos o funciones necesarios para que el software pueda leer las texturas, procesarlos, visualizarlos y exportarlos.

3.3.4.3. ALGORITMO PARA LA DETECCIÓN DE ERRORES

En cuanto al procesamiento de imágenes, vale la pena aclarar que uno de los métodos más utilizados, es la detección de bordes, pero no hay que olvidar que este método puede llegar a ser muy complejo, ya sea utilizando Photoshop u otros programas, lo cual significa que si la imagen que se esté procesando es sencilla, aplicar el método también será sencillo, pero recordemos que en la presente investigación se está trabajando con imágenes resultantes de la fotogrametría, lo que significa que debido a la complejidad de estas imágenes no será posible aplicar este método, en otras palabras se debe implementar un método de detección de errores en un mapa difuso fotogramétrico sin utilizar el método de detección de bordes.

Para poder comprender el tipo de error que trata la presente investigación, es necesario ver este error en el mapa difuso fotogramétrico, de modo que pueda ser posible analizar este error y luego ver la forma de detectar este error y corregirlo.

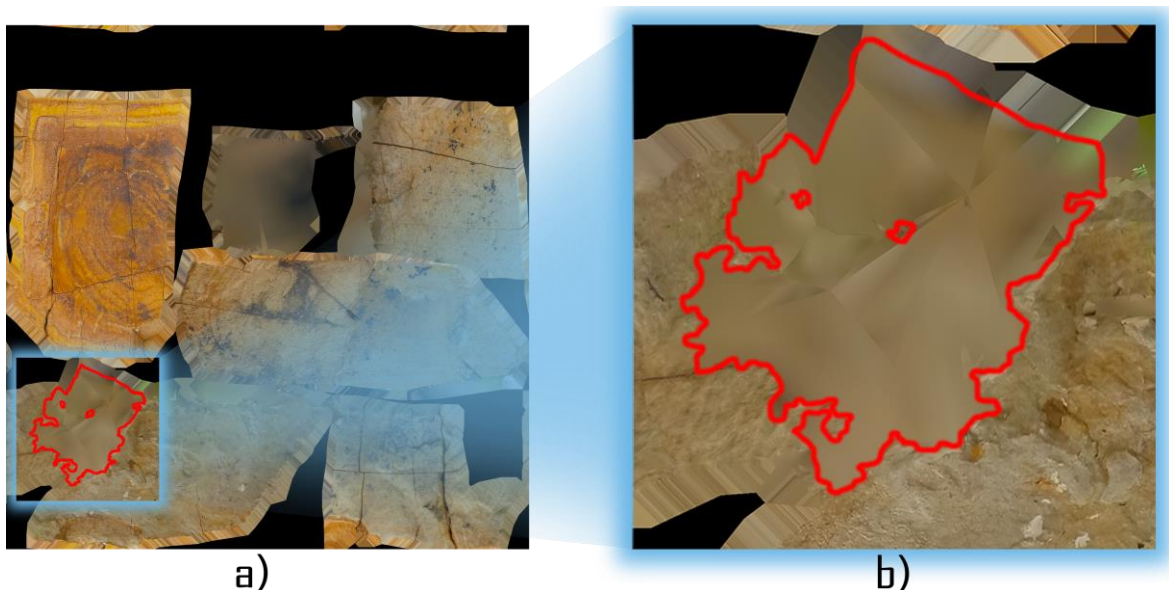


Figura 3.8: Área errónea en un mapa difuso fotogramétrico.

Fuente: Elaboración propia

La imagen b) de la figura 3.8 representa el área errónea remarcado de la imagen a), la ampliación de esta área errónea a una nueva imagen fue necesario para mostrar las características de los pixeles erróneos, los pixeles erróneos son los que parecen ser resultados de un método de suavizado; en otras palabras, el propósito de la presente investigación es detectar estos pixeles y corregirlos.

Para poder detectar todas las áreas erróneas en un mapa difuso fotogramétrico, fue necesario realizar varios análisis de las áreas erróneas y no erróneas, con el fin de ver el comportamiento de los pixeles en estas áreas.

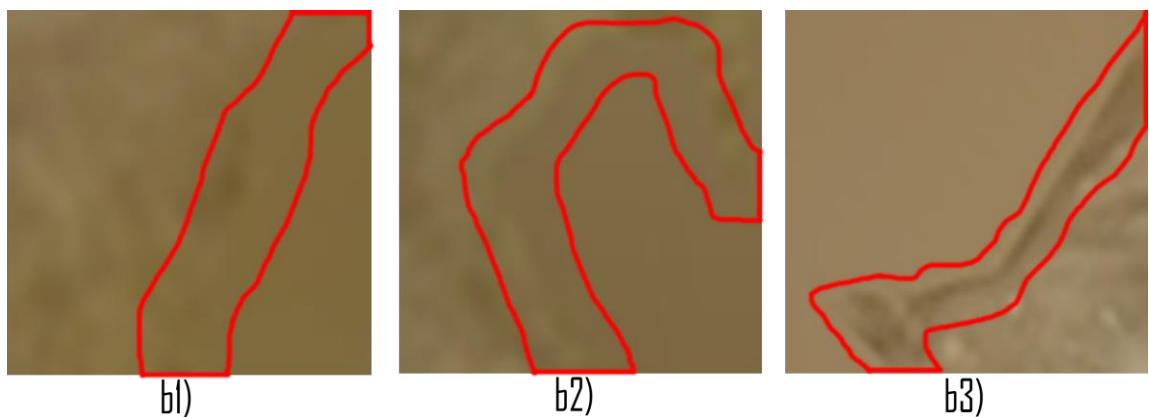


Figura 3.8: Los casos más comunes entre áreas erróneas y no erróneas

Fuente: Elaboración propia

Durante los análisis que se hizo, se encontró tres casos más comunes en mapas difusos fotogramétricos con áreas erróneas, los cuales son:

- Muy complicado de detectar el borde, la imagen b1) de la figura 3.8 representa al caso, donde ver el borde entre una área errónea y no errónea bastante complicado.
- Complicado de detectar el borde, la imagen b2) de la figura 3.8 representa al caso, donde ver el borde entre la área errónea y no errónea es menos complicado.
- Fácil de detectar el borde, la imagen b3) de la figura 3.8 representa al caso, donde ver el borde entre una área errónea y no errónea es fácil.

Luego de los análisis anteriores fue necesario analizar esto tres casos de forma numérica, tal como se muestra a continuación:

Tabla 3.2: Análisis de los pixeles en el caso fácil.

Pixel	R	G	B	Valor
1	141	121	84	346
2	140	120	83	343
3	141	121	84	346
4	140	120	83	343
5	125	106	66	297
6	116	97	57	270
7	125	105	68	298
8	141	123	85	349

En la tabla 3.2 el pixel con el valor 297, representa al borde y tiene una diferencia de 40 con el anterior pixel y 27 con el siguiente pixel lo que significa que detectarlo no será complicado.

Tabla 3.3: Análisis de los pixeles intermedio

Pixel	R	G	B	Valor
1	140	120	83	343
2	141	121	86	348
3	138	118	81	337
4	136	114	75	325
5	1127	108	68	303
6	131	111	74	316
7	139	121	85	345
8	135	117	79	331

En a la tabla 3.3. el pixel con valor 303, representa al borde y tiene una diferencia de 22 con el anterior pixel y 14 con el siguiente pixel, lo que significa que detectarlo será un poco complicado.

Tabla 3.4: Análisis de los pixeles en el caso complicado

Pixel	R	G	B	Valor
1	141	121	84	346
2	140	120	83	343
3	139	119	82	340
4	139	119	82	340
5	136	116	79	331
6	138	119	77	334
7	147	129	89	365
8	144	124	87	355

En la tabla 3.4 el pixel con el valor 331, es el borde, y tiene una diferencia de 9 con el anterior pixel y 3 con su siguiente pixel, al ver que estas diferencias son muy bajas en comparación a las tablas 3.2 y 3.3, esto significa que detectar el borde en este caso, será muy complicado.

Para los análisis que se muestran en las anteriores tablas se tomaron de forma más conveniente para poder mostrar los tres casos más comunes que existen al momento de la detección de bordes, y para este análisis se utilizó Photoshop.

Como resultado de estos tres análisis se vio necesario establecer un valor medio (VM), con el fin de que este valor nos pueda servir como referencia para detectar las áreas erróneas, de acuerdo a los análisis anteriores y otros se notó que cuando la diferencia es de un valor alto, detectar el área errónea será más sencilla, pero cuando pasa lo contrario la detección de las áreas erróneas es más complejo, partiendo de esta lógica se tomó los análisis de la tabla 3.2 y la tabla 3.4, donde las diferencias de los pixeles del borde con sus vecinos son 22,14,9,3, estos valores serán útiles para definir el valor medio, tal como se muestra a continuación:

$$VM = \frac{22 + 14 + 9 + 3}{4} = 12$$

Este valor (VM) será comparado con las diferencias entre pixeles, así para especificar si un pixel es erróneo o no errónea; este valor (VM) representa el valor máximo aceptado para

indicar si un pixel es erróneo o no, y con la ayuda de algunos de los métodos de la morfología de imágenes que viene implementa en la librería open cv, se detectara todas las areas erróneas en el mapa difuso fotogramétrico.

Después de realizar los análisis necesarios a diferentes áreas erróneas se procede a plantear el método para detectar las áreas erróneas en un mapa difuso fotogramétrico en los siguientes pasos:

- **Paso 1.** Pintar de blanco los pixeles cuya diferencia con sus vecinos es menor que el VM, caso contrario pintar de negro, este proceso se lo conoce como umbralizacion de una imagen. El método de umbralizacion consiste en la manipulación de un pixel, viendo las características que tiene este pixel.

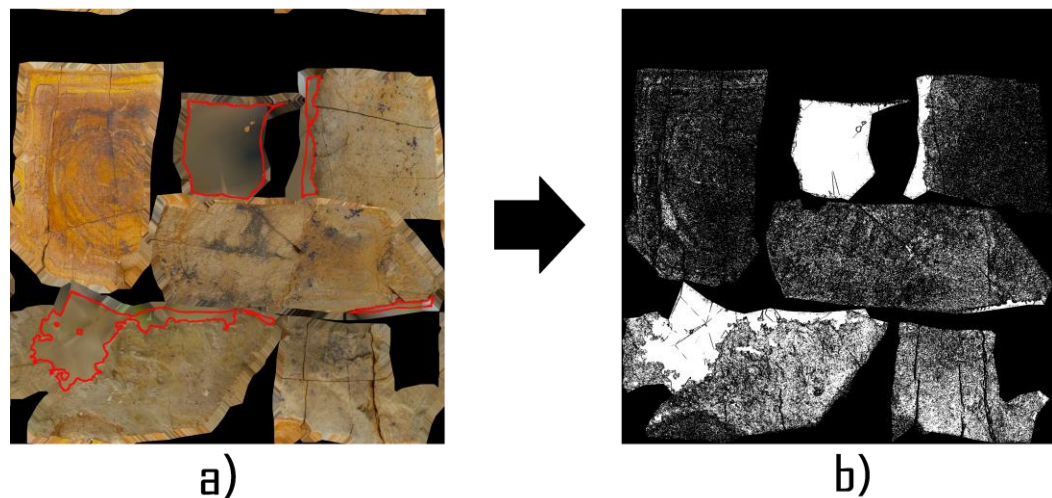


Figura 3.9: Ejemplo de umbralizacion de una imagen

Fuente: Elaboración propia

Viendo el resultado de la umbralizacion de la imagen a) de la figura 3.9, donde las áreas remarcados representan a las áreas erróneas, es necesario resaltar que muchos de los pixeles detectados como erróneas, no son erróneas, lo cual significa que debemos corregir este error, para ello se planteó utilizar los métodos morfológicos.

- **Paso 2.** Aplicar el método morfológico Open (open cv), este método eliminara los trozos de pixeles que no se debieron detectar como erróneas.

El método morfológico open es parecido al método morfológico erosión, lo que hace este método es eliminar los pixeles cuyos vecinos son diferentes, dicho de otra

forma, este método elimina pequeños trozos de píxeles blancos en una imagen en blanco y negro.

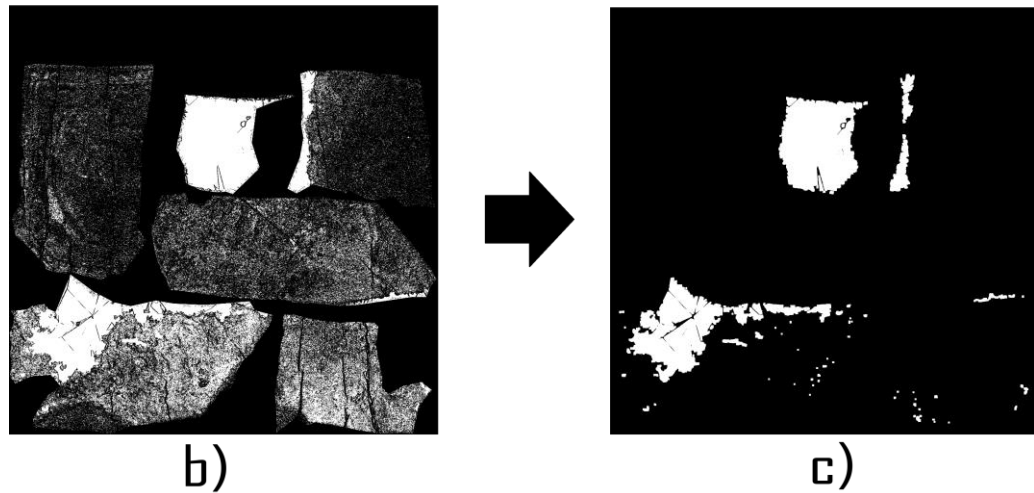


Figura 3.10: Ejemplo del método morfológico Open

Fuente: Elaboración propia

- **Paso 3.** Aplicar el método Close (open cv), aplicar este método eliminara algunos píxeles de color negro aislado de píxeles blancos.

El método morfológico Close es la inversa del método morfológico Open, dicho de otra forma, este método cierra pequeños puntos negros en una imagen, volviéndolos de color blanco.

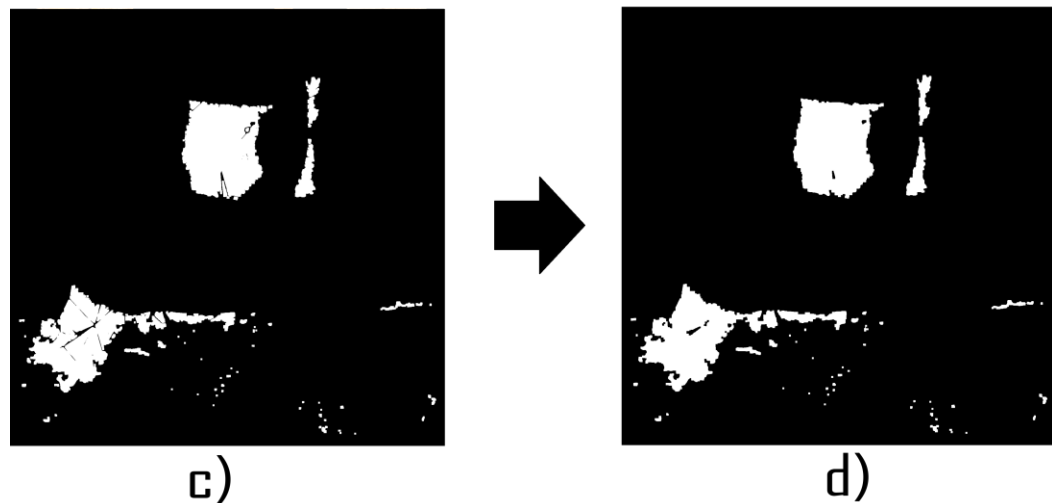


Figura 3.11: Ejemplo del método morfológico Close

Fuente: Elaboración propia

Aplicando el método morfológico Close a la imagen c) de la figura 3.11 se tiene como resultado a la imagen d).

- **Paso 4.** Aplicar el método Dilatation (open cv), este método ayudara a mejorar las áreas erróneas detectados.

En este método morfológico, un elemento de píxel es 1 si al menos un píxel debajo del núcleo es 1. Por lo tanto, aumenta la región blanca en la imagen.

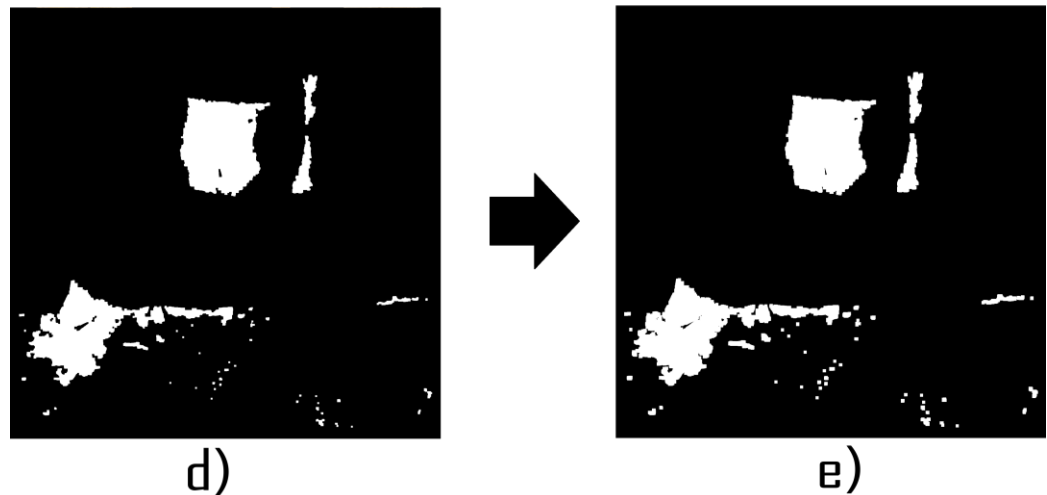


Figura 3.12: Ejemplo del método morfológico Dilatacion

Fuente: Elaboración propia

Aplicando el método Dilatacion a la imagen resultante de la figura 3.11 se tiene como resultado la imagen e).

La imagen resultante e) de la figura 3.12, es el resultado final de la detección de errores, donde, los pixeles blancos representan a los pixeles erróneas del mapa difuso fotogramétrico original, y los pixeles negros representan a los pixeles no erróneas del mapa difuso fotogramétrico original.

El algoritmo de detección de errores planteado anteriormente, está diseñado e implementado específicamente para que pueda detectar errores en un mapa difuso fotogramétrico, de modo que los resultados con otros tipos de mapas, texturas en general, o simples imágenes, son impredecibles, dicho de otro modo el algoritmo planteado para la detección de errores está probado en mapas difusos fotogramétricos con áreas erróneas, pero al contrario no se hizo pruebas con otros tipos de imágenes en general.

3.3.4.4.ALGORITMO PARA LA CORRECCIÓN DE ERRORES

Una vez que se hayan detectado las diferentes áreas erróneas y se haya generado la textura resultante, se procede con la corrección de estas áreas erróneas detectados, para ello es necesario resaltar que algunos algoritmos relacionados con la presente investigación están implementados en la librería open cv y que los resultados de estos algoritmos en un mapa difuso fotogramétrico no son los que se espera.

Después de una larga investigación, surgió la idea de corregir las áreas erróneas en un mapa difuso fotogramétrico mezclando esta imagen con una imagen auxiliar en escala de grises, vale la pena resaltar que los resultados de esta mezcla eran bastante satisfactorios, pero a pesar de corregir las áreas erróneas detectados, el borde que separaba estas áreas erróneas y las áreas no erróneas era fácil de notar, entonces fue necesario investigar la forma de corregir este inconveniente, y como resultado de esta investigación se concluyó que uno de los algoritmo de reconstrucción de imágenes, mismo que viene implantado en la librería open cv nos ayudaría a solucionar este problema.

A continuación, se detalla todos los pasos necesarios para la corrección de las áreas erróneas en un mapa difuso fotogramétrico:

➤ **Paso 1.** Aplicar el método inpainting telea (open cv).

El método inpainting telea requiere como datos de entrada dos imágenes, una imagen con errores, y otra imagen en blanco y negro, donde los pixeles blancos representan a los pixeles erróneos, y los pixeles negros representan a los pixeles no erróneos.

Las imágenes de entrada en el método inpainting telea, en la presente investigación son:

- Imagen original, esta imagen es el mapa difuso original, el mismo que se muestra como la imagen a) en la figura 3.9.
- Imagen resultante de la detección de errores, esta es la textura resultante del método de detección de errores, el mismo que se muestra como la imagen e) en la figura 3.12.

El método inpainting telea, tomando las imágenes de entrada, trata de dar color a los pixeles erróneos, para ello utiliza los pixeles no erróneos, y este resultado disimulara el borde entre las áreas erróneas y las áreas no erróneas.

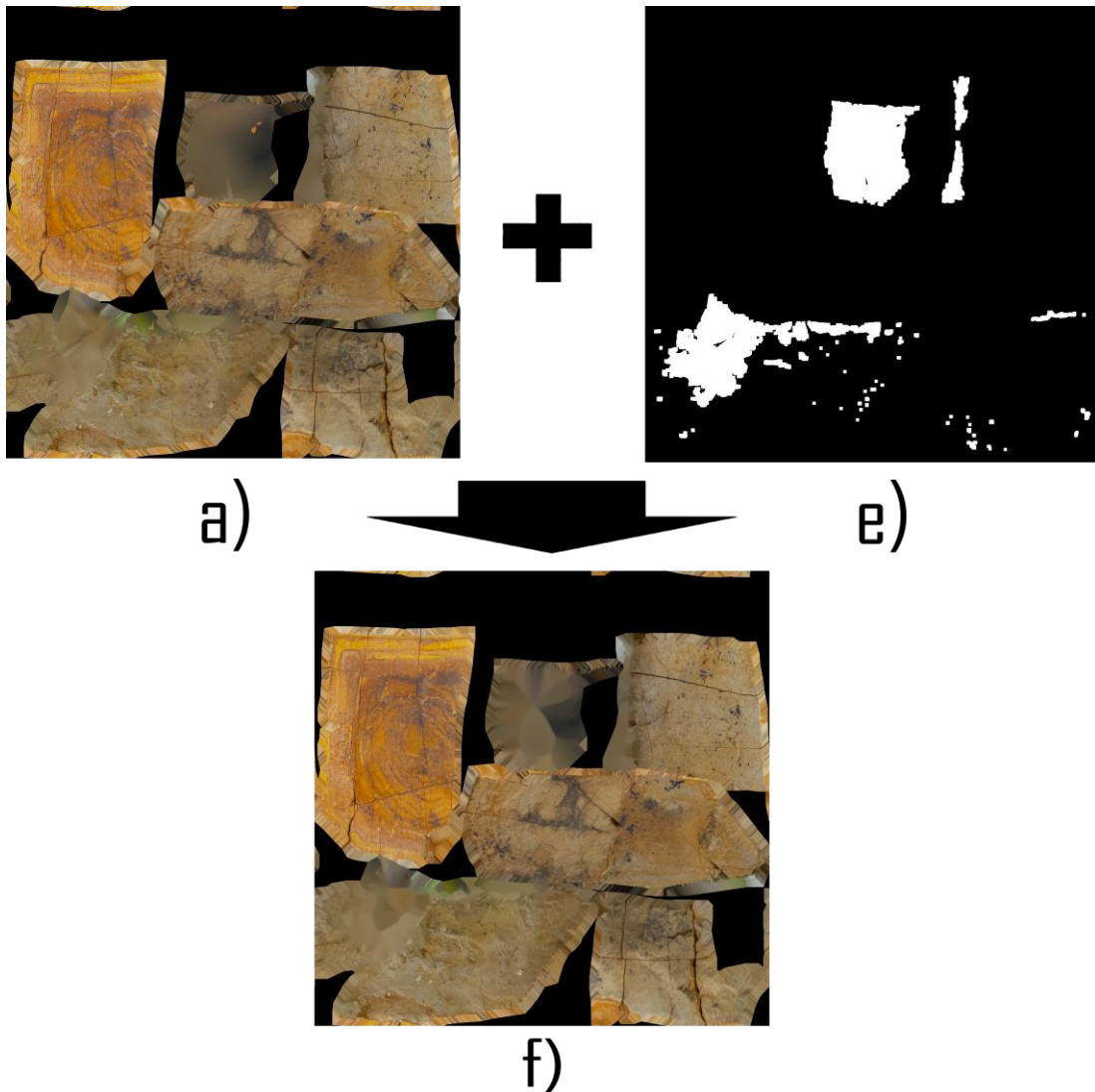


Figura 3.13: Ejemplo del método Inpainting telea

Fuente: Elaboración propia

La imagen resultante f) de la figura 3.13, es el resultado de aplicar el método inpainting telea, en la imagen a), que representa el mapa difuso original con errores y la imagen e), que representa al resultado de la detección de errores en el mapa difuso fotogramétrico.

➤ **Paso 2.** Aplicar el método de mezcla hard light (blend modes).

En este paso se plantea Mesclar la imagen resultante f) de la figura 3.13 con una imagen auxiliar en escala de grises.

El método de mezcla hard light tiene la capacidad de mezclar dos capas, utilizando una de ella para dar el nivel de luz en la otra imagen, dicho de otra forma, esta imagen dará detalle a nuestra imagen resultante f) de la figura 3.13.

El método hard light. Usa la media del color blanco y el color negro, y hace una combinación de estas (Robert, T. 20180).

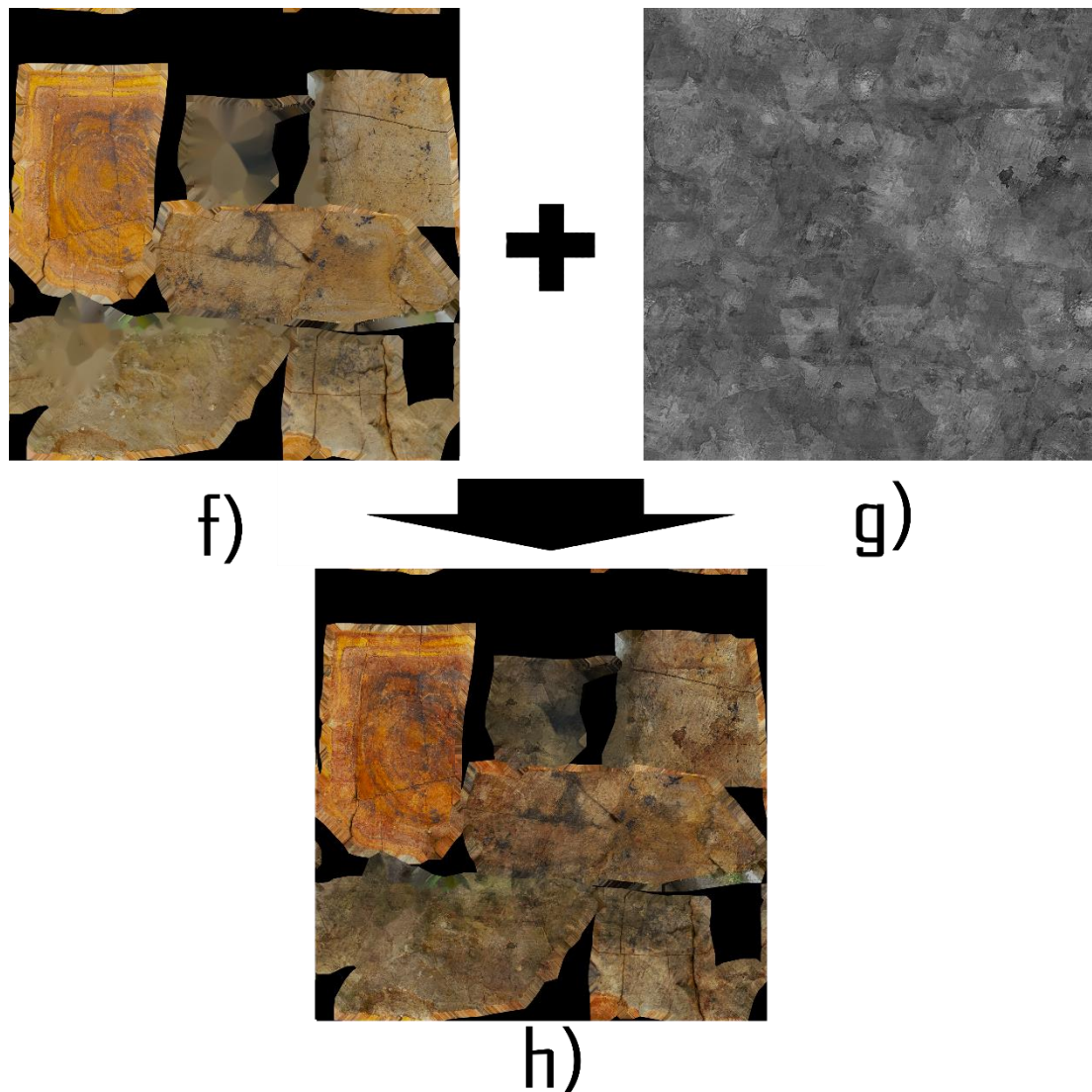


Figura 3.14: Ejemplo del método Hard_light

Fuente: elaboración propia

Utilizando la imagen f) de la figura 3.13 y una imagen auxiliar en escala de grises se tiene como resultado la imagen h) de la figura 3.14.

➤ **Paso 3.** Aplicar el método Screen (blend modes).

Luego de obtener la imagen h) como resultado en la figura 3.14, y comparando este resultado con el mapa difuso original, se tiene claro que los pixeles erróneos han mejorado bastante, pero los pixeles no erróneos también sufrieron cambios, entonces para tratar de recuperar el color inicial de estos pixeles, se propone aplicar el método de mezcla screen.

El método de mezcla screen ayuda a calibrar los colores manteniendo los blancos y tratando de desaparecer los negros (Robert, T. 20180).

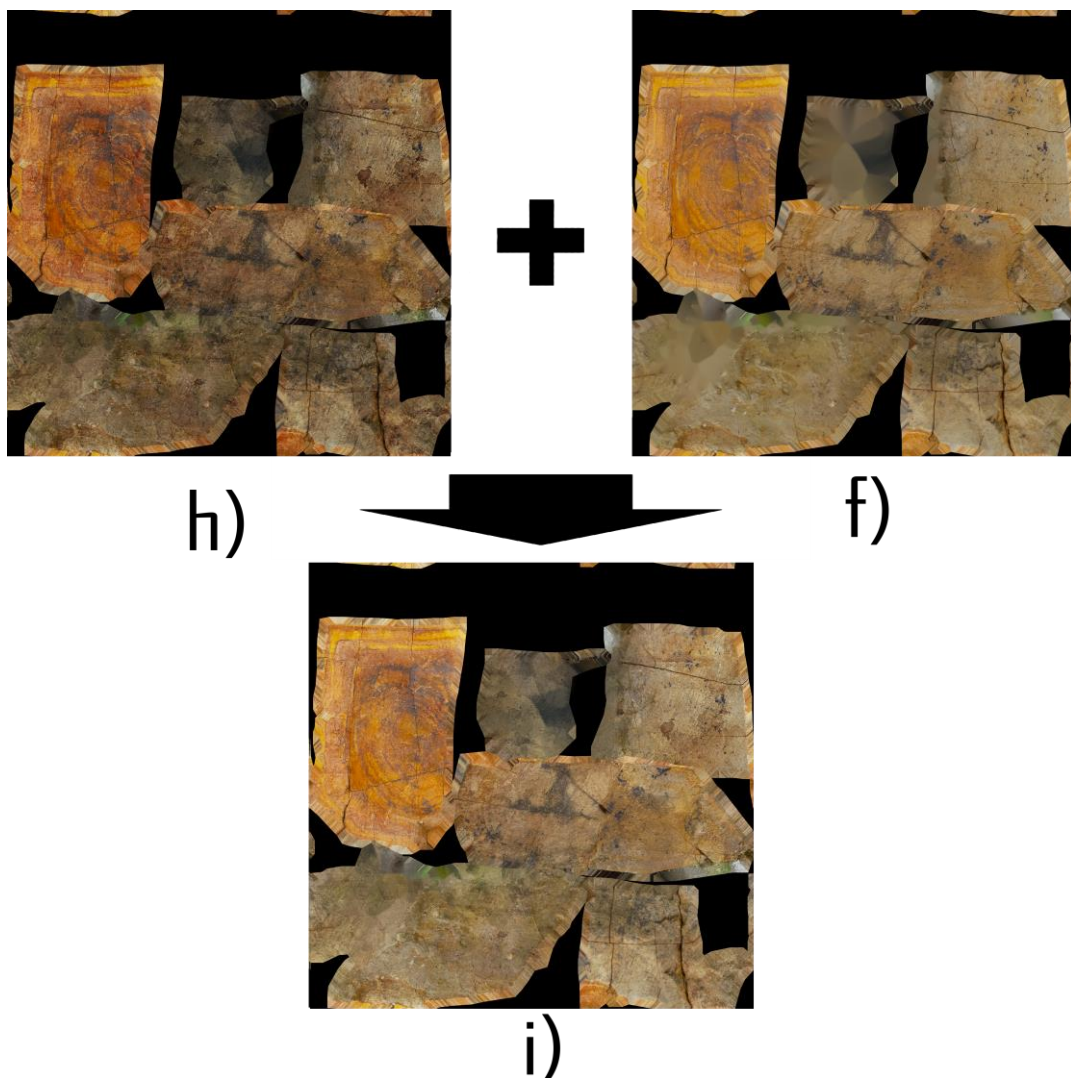


Figura 3.15: Ejemplo del método Screen

Fuente: Elaboración propia

La imagen i) de la figura 3.15 representa a los pixeles detectados como erróneas en su versión corregido.

➤ **Paso 4.** Recuperación de pixeles no erróneas.

Es necesario resaltar que, durante los procesos anteriores al corregir los pixeles erróneos, los pixeles no erróneos también sufrieron cambios, lo que significa que se deben recuperar los pixeles no erróneos en su versión original, para el caso, recordemos que la imagen e) de la figura representa todos los pixeles erróneos detectados, lo que significa que utilizando esta imagen podremos recuperar los pixeles no erróneos.

Luego para recuperar los pixeles no erróneos se utilizan las siguientes texturas:

- Mapa difuso original, este mapa se muestra como la imagen a) en la figura
- La textura resultante de la detección de errores, este resultado se muestra como imagen e) en la figura
- La textura resultante de la corrección de errores, este resultado se muestra como la imagen i) en la figura

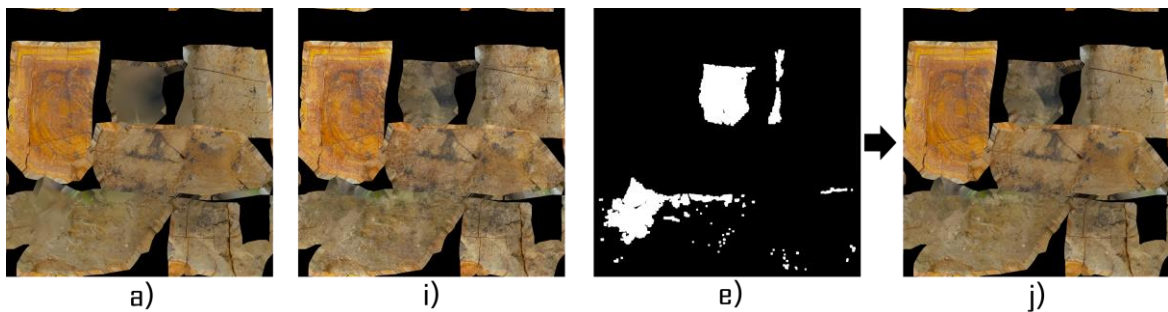


Figura 3.16: Recuperación de los pixeles no erróneos

Fuente: Elaboración propia

La imagen resultante j) de la figura 3.16 se obtiene combinando la imagen a) y la imagen i), además la imagen e) se utiliza como mascara en la imagen i) en el proceso de combinación de la imagen a), y la imagen i).

La imagen resultante k) de la figura 3.16, es el resultado final obtenido con la presente investigación.

3.3.5. PRUEBAS Y ENTREGA

Luego de concluir con el desarrollo del software se procede a probar la calidad y eficiencia de este software, utilizando una variedad de mapas difusos pertenecientes a diferentes modelos 3d, y además se recomienda, contar con una variedad de texturas noise, para que, al momento de realizar el procesamiento, vea cual es la textura que mejor se ajusta al mapa difuso fotogramétrico en procesamiento.

La aplicación final tiene la siguiente estructura:

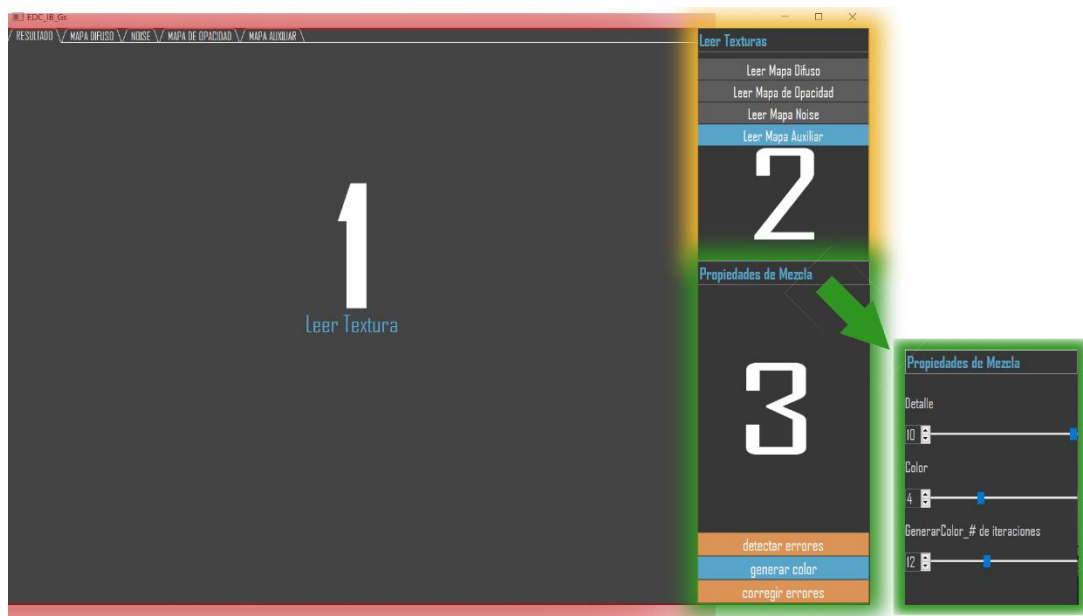


Figura 3.17: Software en su versión final.

El software se divide prácticamente en 3 partes:

Parte 1, esta parte está diseñada para mostrar todas las imágenes de entrada y salida.

Parte 2, en esta parte se muestra todas las opciones para leer los datos de entrada.

Parte 3, en esta parte se muestran las propiedades de mezcla, que prácticamente son para variar los resultados según a los resultados que se espera; con la opción “mezcla normal” se tenía como propósito mezclar todas las imágenes resultantes, pero no se pudo implementar.

Para utilizar este software se debe tener muy en cuenta lo siguiente:

- **Mapa Difusa**, este mapa debe tener 4 canales (RGBA).
- **Mapa de opacidad**, como es de esperar este mapa debe representar al mismo modelo que el mapa difuso, y además solo debe tener los canales RGB.
- **Textura Noise**, esta textura debe tener 4 canales (RGBA), además debe estar en escala de grises.

Otro caso muy importante es que todas estas texturas deben tener el mismo tamaño.

Una vez que se tengan todas las imágenes necesarias se procede con la detección de errores, para ello, se deben leer, el mapa difuso, el mapa de opacidad, y luego presionar el botón “detectar errores”.

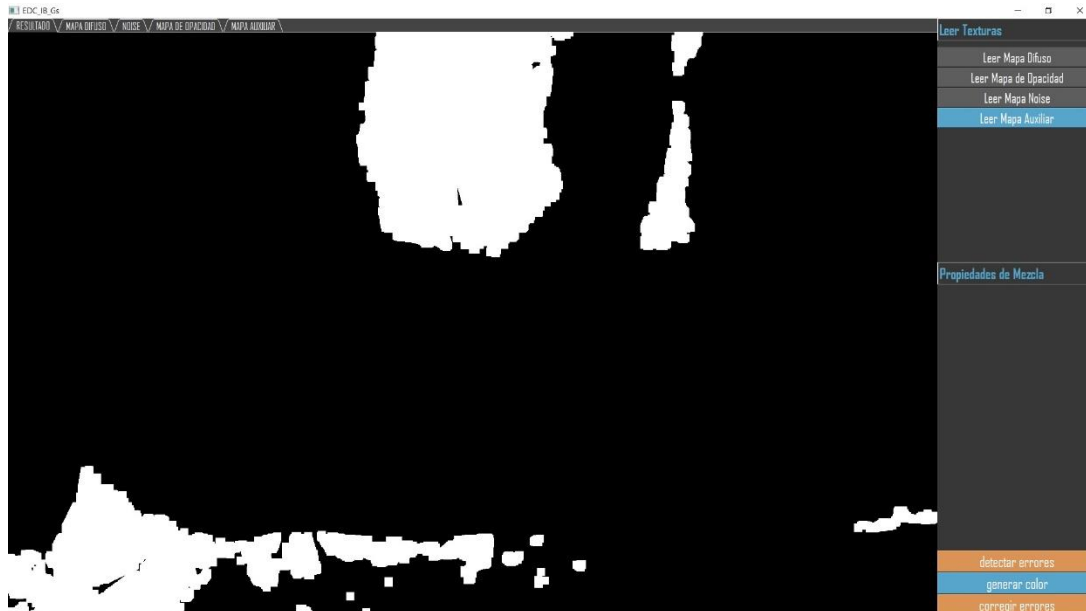


Figura 3.18: Resultado de la detección de errores

Una vez que el software muestra el resultado de la detección de errores, exportara de forma automática la textura resultante.

Después de detectar las áreas erróneas en el mapa difuso, se recomienda utilizar el algoritmo de reconstrucción de imágenes que viene implementada en open cv, haciendo clic en el botón “generar color”, este método ayudara a disimular el borde entre las áreas erróneas y no erróneas, pero antes de utilizar esta opción si o si se debe realizar el proceso de la detección de errores, ya que este método la textura resultante de la detección de errores.

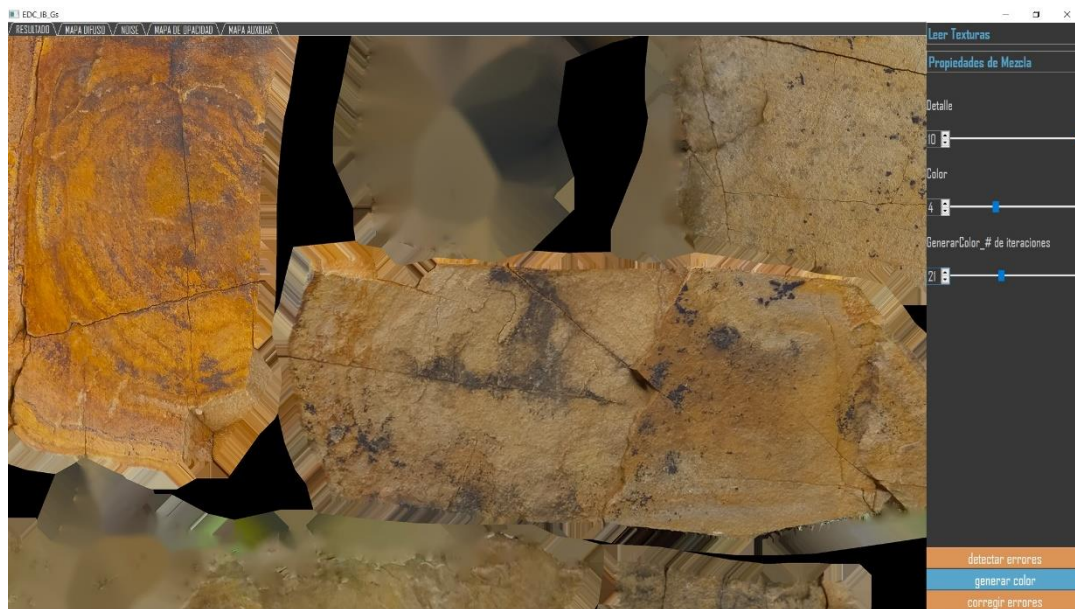


Figura 3.19: Resultado del algoritmo de reconstrucción inpainting telea.

El resultado de la figura 3.21 se puede manipular utilizando la opción “GenerarColor_# de iteraciones”, esta opción prácticamente está diseñada para poder mejorar más el resultado final, pero cuanto mayor sea el número de iteraciones, el algoritmo tardar más en sacar los resultados.

La opción “generar color” exportara de forma automática el resultado que se muestra en el software, pero para poder utilizar este resultado, se le debe adicionar el canal alpha en Photoshop manualmente y guardarlo en formato tiff, luego leer este mapa en el software con la opción “leer mapa auxiliar”, para que el software al momento de realizar la corrección de errores, reconozca este mapa difuso y trabaje con este mapa, en caso de que no se lleve a cabo este proceso, el software también podrá corregir los errores.

El algoritmo inpainting telea que esta implementada en la opción generara color, funciona de forma independiente y fue implementada para ayudar a mejorar el resultado final.

Después de tener el resultado de la detección de errores, se procede con la corrección de estos errores detectados, para ello se debe leer la textura noise(en escala de grises), luego presionar el botón “corregir errores”.

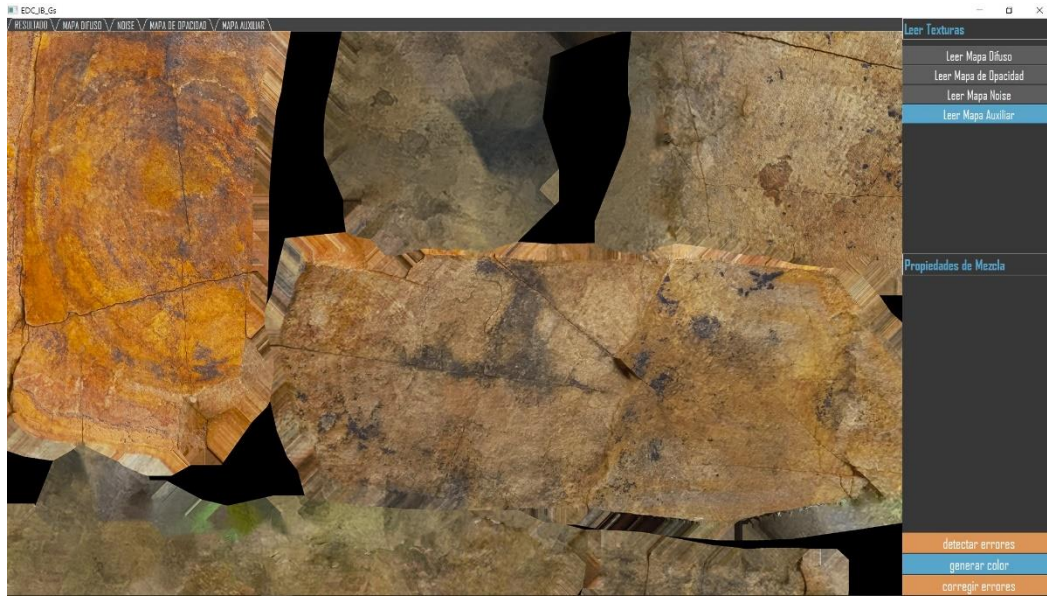


Figura 3.20: Resultado de corrección de errores.

Luego de que el software visualiza el resultado de la corrección de imágenes automáticamente este resultado es exportado.

Recordemos que en la presente investigación se optó por utilizar el motor gráfico Cryengine V, entonces antes de probar los resultados obtenidos con el software desarrollado debemos configurar estos resultados en Photoshop, de tal forma que el motor gráfico Cryengine pueda reconocer estos resultados.

4.2. RESULTADOS

Luego de hacer los ajustes necesarios en Photoshop se procede con la prueba de estas texturas resultantes en el motor gráfico Cryengine V, de tal forma que se pueda apreciar los resultados obtenidos, y además de comparar estos resultados con las texturas iniciales, las cuales tienen errores.

Con el propósito de visualizar y analizar los resultados obtenidos con la presente investigación, es para comparar estos resultados con los resultados iniciales de la fotogrametría, de tal forma que podamos decir si el software cumple o no con los objetivos propuestos en el capítulo I, para ello se muestra 2 modelos 3d en su versión original y su versión corregida.

El modelo 3d con el nombre “hk9” se muestra a continuación, en su versión original con las áreas erróneas remarcadas en el motor gráfico Cryengine V.

El mapa difuso del modelo 3d con el nombre “hk9”, es el mapa difuso que se utilizó como ejemplo en la presente investigación.

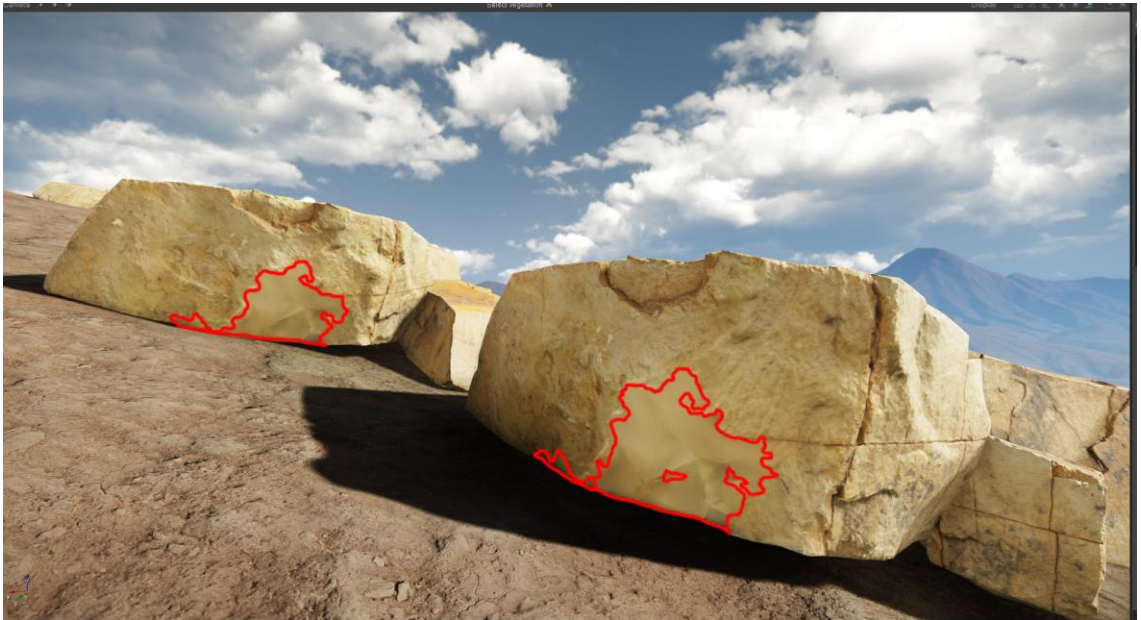


Figura 3.21: El modelo 3d “hk9”, con el mapa difuso original.



Figura 3.22: El modelo 3d “hk9”, con el mapa difuso con las áreas erróneas corregidas.

La figura 3.25 representa al mismo modelo 3d de la figura 3.24, con la diferencia que este utiliza el mapa difuso con las áreas erróneas corregidas.

Viendo los resultados ilustrados en la figura 3.24 y la figura 3.25, se puede decir que los resultados obtenidos con el software que se desarrolló en la presente investigación es muy satisfactoria, ya que diferenciar las áreas erróneas corregidas y las áreas no erróneas es bastante complicado.

Luego de ver los resultados obtenidos en la figura 3.25, a continuación, se optó por tomar un modelo 3d cuyo mapa difuso fotogramétrico tenga bastante cantidad de áreas erróneas, de tal forma ver el resultado que se tiene con el software desarrollado, entonces se tomó el modelo 3d con el nombre “hk1”, tal como se muestra a continuación.

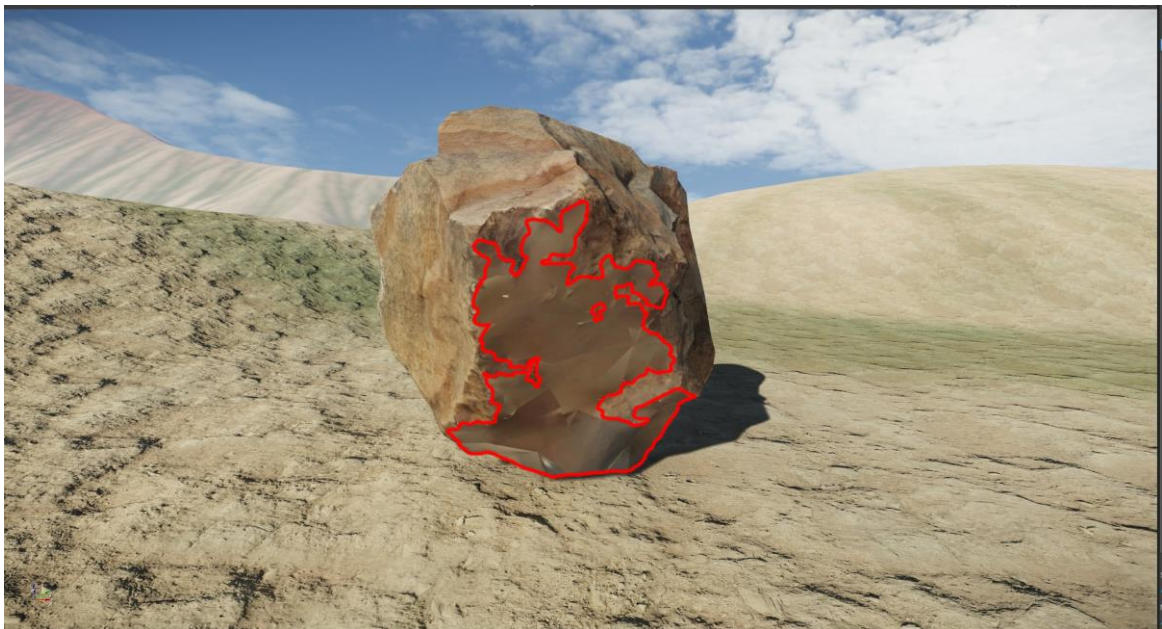


Figura 3.23: El modelo 3d “hk1”, con el mapa difuso original.

El modelo 3d de la figura 3.26 está representado en su versión original con las áreas erróneas remarcadas, en el motor gráfico Cryengine V, de tal forma que es bastante sencillo notar todas las áreas erróneas que tiene este modelo 3d en su mapa difuso.

A continuación, se muestra el mismo modelo 3d de la figura 3.26, con la diferencia que ahora este modelo 3d utiliza el mapa difuso fotogramétrico corregido por el software desarrollado en la presente investigación.



Figura 3.24: el modelo 3d “hk1”, con el mapa difuso corregido.

al ver el resultado obtenido en la figura 3.27, y comparando este resultado con el resultado de la figura 3.26, es casi imposible notar la diferencia entre las áreas erróneas corregidas y las áreas no erróneas.

Analizando los resultados obtenidos visualmente en Photoshop y el motor gráfico Cryengine V se tiene claro que este resultado obtenido en comparación a los resultados iniciales se considera bastante aceptable.

Además, con las opciones que brinda el software desarrollado, se pueden perfeccionar los resultados obtenidos con este software haciendo uso del software Photoshop, perfeccionando la detección y corrección de errores manualmente, además para perfeccionar los resultados obtenidos con la presente investigación sería, corrigiendo las áreas erróneas en el mapa normal, ya que un modelo 3d al tener errores en el mapa difuso, en consecuencia, también tiene errores en el mapa normal.

Viendo los resultados que se obtuvo con la presente investigación todos los modelos 3d escaneados con errores se pueden mejorar con el software desarrollado excluyendo el mapa normal, el mapa especular, ya que manipular estos mapas son más complicados en comparación al mapa difuso.

CAPITULO IV

PRUEBA DE HIPÓTESIS

En este capítulo se analizará los resultados obtenidos con la presente investigación, y se procederá con la prueba de la hipótesis planteado en el capítulo I, con el fin de validar estos resultados. La hipótesis que se planteo es, “La morfología de imágenes y la librería de visión artificial, permitirá al software detectar errores en el mapa difuso fotogramétrico y corregirlos con un acertamiento mayor al 85%”.

4.1. ANÁLISIS DE LOS RESULTADOS

La idea que se tomó para probar la hipótesis es, comparar dos resultados, estos resultados son:

- El resultado obtenido, es el resultado final, obtenido con la presente investigación.
- El resultado esperado, este resultado se obtiene en Photoshop, tomando el mismo mapa difuso fotogramétrico original con áreas erróneas, y corrigiendo estos errores manualmente.

La idea de comparar el resultado obtenido y el resultado que se esperaba obtener, es debido a que el algoritmo propuesto en la presente investigación no logra detectar todas las áreas erróneas al 100%, y para poder ver que tanto ha acertado el algoritmo propuesto, surge la necesidad de tener a disposición un resultado con todas las áreas erróneas corregidas, y por esa razón se usó Photoshop para obtener el resultado esperado.

Para poder comparar estos dos resultados, para empezar, se decidió calcular los valores de forma independiente, utilizando el mapa difuso original, junto al resultado obtenido y el resultado que se esperaba obtener.

- Caso 1, en este caso se toma el mapa difuso fotogramétrico original, y el resultado obtenido con la presente investigación, con el fin de calcular la diferencia entra estas dos imágenes, dicho de otra forma, al calcular la diferencia entre estas dos imágenes podremos saber que tanto ha cambiado el resultado final en comparación al resultado inicial con áreas erróneas.

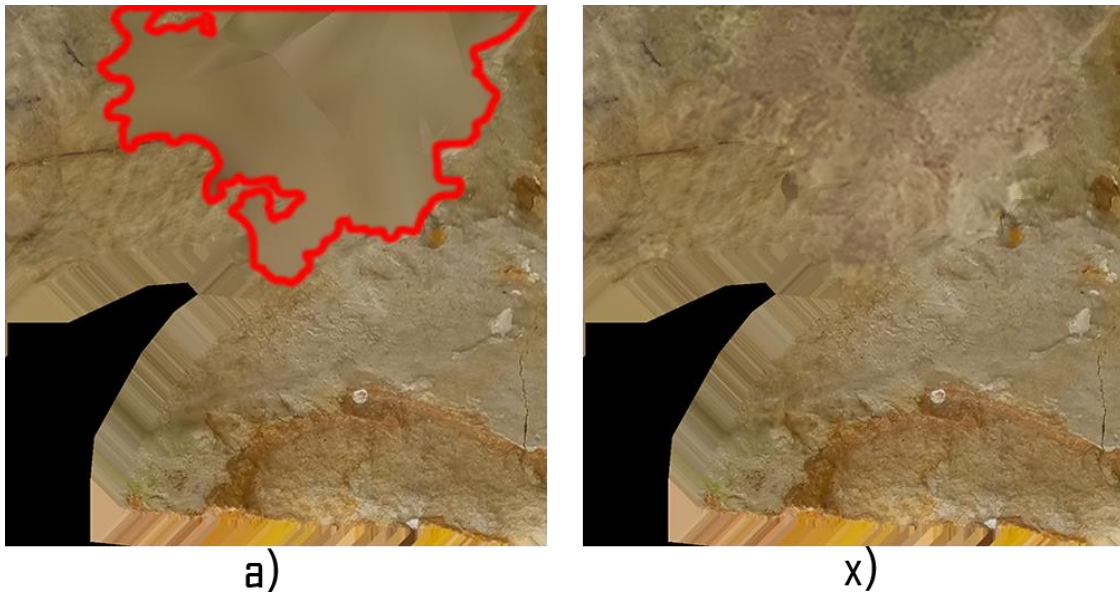


Figura 4.1: Resultado inicial y resultado obtenido

Fuente: Elaboración propia

La imagen a) de la figura 4.1 representa al mapa difuso fotogramétrico original, donde el área remarcada es el área errónea, mientras la imagen x) es el resultado obtenido con el algoritmo propuesto.

La idea es calcular las diferencias entre los píxeles de la imagen a) y la imagen x), y luego calcular la sumatoria de estas diferencias de la siguiente manera:

$$R_{i,j} = |aR_{i,j} - xR_{i,j}|$$

$$G_{i,j} = |aG_{i,j} - xG_{i,j}|$$

$$B_{i,j} = |aB_{i,j} - xB_{i,j}|$$

$$S_{i,j} = R_{i,j} + G_{i,j} + B_{i,j}$$

$R_{i,j}$, es la diferencia en valor absoluto del canal R del píxel i,j de la imagen a) y el píxel i,j de la imagen x).

$G_{i,j}$, es la diferencia en valor absoluto del canal G del píxel i,j de la imagen a) y el píxel i,j de la imagen x).

$B_{i,j}$, es la diferencia en valor absoluto del canal B del pixel i,j de la imagen a) y el pixel i,j de la imagen x).

Luego la sumatoria de todas las diferencias entre los pixeles de la imagen a) y la imagen x) de la figura 4.1, se calcula de la siguiente manera:

$$RO = \sum_{i,j}^{h,w} S_{i,j}$$

$$RO = 6812772$$

RO, es la sumatoria de las diferencias entre pixeles de la imagen a) y x), además este resultado representa al resultado obtenido.

- Caso 2, en este caso se toma el mapa difuso fotogramétrico original y el resultado obtenido manualmente en Photoshop.

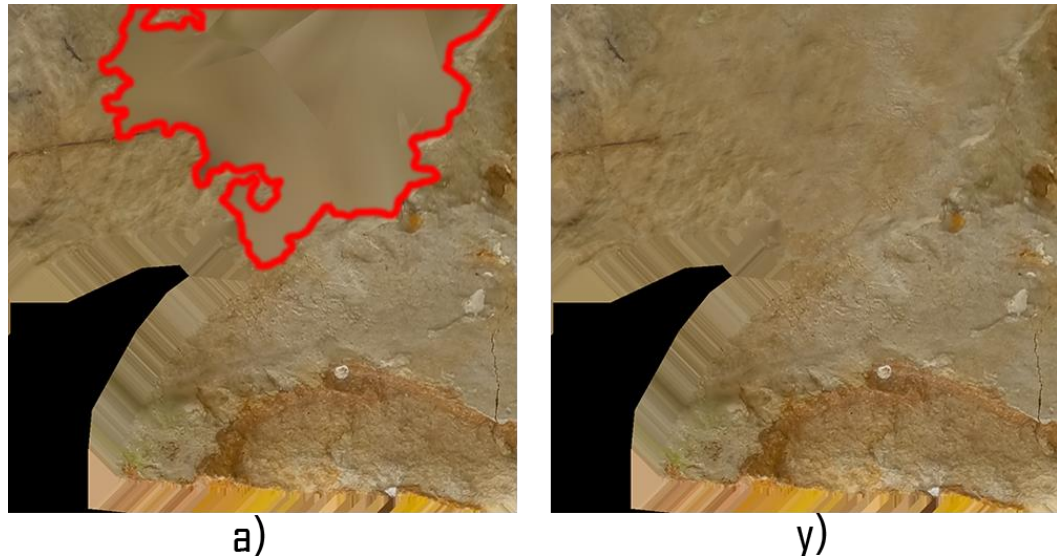


Figura 4.2: Resultado inicial y el resultado esperado

Fuente: elaboración propia

La imagen a) de la figura 4.1 representa al mapa difuso fotogramétrico original, donde el área remarcada, es el área errónea, mientras la imagen x) es el resultado obtenido manualmente en Photoshop.

Luego se procede con los mismos cálculos que en el caso anterior:

$$R1_{i,j} = |aR_{i,j} - yR_{i,j}|$$

$$G1_{i,j} = |aG_{i,j} - yG_{i,j}|$$

$$B1_{i,j} = |aB_{i,j} - yB_{i,j}|$$

$$S1_{i,j} = R1_{i,j} + G1_{i,j} + B1_{i,j}$$

$R1_{i,j}$, es la diferencia en valor absoluto del canal R del pixel i,j de la imagen a) y el pixel i,j de la imagen y).

$G1_{i,j}$, es la diferencia en valor absoluto del canal G del pixel i,j de la imagen a) y el pixel i,j de la imagen y).

$B1_{i,j}$, es la diferencia en valor absoluto del canal B del pixel i,j de la imagen a) y el pixel i,j de la imagen y).

Luego la sumatoria de todas las diferencias entre los pixeles de la imagen a) y la imagen y) de la figura 4.2, se calcula de la siguiente manera:

$$RE = \sum_{i,j}^{h,w} S1_{i,j}$$

$$RE = 7075265$$

RE, es la sumatoria de las diferencias de pixeles de la imagen a) y la imagen y) de la figura 4.2, además este resultado representa al resultado esperado.

Luego aplicando la regla de tres con en el resultado obtenido(RO) y el resultado esperado(RE) se tiene:

$$PA = \frac{100\% \times RO}{RE} = \frac{100 * 6812772}{7075265} = 96\%$$

PA, representa el porcentaje acertado.

Luego el porcentaje de error es:

$$PE = 100 - 96 = 4\%$$

El error del 5% es menor al 15% que se planteó, entonces para este caso la hipótesis que se planteó se cumple, pero para validar la hipótesis planteada de forma correcta se debe tomar un conjunto de mapas difusos fotogramétricos pertenecientes a diferentes modelos 3d, así

para tomar estos mapas y procesarlos con el software desarrollado y posteriormente calcular los valores numéricos de forma independiente tal como se hizo anteriormente, para el caso se toma una muestra de 15 mapas difuso fotogramétricos tal como se muestra a continuación:

Tabla 4.1 Muestra para la prueba de la hipótesis

Nombre	Resultado esperado	Resultado obtenido	P.A	P.E
Md	7075265	6812772	96%	4%
Md1	1520601	1805788	84%	16%
Md2	8247402	7328499	88%	12%
Md3	11592602	10151959	87%	13%
Md4	6171004	5258364	85%	15%
Md5	3568544	3295873	92%	8%
Md6	8416413	7770335	98%	2%
Md7	8804179	7282319	88%	12%
Md8	7962367	7241201	90%	10%
Md9	10393979	8722120	83%	17%
Md10	2553258	2418080	94%	6%
Md11	1183709	1097588	92%	8%
Md12	4451670	4149356	93%	7%
Md13	1251700	1088172	86%	14%
Md14	1714381	1530510	89%	11%

Los valores de la tabla 4.1 serán utilizados para probar la hipótesis, haciendo uso del estadístico t student.

4.2. PRUEBA DE HIPÓTESIS

Luego de analizar los resultados y verificar los porcentajes acertados, y ver que estos resultados varían en un rango de 83% y 98%, se procede a validar la hipótesis que indica que el software desarrollado detectará y corregirá errores en un mapa difuso fotogramétrico con un acertamiento mayor al 85%.

Utilizando el estadístico t student se tiene que, la hipótesis nula es lo contrario de la hipótesis alternativa, es necesario aclarar que la hipótesis alternativa es la hipótesis que se planteó en la presente investigación, entonces se tiene:

H₀: hipótesis nula

H₁: hipótesis alternativa

$$H_0 \leq 85$$

$$H_1 > 85$$

Luego de establecer los valores de la hipótesis nula y alternativa, se procede con la demostración usando la distribución t student.

- **Calculado la Media**

La media es la sumatoria de todos los porcentajes acertados de la tabla 4.1 dividido entre 15.

$$\bar{x} = \frac{\sum_{i=1}^n X_i}{n} = 90$$

- **Calculado la Varianza**

$$V = \sqrt{\sum_{i=1}^n \frac{(X_i - \bar{x})^2}{n - 1}} = 4,44$$

Luego de calcular la media, la varianza, se procede con reemplazar estos valores en la fórmula de distribución de t student.

$$t = \frac{X - H_0}{V/\sqrt{n}}$$

$$t = 4,36$$

Considerando el nivel de significancia que se toma en proyectos de investigación, se tiene:

$$\alpha = 0.05$$

Considerando el tamaño de muestra, calculamos el grado de libertad(gl) de la siguiente forma:

$$gl = n - 1 = 14$$

Luego considerando el nivel de significancia y el grado de libertad en la tabla t student, se tiene:

$$t_{0.05,14} = 1,761$$

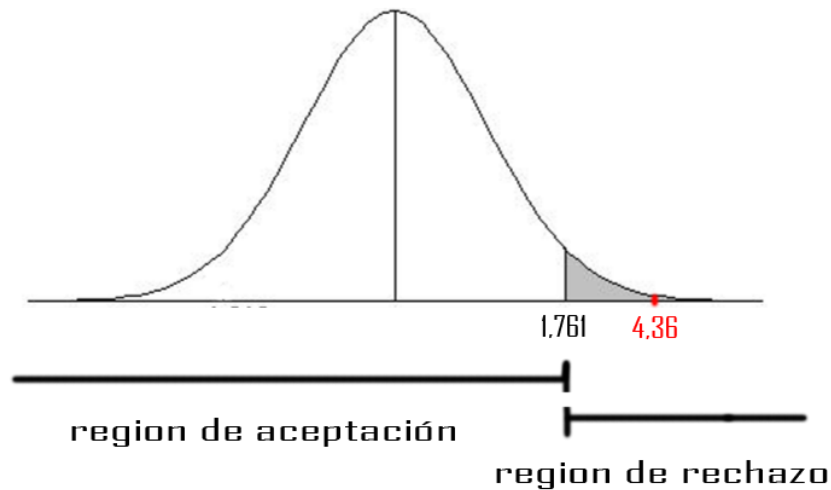


Figura 4.3: Región de aceptación de la hipótesis nula

Fuente: Elaboración propia

Como el resultado obtenido 4,36 es mayor que 1,76, la hipótesis nula se rechaza y la hipótesis alternativa se acepta. Luego al probar que la hipótesis planteado se cumple, podemos afirmar que el software desarrollado es de calidad e eficiente.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

1.1.CONCLUSIONES

De acuerdo a los resultados obtenidos anteriormente se concluye lo siguiente:

- ✓ El objetivo principal de la tesis fue cumplido, ya que se cumplió con la creación de un software dinámica que pueda detectar y corregir las arreas erróneas de un mapa difuso fotogramétrico con un acertamiento mayor al 85%.
- ✓ El objetivo de tener la menor cantidad de errores posibles en el mapa difuso fotogramétrico se cumplió, ya que, al considerar el clima en el proceso de fotogrametría, se obtuvo la menor cantidad de errores, y además esto también favoreció a las técnicas de eliminación de sombra en el mapa difuso fotogramétrico.
- ✓ El objetivo de evitar tener áreas de pixeles anormales en el mapa difuso fotogramétrico fue complicado, ya que al considerar todos los objetos que rodean a nuestro objetivo a escanear, viendo que ninguno de ellos logre afectar a nuestro objeto, ya sea con el brillo o sombra.

1.2.RECOMENDACIONES

Es necesario que se tomen en cuenta ciertas recomendaciones para mejorar este trabajo o futuras investigaciones relacionadas que tengan algún tipo de relación con esta:

- ✓ Para obtener mejores resultados en comparación con la presente investigación se recomienda aplicar un algoritmo con el que se pueda calcular el vector normal para cada pixel perteneciente al borde de las áreas no erróneas y calcular la interpolación de esta con la textura auxiliar (textura noise) que se utilizó en la presente investigación.
- ✓ Para perfeccionar los resultados se recomienda ajustar el modelo 3d escaneado de alta calidad antes de obtener el mapa normal, de tal forma que la arreas erróneas no puedan ser afectados por este mapa.
- ✓ Para obtener un resultado sin errores en el proceso de fotogrametría se recomienda, ver si el objeto que se quiere escanear, podrá ser posible fotografiarlo desde todos sus ángulos.

BIBLIOGRAFÍA

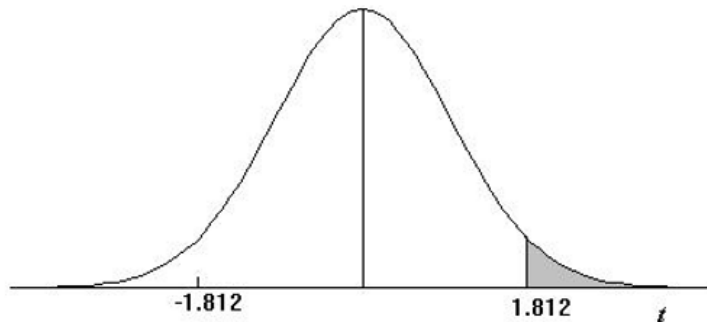
- UNITY3D. 2017. Crear activos de juegos con calidad de imagen fotográfica. [en línea]<<https://unity.com/es/solutions/photogrammetry>>[consulta: 7 mayo 2019]
- MOORE, H. 2015. Creando activos para la demo del mundo abierto para Unreal Engine 4. [en línea] <<https://www.unrealengine.com/en-US/blog/creating-assets-for-open-world-demo>>[consulta: 27 abril 2019]
- HAMILTON, A. BROWN, K. 2016. photogrammetry and star wars battlefront, [en línea] <<https://www.ea.com/frostbite/news/photogrammetry-and-star-wars-battlefront>>[consulta: 24 febrero 2019]
- OH, M. 2015. Imperfección por la perfección en texturas. [en línea]<<https://www.unrealengine.com/en-US/blog/imperfection-for-perfection>>[consulta: 27 abril 2019]
- UNITY3D. 2017. Manual oficial del flujo de trabajo de la fotogrametría en unity3d. [en línea] <https://unity3d.com/files/solutions/photogrammetry/Unity-Photogrammetry-Workflow_2017-07_v2>[consulta: 4 mayo 2019]
- FLYNN, T. 2018. Unity photogrammetry De-lighting Workflow. [en línea] <<https://sketchfab.com/blogs/community/unity-photogrammetry-de-lighting-workflow/>>[consulta: 24 mayo 2019]
- COLLINS, T. 2017. Una mirada a la fotogrametría y los videojuegos. [en línea] <<https://medium.com/@homicidalnacho/a-look-into-photogrammetry-and-video-games-71d602f51c31>>[consulta: 24 mayo 2019]
- 3DCOLLECTIVE. 2018. Guía de fotogrametría. [en línea] <<https://3dcollective.es/fotogrametria/>>[consulta: 25 mayo 2019]
- LACHAMBRE, S. LAGARDE, S. 2017. Photogrametry Workflow Layered shader. [en línea] <https://unity3d.com/files/solutions/photogrammetry/Unity-Photogrammetry-Workflow-Layered-Shader_v2.pdf?_ga=2.262382732.1841641142.1560343316-1571804154.1555436954>[consulta: 5 abril 2019]
- UNITY. 2017. Creating Assets with Photogrammetry using a rotating Surface. [en línea] <https://docs.unity3d.com/uploads/ExpertGuides/Creating_Assets_with_Photogrammetry_using_a_rotating_surface.pdf?_ga=2.198839599.1841641142.1560343316-1571804154.1555436954>[consulta: 12 abril 2019]
- ARIALROBOTICS. 2013. Secrets of UAV photomapping-Pteryx UAV. [en línea] <<https://www.yumpu.com/en/document/view/8979635/secrets-of-uav-photomapping-ptyryx-uav>>[consulta: 24 mayo 2019]
- MAURICIO. 2017 metodología RAD. [en línea] <<https://sisingblog.wordpress.com/2017/04/03/metodologia-rad/>>[consulta: 25 mayo 2019]

- GLOBALMEDITERRANEAGEOMATICA. 2017 Fotogrametría y realismo en los videojuegos. [en línea]<<https://www.globalmediterranea.es/fotogrametria-y-realismo-en-videojuegos/>>[consulta: 3 septiembre 2019]
- MERINO, M. Pérez, J. 2013. Definición de texturas. [en línea] <<https://definicion.de/textura/>>[consulta: 12 septiembre 2019]
- GONZALES, A. 2017. Morfología de imágenes binarias. [en línea] <https://issuu.com/alexgonzalez482/docs/ag_1_procesamiento_de_imagenes> [consulta: 12 septiembre]
- MICHAEL, E. 2017. Open CV. [en línea] <<http://acodigo.blogspot.com/p/tutorial-opencv.html>>[consulta: 12 septiembre 2019]
- RUBI3D. 2017. ¿Qué es una textura? [en línea] <<https://rubi3d.com/index.php/2017/12/13/48-que-es-una-textura/>>[consulta: 21 octubre 2019]
- BRENDAN, G. 2017. Escaneo facial. [en línea]<<https://www.artstation.com/artwork/8OPrE>>[consulta: 27 enero 2020]
- MEDICOABOGADO. 2016. Distribucion t student.[en línea] <<https://estadisticaeninvestigacion.wordpress.com/distribucion-t-de-student/>>[consulta: 12 febrero 2020]
- JORGE, F. 2014. Tipos de filtro. [en línea]<<https://sites.google.com/site/megafoniaysonorizacionjorge/10-megafonia-y-sonorizacion/4-equipos-y-transductores-electroacusticos-amplificadores-y-altavoces/4-6-filtros-de-cruce/4-6-2-tipos-de-filtro>>[consulta: 27 septiembre 2019]
- JAN. 2015. Filtros sobre una imagen. [en línea] <https://www.researchgate.net/figure/Figura-4-Producto-de-la-funcion-de-filtrado-por-la-imagen-ambos-en-el-dominio-de-la_fig1_39195331>[consulta: 4 septiembre 2019]

ANEXOS

• ANEXO A – TABLA T-STUDENT

Puntos de porcentaje de la distribución t



Ejemplo

Para $\phi = 10$ grados de libertad:

$$P[t > 1.812] = 0.05$$

$$P[t < -1.812] = 0.05$$

α Γ	0,25	0,2	0,15	0,1	0,05	0,025	0,01	0,005	0,0005
1	1,000	1,376	1,963	3,078	6,314	12,706	31,821	63,656	636,578
2	0,816	1,061	1,386	1,886	2,920	4,303	6,965	9,925	31,600
3	0,765	0,978	1,250	1,638	2,353	3,182	4,541	5,841	12,924
4	0,741	0,941	1,190	1,533	2,132	2,776	3,747	4,604	8,610
5	0,727	0,920	1,156	1,476	2,015	2,571	3,365	4,032	6,869
6	0,718	0,906	1,134	1,440	1,943	2,447	3,143	3,707	5,959
7	0,711	0,896	1,119	1,415	1,895	2,365	2,998	3,499	5,408
8	0,706	0,889	1,108	1,397	1,860	2,306	2,896	3,355	5,041
9	0,703	0,883	1,100	1,383	1,833	2,262	2,821	3,250	4,781
10	0,700	0,879	1,093	1,372	1,812	2,228	2,764	3,169	4,587
11	0,697	0,876	1,088	1,363	1,796	2,201	2,718	3,106	4,437
12	0,695	0,873	1,083	1,356	1,782	2,179	2,681	3,055	4,318
13	0,694	0,870	1,079	1,350	1,771	2,160	2,650	3,012	4,221
14	0,692	0,868	1,076	1,345	1,761	2,145	2,624	2,977	4,140
15	0,691	0,866	1,074	1,341	1,753	2,131	2,602	2,947	4,073
16	0,690	0,865	1,071	1,337	1,746	2,120	2,583	2,921	4,015
17	0,689	0,863	1,069	1,333	1,740	2,110	2,567	2,898	3,965
18	0,688	0,862	1,067	1,330	1,734	2,101	2,552	2,878	3,922
19	0,688	0,861	1,066	1,328	1,729	2,093	2,539	2,861	3,883
20	0,687	0,860	1,064	1,325	1,725	2,086	2,528	2,845	3,850
21	0,686	0,859	1,063	1,323	1,721	2,080	2,518	2,831	3,819
22	0,686	0,858	1,061	1,321	1,717	2,074	2,508	2,819	3,792
23	0,685	0,858	1,060	1,319	1,714	2,069	2,500	2,807	3,768
24	0,685	0,857	1,059	1,318	1,711	2,064	2,492	2,797	3,745
25	0,684	0,856	1,058	1,316	1,708	2,060	2,485	2,787	3,725
26	0,684	0,856	1,058	1,315	1,706	2,056	2,479	2,779	3,707
27	0,684	0,855	1,057	1,314	1,703	2,052	2,473	2,771	3,689
28	0,683	0,855	1,056	1,313	1,701	2,048	2,467	2,763	3,674
29	0,683	0,854	1,055	1,311	1,699	2,045	2,462	2,756	3,660
30	0,683	0,854	1,055	1,310	1,697	2,042	2,457	2,750	3,646
40	0,681	0,851	1,050	1,303	1,684	2,021	2,423	2,704	3,551
60	0,679	0,848	1,045	1,296	1,671	2,000	2,390	2,660	3,460
120	0,677	0,845	1,041	1,289	1,658	1,980	2,358	2,617	3,373
∞	0,674	0,842	1,036	1,282	1,645	1,960	2,326	2,576	3,290

DOCUMENTACIÓN