Sistema de Visualización tridimensional de imágenes con uso de marca flotante

Autor: Jesús Ignacio Becerra.
Tutor: Prof. Domingo Hernández
Cotutor: Prof. Luís Jáuregui

Proyecto de Grado presentado ante la ilustre Universidad de Los Andes como requisito final para optar al titulo de Ingeniero de Sistemas.

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA DE SISTEMAS

Mérida, 15 de Septiembre de 2006

INGENIERIA
UNIVERSIDAD DE LOS ANDES
MERIDA - VENEZUELA

Escuela de Ingeniería de Sistemas

PROYECTO DE GRADO
CALIFICACIÓN FINAL

Título del Proyecto de Grado: "**SISTEMA DE VISUALIZACIÓN TRIDIMENSIONALES DE IMÁGENES CON USO DE MARCA FLOTANTE**".

Bachiller: **JESÚS IGNACIO BECERRA** **C.I. 15.293.115**

1) Calificación del (de la) Profesor(a) Tutor(a):	20	(20%)
2) Calificación del manuscrito final:	20	(40%)
3) Calificación de la defensa oral:	20	(40%)
Calificación final:	20	**puntos**

Los suscritos miembros del Jurado asignan como calificación final del Proyecto de Grado la nota de:

20 (puntos) Números — **Veinte (puntos)** Letras

Observaciones:

Prof. Domingo Hernández
Tutor

Prof. Luis Jáuregui
Cotutor

Prof. Víctor Bravo Bravo
Jurado

Prof. Wladimir Rodríguez
Jurado

Mérida, 09 de Octubre de 2006

DEDICATORIA

A mi madre Ramona Becerra por brindarme su apoyo, fortaleza y ser una fuente inagotable de amor incondicional.

A mi abuela Ilvia Rosa por cuidarme y confiar en mí en todo momento, que Dios te bendiga abuelita.

A mi segunda mama Alba Avendaño por velar de mi y acobijarme en su hogar como otro hijo, que Dios te cuide Albita.

A Dios todopoderoso que me ha permitido estar con mis seres queridos.

AGRADECIMIENTOS

A mi mejor amigo Einstein Briceño, por estar conmigo en todo momento, te deseo el mejor de los exitos “hermano”.

A mis amigos Marlon Torres y Juan Miguel Palma por confiar en mí y brindarme de su apoyo.

A todas esas personas que estuvieron y están conmigo, y que para bien o para mal, aportaron algo para este proyecto.

INDICE DE CONTENIDO

Capítulo 1
1.1 Antecedentes. 7
1.2 Definición del Problema. 8
1.3 Objetivos Generales. 8
1.4 Objetivos Específicos. 8
1.5 Metodología 9

Capítulo 2
2.1 Paralaje 10
2.1.1 La percepción del relieve 10
2.1.2 Visión Monocular 10
2.1.3 Visión Binocular 12
2.1.4 La visión estereoscópica. 12
2.1.4.1 Acomodación. 13
2.1.4.2 Convergencia. 13
2.1.4.3 Paralaje. 13
2.1.5 Visión estereoscópica artificial. 15
2.1.5.1 Formas de visión estereoscópica artificial. 16
2.1.6 El paralaje estereoscópico. 19
2.1.7 Principio de la marca flotante 20
2.2 Anaglifos 22
2.2.1 Definición 22
2.2.2 Construcción 23
2.3 Qt Toolkit. 28
2.3.1 Señales y Ranuras 29
2.3.1.1 Señales 31
2.3.1.2 Ranuras 31
2.3.2 *Meta Object System.* 31
2.4 El Modelo en Espiral. 32
2.5 *Unified Modeling Language* (UML). 34

Capítulo 3
3.1 Modelado Funcional del Sistema 38
3.1.1 Requerimientos Funcionales 38
3.1.2 Requerimientos No Funcionales 39
3.1.3 Diagramas de casos de uso 40
3.1.3 Desarrollo 42

3.1.3.1 Planificación y Análisis Inicial de Requisitos 42
3.1.3.2 Prototipo inicial 42
3.1.3.3 Evaluación 42
3.1.3.4 Segundo Prototipo 43
3.1.3.5 Evaluación 43
3.1.3.6 Producto Final 44
3.1.4 Diagramas de Clases 44
3.1.4 Diagramas de Interacción 45
3.1.5 Diagrama de Componentes 46
3.2 El Sistema Programado 47

Capítulo 4
4.1 Pruebas de Caja negra 55
4.2 Pruebas de Caja Blanca 58
4.2.1 Ambiente de Prueba 59
4.2.2 Experimento 59

Capítulo 5
5.1 Conclusiones 63
5.2 Recomendaciones 64

BIBLIOGRAFIA 65

ANEXO A 66
ANEXO B 77
ANEXO C 93

CAPITULO 1
INTRODUCCION

Este Capítulo es un enfoque general de los antecedentes relacionados con el proyecto, así como del problema a resolver, los objetivos, alcances y resultados esperados.

Se describe brevemente la metodología a utilizar y la estructuración de los capítulos contenidos en este proyecto.

1.1 Antecedentes.

Desde sus inicios, la fotogrametría se ha convertido en la herramienta indispensable en la producción de la base cartográfica de todos los países del mundo; de hecho, la mayoría de la cartografía topográfica de nuestro planeta ha sido realizada por medio de esta disciplina.

Si bien la fotogrametría tuvo su inicio en el levantamiento de fachadas arquitectónicas y plantas de edificios, mediante el uso de fotografías terrestres, pronto se utilizaron las fotografías aéreas para el levantamiento de la cartografía de base, lo que le dio el tremendo auge que ha mantenido hasta nuestros días. Esta capacidad de cartografiado de base la convierte también en la fuente primigenia de información para la cartografía temática y para los sistemas de información geográficos. Como consecuencia de la utilización de la fotografía aérea, se desprendió de la fotogrametría la disciplina de la fotointerpretación, la cual comparte sus fundamentos básicos con la fotogrametría aérea.

A partir de los años ochenta, el desarrollo acelerado de la computación, condujo al establecimiento de la teledetección como consecuencia lógica de la evolución de la

fotointerpretación, así como al desarrollo de técnicas de tratamiento computarizado de imágenes digitales y al desarrollo de la visión por computadora.

Actualmente, con el apoyo de la computación, la fotogrametría se ha convertido en una disciplina indispensable en el campo de la cartografía, a la vez que aumenta el número de sus usuarios debido a que los equipos fotogramétricos de elevado costo, están siendo desplazados por programas de precio menor, o por programas desarrollados por los mismos usuarios.

1.2 Definición del Problema.

Debido al alto costo de los equipos digitales para fotogrametría, se plantea el desarrollo de un sistema programado para la visualización tridimensional de imágenes con uso de marca flotante.

1.3 Objetivos Generales.

Desarrollar un sistema capaz de crear una imagen anaglifo con 2 fotografías aéreas de un terreno y almacenar coordenadas tridimensionales del mismo a través del principio de la marca flotante.

1.4 Objetivos Específicos.

- Desarrollar un sistema donde puedan desplegarse simultáneamente dos imágenes de una misma área y a su vez sobreponerse una sobre otra, pudiéndose variar el porcentaje de sobre-posición, según los principios de construcción del anaglifo.

- Asociar el movimiento de la rueda de desplazamiento vertical del Mouse (scroll), con la separación horizontal entre las imágenes.

- Registrar y almacenar coordenadas x, y, z de las fotografías aéreas a través de la imagen anaglifo.

1.5 Metodología

- Desarrollo del proyecto: Modelo en espiral.

  El proyecto es creado a través de múltiples repeticiones del proceso del ciclo de vida, que considera 4 fases: Planificación, análisis de riesgo, ingeniería y evaluación de proyectos.

- Modelado del Sistema: Lenguaje de Modelado Unificado, mejor conocido como UML**.**

CAPITULO 2
MARCO TEORICO

En este capítulo se presentan los principales conceptos que se manejan en este proyecto, así como definiciones e información alterna que sirve de apoyo para la comprensión de los mismos y la descripción de las herramientas utilizadas para la implementación del sistema programado.

2.1 Paralaje

Las definiciones a continuación sirven de base para el entendimiento del paralaje, concepto primordial en este proyecto.

2.1.1 La percepción del relieve

El sistema visual humano posee la capacidad de percibir el relieve mediante el uso de mecanismos tanto psicológicos como fisiológicos, según la visión sea monocular o binocular, respectivamente. Así pues, la percepción del relieve presenta un doble aspecto: en primer lugar monocular, ya que estamos en capacidad de reconstituir el espacio a partir de la visión obtenida a través de un solo ojo, y siguiente y con especial relevancia binocular, ya que al mirar simultáneamente con ambos ojos se obtiene la visión estereoscópica o esteropsis, la cual nos permite apreciar realmente el relieve, en forma independiente de la visión monocular.

2.1.2 Visión Monocular

La simple visión monocular nos da la visión de relieve; cuando se observa un cuadro, una fotografía una pantalla de cine o de televisión, no se tiene ninguna dificultad a

ubicar los objetos, unos respecto a otros. No existe en general ninguna ambigüedad en darles sus posiciones relativas. Para ello, se hace uso de mecanismos psicológicos, que tienen su origen en las numerosas informaciones que el sentido de la visión posee, producto de la experiencia. Entre estos tenemos:

- El tamaño relativo de los objetos: si un objeto A que posee las mismas dimensiones que un objeto B, se ve de tamaño mayor, entonces el objeto A está mas cerca; por supuesto, es necesario previamente conocer el tamaño que tienen estos objetos. Esta propiedad es muy utilizada en la industria del cine, en la creación de maquetas.

- El grado de detalles o nitidez que posee un objeto: se asocia la cercanía con un mayor grado de detalle.

- El desplazamiento relativo de los objetos: cuando estamos desplazándonos los objetos más cercanos se mueven con mayor velocidad que los más alejados. Esta propiedad es vital para los animales depredadores.

- La perspectiva: es el origen esencial de la reconstrucción del relieve, a partir de una vista monocular. Se conoce por la experiencia visual, que el espacio se representa en el plano mediante ciertas reglas geométricas, que constituyen las reglas que crean la ilusión de la perspectiva.

- Decoloración de los objetos. Asociada a la perspectiva está la decoloración de los objetos, que ocurre en forma directamente proporcional a la distancia.

- El recubrimiento de un objeto por el otro: si el objeto A recubre el objeto B, entonces el A está más cercano.

- Las sombras: determinan principalmente el relieve de elementos verticales. Cuando la sombra de un objeto está por debajo del mismo, es porque sobresale del plano vertical en el que está contenido. Cuando la sombra se encuentra por encima, entonces el plano está por delante del objeto. También, y a semejanza del recubrimiento parcial de un objeto por otro, aquellos elementos de la imagen monocular cubiertos parcial o totalmente por la sombra de otro elemento, estarán por detrás de éste.

- Diferencias en el enfoque del ojo, que son un factor fisiológico complementario en la determinación de la cercanía de los objetos.

2.1.3 Visión Binocular

En la visión binocular ambos ojos observan la misma escena, pero desde un punto de vista ligeramente diferente, ya que están separados por una distancia que es, en promedio, de 65mm, denominada ***distancia interpupilar***. Esta diferencia de posición hace que cada ojo posea una vista ligeramente diferente de la escena visada; estas diferencias son procesadas por el sentido de la visión y el resultado del mismo es la sensación de relieve, la cual puede variar ligeramente entre los individuos, dependiendo de su distancia interpupilar y de las características propias de su visión.

Algunos individuos no tienen capacidad de visión estereoscópica, aunque pueden ver por ambos ojos. La razón de ello estriba en defectos de la visión, como por ejemplo, el estrabismo.

2.1.4 La visión estereoscópica.

Tal como se afirmó anteriormente, el aspecto más relevante de la visión humana es su capacidad de percibir el relieve o profundidad. Esta visión estereoscópica se logra mediante mecanismos fisiológicos siendo éstos los siguientes:

2.1.4.1 Acomodación.

Es la tensión del músculo que cambia la curvatura del cristalino, y por ende, la distancia focal del mismo. Como la tensión se hace mayor y por lo tanto notable, para enfocar objetos cercanos, este mecanismo sólo es útil para cortas distancias.

2.1.4.2 Convergencia.

Es el ángulo formado por la intersección de los ejes ópticos de los ojos. Este mecanismo es efectivo a cortas distancias por las mismas razones del mecanismo anterior.

2.1.4.3 Paralaje.

Si un lápiz es mantenido a 30cm de distancia de los ojos, frente a un determinado objeto ubicado en el fondo, y haciendo énfasis en éste, se cierran y abren alternativamente ambos ojos, en rápida sucesión, se tiene la sensación de que el lápiz se desplaza con respecto al objeto. Si se enfoca ahora alternativamente al lápiz y al objeto, se tiene la impresión de que los objetos se encuentran a distancias diferentes (de hecho, cuando se enfoca el objeto, el lápiz se ve doble). Un esquema semejante al de la situación aquí descrita se puede observar en la figura 5.3, donde el objeto “A” se encuentra más cercano que el objeto “B”

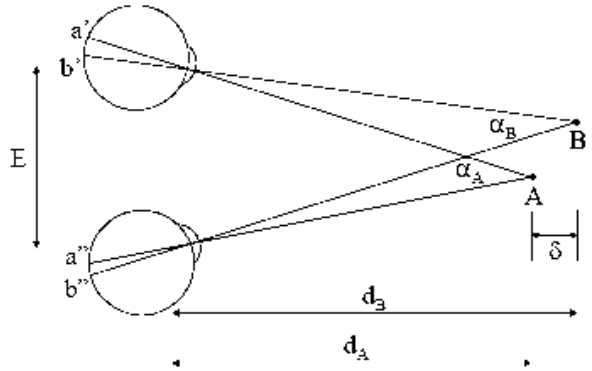

Figura Nº 1. Esquema de las trayectorias visuales y su proyección en la retina.

La diferencia en la retina, de los dos ángulos de convergencia, es el paralaje relativo entre los dos objetos, es decir a'-a" y b'-b". Esta diferencia en paralaje es el factor más importante para la percepción del relieve. Al igual que para la estimación de distancia, su limite está en un minuto de arco. Esto significa que la visión estereoscópica puede apreciar diferencias en profundidad de hasta 0,075mm. Este valor proviene no del ángulo en sí, sino del espaciamiento entre los bastoncitos en la retina. El ángulo de convergencia es leído en la retina mediante las distancias entre la posición de la imagen del objeto con respecto a la posición de la fóvea, que es el origen de coordenadas de cada ojo. Los límites para la visión estereoscópica se encuentran a 150mm para el inferior, ya que a esta distancia no se consigue la fusión del objeto visado por ser el límite de la convergencia y el enfoque, y 200m para el superior, debido limite en la resolución angular del ojo.

Cuando los rayos de luz que proceden de un punto de un objeto llegan a los ojos, están separados por un ángulo, llamado ángulo paraláctico, que decrece a medida que la distancia del objeto aumenta. Este ángulo, en radianes, puede expresarse como:

$$\alpha = \frac{E}{d}$$

Donde E es la distancia interpupilar o base de los ojos, que varía de 60mm a 70mm en los humanos, y d es la distancia desde la base de los ojos hasta el objeto visado. Si tomamos la distancia interpupilar promedio como E = 65mm, y $\alpha = 1'$, tendremos que d = 200m, aproximadamente.

2.1.5 Visión estereoscópica artificial.

En fotogrametría se hace uso de la visión estereoscópica para la observación de los pares estereoscópicos, donde los paralajes, relativamente grandes, son producidos mediante los métodos de toma de vistas.

Ya que cada ojo observa en su retina direcciones hacia puntos, se puede reemplazar la visión tridimensional del entorno natural superponiendo delante de cada ojo una fotografía que contenga las mismas direcciones dadas por la visión natural del entorno, ya que al existir los mismos paralajes, se producirán en la visión los mismos mecanismos de percepción que dan origen a la visión estereoscópica. La observación independiente de cada ojo sobre su respectiva fotografía da origen entonces a un modelo tridimensional del terreno en la mente del observador.

Los requisitos para realizar la visión estereoscópica son los siguientes:

- A cada ojo hay que presentarle una imagen que cubra un área común y que presente paralaje con respecto a la que observa el otro ojo.

- Los rayos visuales hacia puntos homólogos deben intersectarse en el espacio de dos en dos. Esto significa que la observación debe realizarse según planos epipolares, es decir, que los rayos epipolares homólogos estén sobre la misma recta.

- Los puntos más lejanos de ambas imágenes no han de tener una separación mayor que la distancia interpupilar del observador, o a la base de visión del aparato utilizado.

La relación entre la acomodación y la convergencia es muy importante en la visión estereoscópica. Cuando los ojos enfocan un objeto cercano, convergen a su vez automáticamente, de forma que las líneas de visión intersectan al objeto. Cuando enfocamos al infinito, las líneas de visión son paralelas. Cuando se observa un par de fotografías estereoscópicas a través de los estereoscopios, los ojos convergen para adaptarse a la distancia de observación, pero la visión debe realizarse con las líneas de visión paralelas, para adaptarse a los ejes ópticos del aparato. Esta contradictoria forma de relación, opuesta a lo acostumbrado por el sentido de la visión, toma algo de práctica para ser asimilado, y por lo tanto observar los pares de fotos con comodidad.

2.1.5.1 Formas de visión estereoscópica artificial.

Existen tres formas de crear el efecto tridimensional, en base a los ejes de convergencia de la visión, clasificándose en:

- Ejes cruzados
- Ejes paralelos
- Ejes convergentes

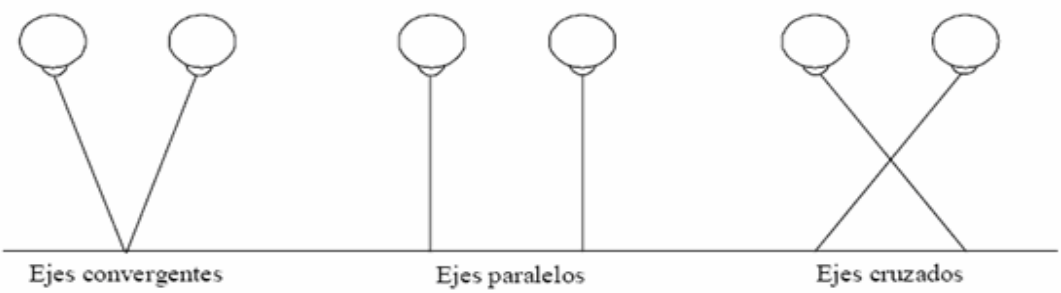

Figura Nº 2. Ejes de visión binocular

Ejes cruzados: este tipo de observación si bien es factible, no se utiliza en la práctica, salvo en casos contados, debido a que el tipo de dificultad que posee, lo hace accesible solamente a los más experimentados, cuando se realiza al ojo desnudo. Anteriormente se utilizó en algunos aparatos restituidores, que poseían el llamado "paralelogramo de Zeiss", hoy día en desuso.

Formas de visión mediante ejes paralelos: esta modalidad es la mas utilizada actualmente, tanto en los estereoscopios como en la mayoría de los aparatos de restitución, y en la cual la óptica obliga a los ojos a mantener los ejes de la visión paralelos entre si.

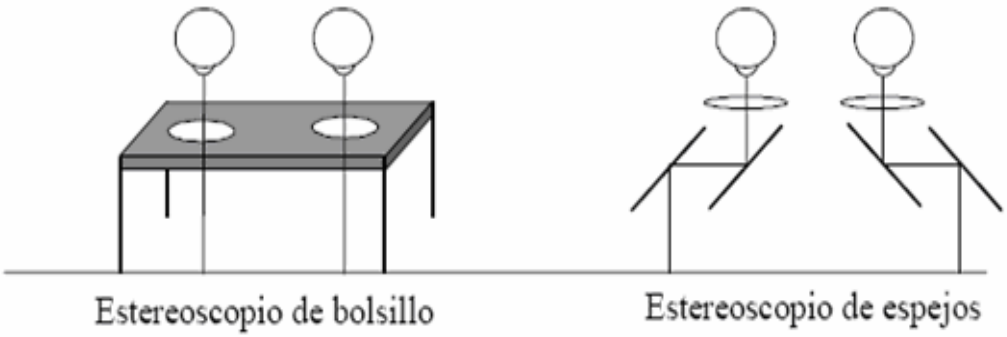

Figura Nº 3. Visión por ejes paralelos en estereoscopio

Formas de visión mediante ejes convergentes: en este caso la estereoscopia se consigue por medios que transmiten a cada ojo su respectiva imagen, impidiendo la visualización de la otra imagen.

La visión mediante ejes convergentes puede realizarse mediante tres diferentes formas posibles para controlar la visualización:

- Anaglifos: utilizan filtros de colores primarios (azul y rojo o verde y rojo), para la proyección de la diapositiva y para su observación.

- Filtros polarizados: utiliza este tipo de filtros, con una rotación de 90ª, entre los filtros de la izquierda y los de la derecha.

- Obturadores sincronizados: esta forma hace uso del principio de retención de una imagen en la retina durante un lapso de tiempo (alrededor de 1/30seg). El obturador del proyector izquierdo deja pasar la luz, al igual que el obturador de observación izquierdo, mientras que los de la derecha permanecen cerrados; una fracción después, se abren los derechos y se cierran los izquierdos.

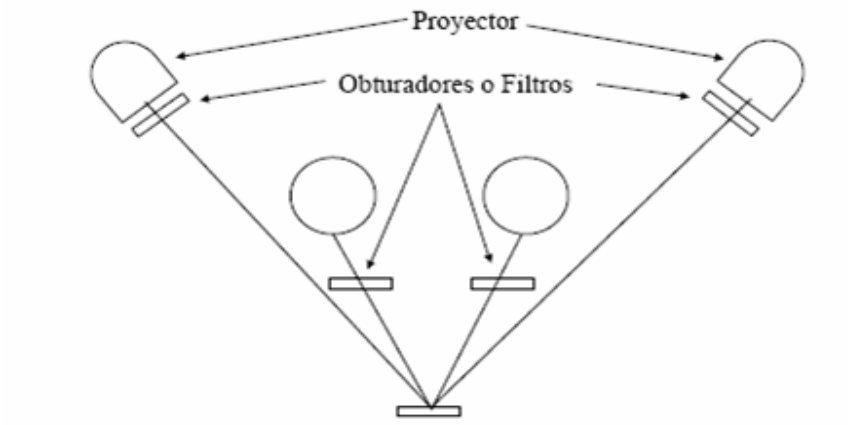

Figura Nº 4. Visión por ejes convergentes.

2.1.6 El paralaje estereoscópico.

Paralaje es el cambio de posición de la imagen de un punto en dos fotografías sucesivas, debido al cambio de posición de la cámara en el momento de la toma. A partir de esta definición y de las **figuras Nº 5 y 6**, se puede deducir una formula que permita calcular las diferencias de altura.

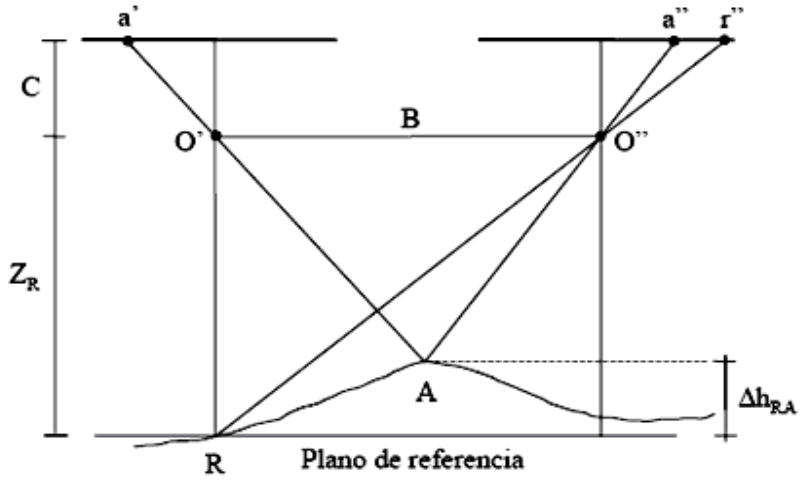

Figura Nº 5. Relación paralaje-altura

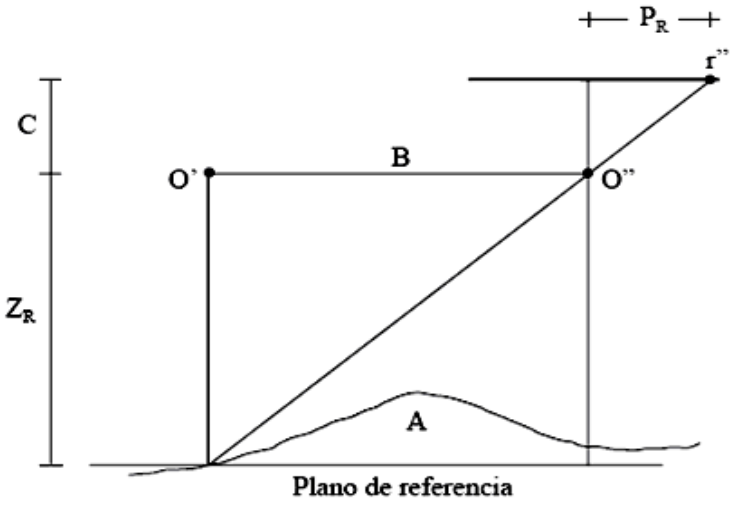

Figura Nº 6. Relación paralaje-altura, punto R

Por relación de triangulos se obtiene de la **figura Nº 5**, la ecuación Nº 1:

$$\frac{Z_R}{B} = \frac{C}{P_R}$$

De igual manera de la **figura Nº 6**, se obtiene la ecuación Nº 2:

$$\frac{Z_A}{B} = \frac{C}{P_A}$$

De ambas figuras ademas se obtienen las ecuaciones Nº 3 y Nº 4:

$$\Delta Z_{RA} = Z_A - Z_R \text{ (negativo)}$$

$$\Delta h_{RA} = h_A - h_R \text{ (positivo)}$$

Ya con las ecuaciones 1, 2, 3 y 4 se obtiene la ecuación definitiva:

$$\Delta hRA = \frac{Z_R \times AP_{RA}}{P_R + AP_{RA}}$$

2.1.7 Principio de la marca flotante

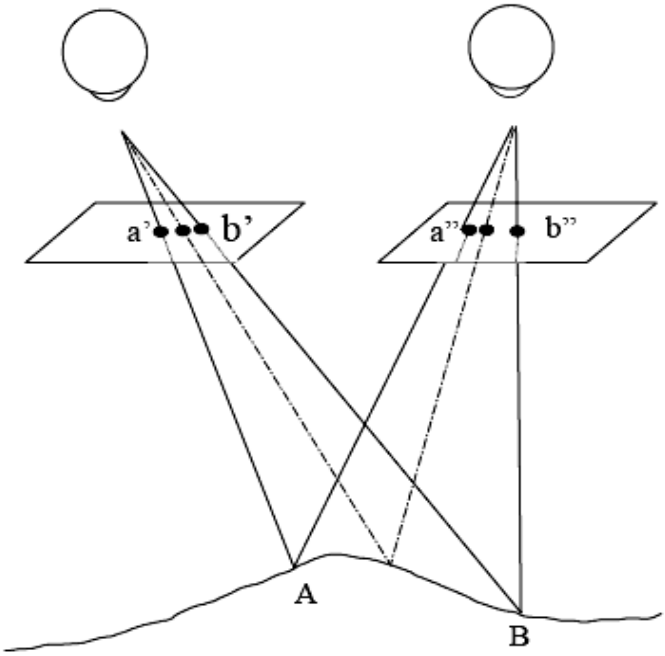

Figura Nº 7. Esquema de la marca flotante

El principio de la marca flotante consiste en colocar dos marcas idénticas, una sobre cada fotografía. Cuando cada una de ellas se encuentre sobre un punto homologo en la respectiva fotografía, se verán entonces como un solo punto en contacto con el terreno. Si una de las marcas se acerca o se aleja respecto a la otra, se tendrá la impresión de que el punto subirá o bajará con respecto al terreno.

Los paralajes de los puntos pueden ser medidos en forma estereoscópica, con la ventaja de la rapidez y de la precisión, mediante el uso de la marca flotante.

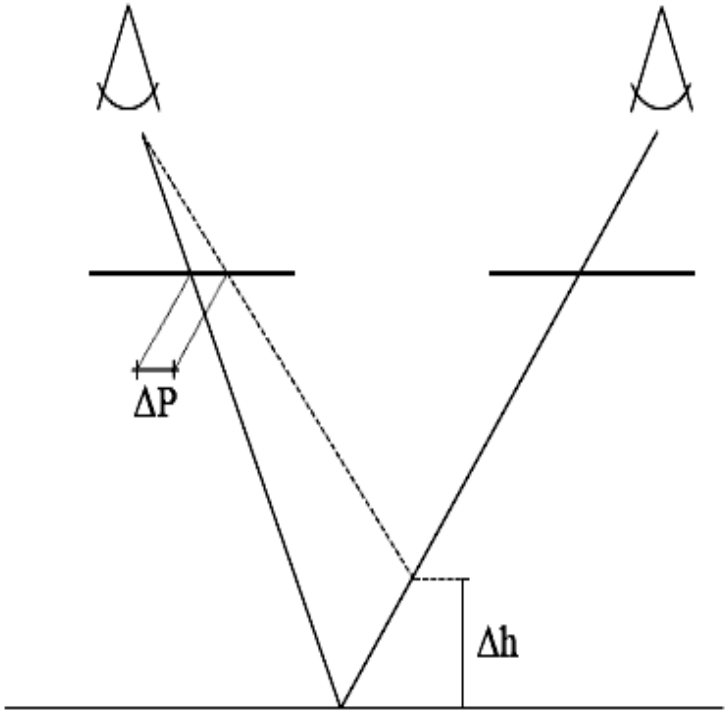

Figura Nº 8. Relación entre paralaje y altura en la marca flotante

2.2 Anaglifos

El anaglifo constituye el núcleo del sistema de visualización de imágenes tridimensionales

2.2.1 Definición

Su nombre deriva del griego αναγλυ.ος (ανα alto y γλυ.ο esculpir); es un par estereoscópico, correctamente orientado y montado, donde las imágenes se superponen. Para que cada ojo vea su respectiva fotografía, las mismas son impresas en colores complementarios (rojo y verde, o rojo y azul) y son observadas a través de filtros de un color complementario al usado en la impresión. Con esto se consigue que cada ojo observe su respectiva imagen, ya que el filtro delante de cada ojo deja pasar la luz proveniente de la imagen correspondiente, haciendo negra la otra.

Según el *Diccionario Enciclopédico Larousse*, 1990:

> La estereoscopia por anaglifos en el arte fotográfico *«es un sistema especial de fotografías estereoscópicas de dos imágenes, una con coloración roja y la otra azul verdosa, que se superponen, ligeramente desplazadas o desfasadas, en la copia fotográfica»*

2.2.2 Construcción

En la **figura Nº 9** se muestra de una forma muy sencilla un cubo cromático donde tendrían cabida todos los colores que se pueden mostrar partiendo de los primarios: el rojo, el verde y el azul.

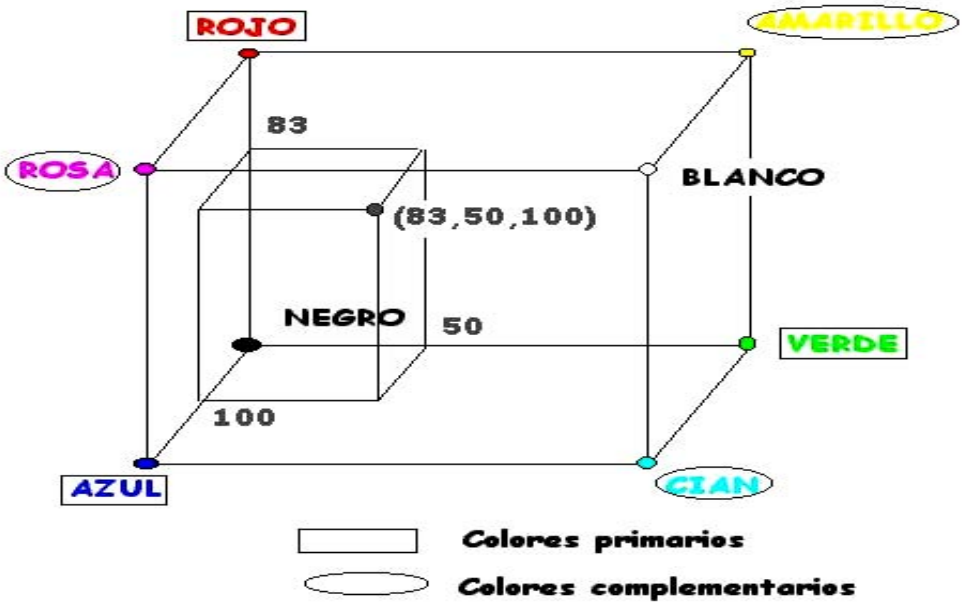

Figura Nº 9. Cubo cromático

Si se empezara en el negro (ausencia de color) el moverse por las aristas del cubo hacia el rojo, el azul o el verde se traduciría en un aumento de la intensidad de ese color. Si la dirección del movimiento es en sentido diagonal, hacia el blanco, se estaría atravesando la escala de los grises (que se caracterizan porque todos los componentes tienen las misma "cantidad" de rojo, azul y verde). Cuanto más posibles divisiones se puedan tener en cada eje, más tonalidades se pueden generar. Así, un fichero jpg por ejemplo se caracteriza por poder guardar por cada pixel 24 bits de color repartiendo 8 bits para cada uno de los colores (rojo, verde, azul), lo que en sistema decimal quiere decir que podemos variar cada uno de ellos en una gama entre 0 y 255.

Cuando se quiere representar un color determinado, solo es necesario especificar una cifra que puede variar entre 0 y 255 para cada color, dicho de otra manera, un color viene dado por una cifra para el rojo, otra para el azul y por ultimo una cifra para el verde. A esta representación de colores se le denomina RGB por las siglas *Red*, *Green*, *Blue.* Este modelo es muy utilizado en informática.

En el cubo cromático son complementarios cada vértice con su opuesto. De ahí que los colores complementarios serían los pares: (Negro, Blanco), (Rojo, Cian), (Azul, Amarillo), (Verde, Rosa). Analizando un par de ellos, por ejemplo el (Rojo, Cian). Su representación en el cubo sería ((255,0,0), (0,255,255)). La suma de cada uno de sus componentes dará como resultado, el (255, 255, 255), es decir, el blanco. Esto también ocurre con los demás. La suma de las partes es igual al total. Un anaglifo se basa en esta propiedad. Unas gafas Rojo-Cian típicas están formadas por dos cristales con colores complementarios. La idea es simple. Se quiere ver una imagen, mezcla de otras 2, perdiendo la mínima información. Lo que un ojo no ve, lo ve el otro. El ojo izquierdo ve sólo la información de la foto izquierda y el ojo derecho solo la información de la foto derecha.

Aclarado el punto de la paleta de colores RGB, esencial para la construcción de un anaglifo, el proceso a continuación muestra la generación de un anaglifo a partir de 2 fotografías, el proceso en si es bastante sencillo.

Un punto importante para hacer una imagen tridimensional es fotografiar un objeto y/o paisaje desde dos puntos de vista diferentes (par estereoscópico) con una separación horizontal entre fotografía y fotografía que dependerá de la distancia a la que se encuentra el objeto a fotografiar. El motivo a fotografiar no debe sufrir variación en su estado durante el tiempo que dure la sesión fotográfica. Esto significa que aquellos motivos que presenten movimiento no son adecuados para su representación tridimensional.

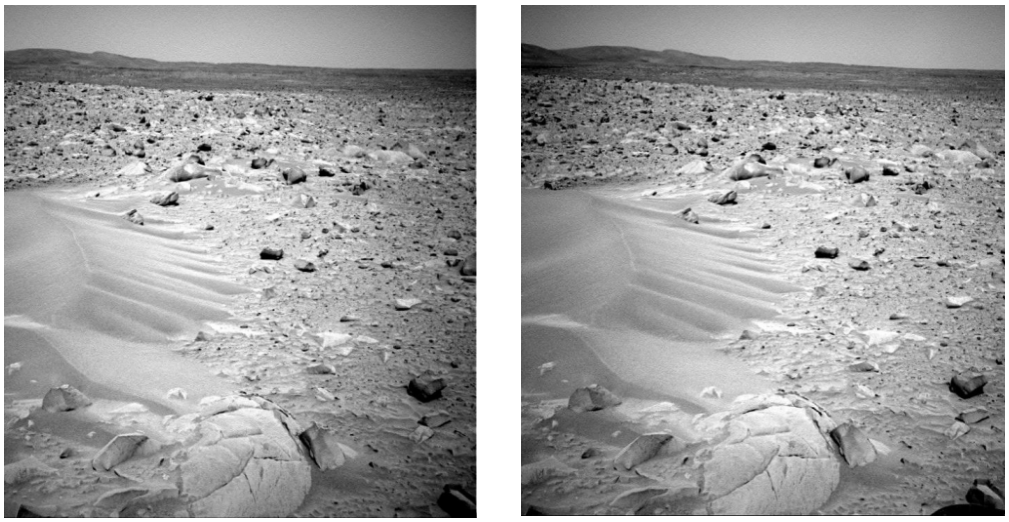

Figura Nº 10. Par Estereoscópico

Una vez obtenido el par estereoscópico comienza el proceso de creación del anaglifo. Dado que cada punto (píxel) que conforma cada fotografía esta compuesto de tres componentes (Rojo, Verde, Azul) para representar un color determinado, a la imagen izquierda se le suprime toda la información de verde y azul, dejando sólo la

componente rojo, es decir, para todos los puntos $(R_i, V_i, A_i) \rightarrow (R_i, 0, 0)$. El resultado se muestra en la **figura Nº 11.**

Figura Nº 11. Imagen izquierda modificada

A la fotografía derecha se le aplica un procedimiento similar, se suprime toda la información del componente rojo, dejando sólo los componentes restantes $(R_i, V_i, A_i) \rightarrow (0, V_i, A_i)$. El resultado se muestra en la figura siguiente:

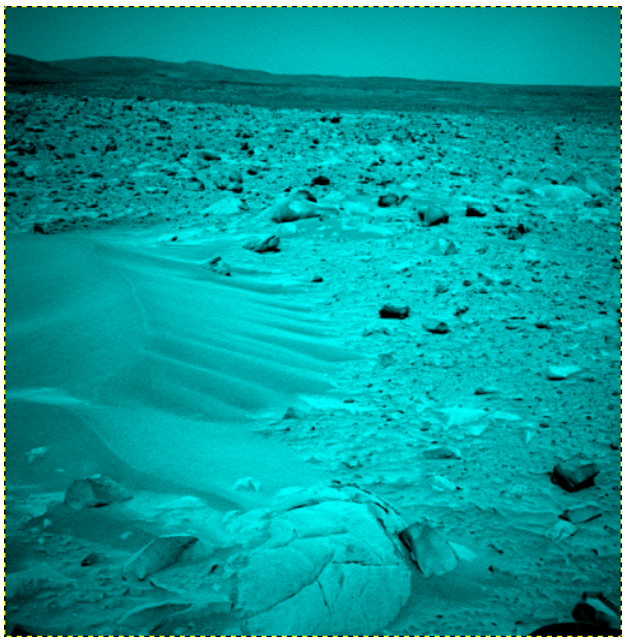

Figura Nº 12. Imagen derecha modificada

Obtenidas las fotografías modificadas, ya solo resta combinarlas. La combinación se logra sumando cada uno de los componentes de color que componen cada píxel de ambas fotografías, dicho esto de otra manera:

$$(R_i,0,0)_{izquierda} + (0,V_i,A_i)_{derecha} = (R_i,V_i,A_i)_{anaglifo}$$

La imagen resultante de la combinación de ambas fotografías es la imagen anaglifo y es mostrada en figura siguiente:

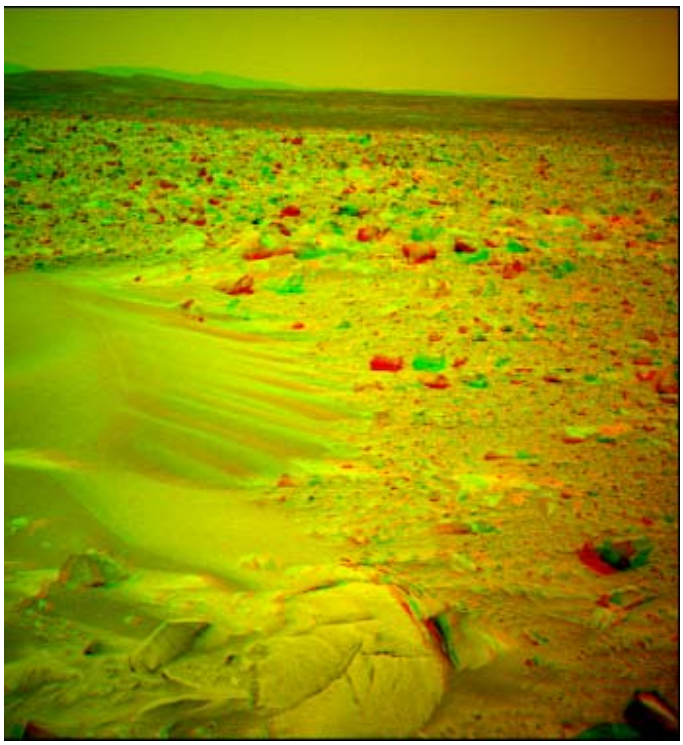

Figura Nº 13. Anaglifo resultante de la combinación

2.3 *Qt Toolkit*.

Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario usando un enfoque "Una vez escrito, compilalo donde quieras". Fue creada por la compañía noruega Trolltech. Es utilizada en KDE, un entorno de escritorio para sistemas operativos como linux o FreeBSD, entre otros. Utiliza el lenguaje de programación C++ de forma nativa y además existen bindings para C, Python, Perl y Ruby entre muchos.

El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML y una multitud de otros para el manejo de ficheros, además de estructuras de datos tradicionales.

Qt añade estas características a C++:

- Un poderoso mecanismo para la comunicación entre objetos llamado señales (*signa*ls) y ranuras (*slots*).
- Un manejo y filtrado de eventos muy eficaz.
- Traducción contextual de cadenas para internacionalización
- Sofisticado temporizador controlado de intervalos (*timers*) que hace posible integrar elegantemente cualquier tarea en un evento-controlado GUI.
- Árboles de Objetos jerárquicos que organizan propiedades de objetos de manera natural.
- Punteros protegidos, QGuardedPtr, que son automáticamente asignados a 0 cuando la referencia al objeto es destruida, a diferencia de los punteros normales de C++ que llegan a convertirse en "punteros perdidos o suspendidos" cuando sus objetos son destruidos.

Cualquiera de estas características de Qt están implementadas con técnicas del estándar C++ en el legado de QObject. Demás mecanismos de comunicación y el sistema dinámico de propiedades, requieren del *Meta Object System* provisto por el propio *Meta Object Compiler* (moc) de Qt.

El *Meta Object System* es una extensión de C++ que hace al lenguaje mejor adaptable para la programación de componentes GUI. Aunque las plantillas (templates) pueden ser usadas para extender C++, El *Meta Object System* proporciona beneficios que no pueden ser logrados con plantillas, como una sintaxis intuitiva, simple de usar y facil de leer, alta flexibilidad, rendimiento y escalabilidad unica.

2.3.1 Señales y Ranuras

Las señales (*signals*) y ranuras (*slots*) son usadas para la comunicación entre objetos. El mecanismo *signal/slot* es una característica central y probablemente la parte que mas la diferencia de los demás juegos de herramientas (*toolkits*).

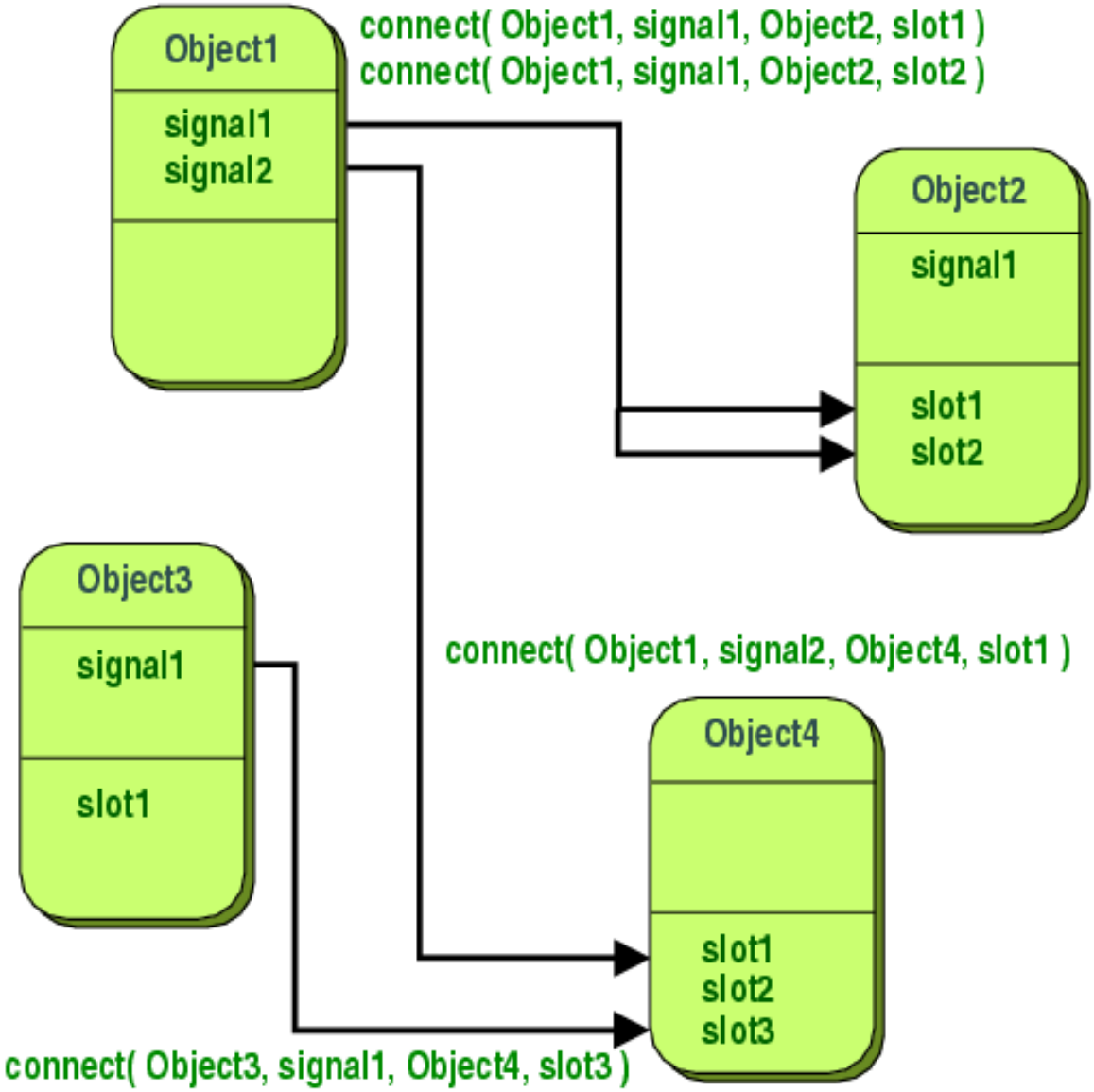

Figura Nº 14. Una vista abstracta de algunas conexiones entre señales y ranuras.

2.3.1.1 Señales

Las señales son emitidas por un objeto cuando su estado interno ha cambiado y de alguna manera esto podría ser de interés para el objeto cliente o él mismo. Cuando una señal es emitida, las ranuras conectadas a el son ejecutadas inmediatamente. Este mecanismo *signal/slot* es totalmente independiente de cualquier GUI *even loop*.

2.3.1.2 Ranuras

Una ranura es llamada cuando una señal conectada a el es emitida, son funciones comunes y corrientes que pueden ser llamadas normalmente, su característica particular es que una señal puede ser conectada a esta.

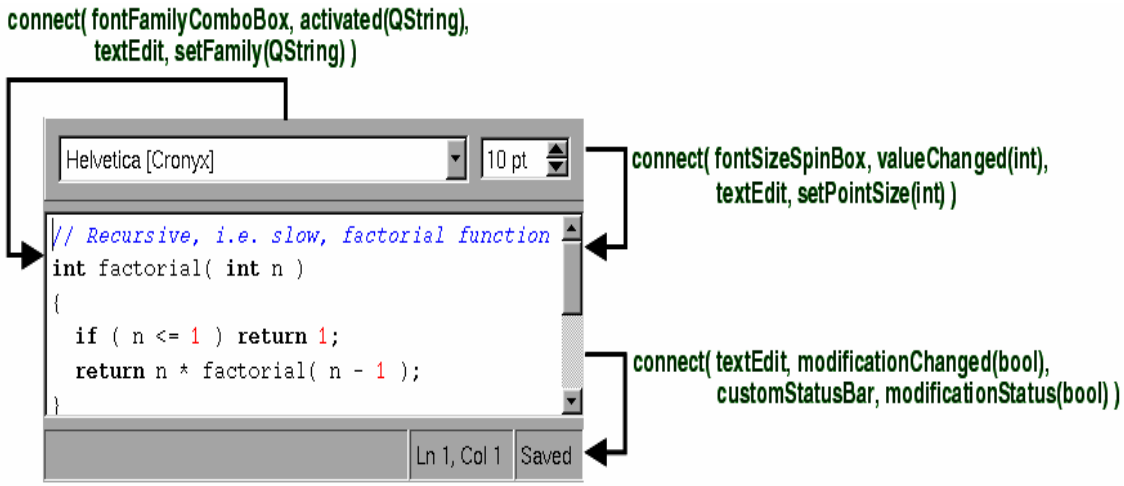

Figura Nº 15. Un ejemplo de conexiones entre señales y ranuras.

2.3.2 *Meta Object System.*

El *Meta Object System* de Qt proporciona el mecanismo para la comunicación entre objetos, información de tipos en tiempo de ejecución y el sistema dinámico de propiedades.

El *Meta Object System* esta basado en 3 cosas:

1. La clase QObject.
2. La macro Q_OBJECT en la sección privada de la declaración de la clase.
3. El *Meta Object Compiler* (moc).

El moc lee un archivo fuente C++. Si encuentra una o más declaraciones de clase que contienen la macro Q_OBJECT, produce otros archivos fuentes C++ los cuales contienen el código *meta object*. El código fuente C++ generado por la herramienta moc solo puede ser compilado y enlazado con la implementación de la clase.

2.4 El Modelo en Espiral.

La metodología para el desarrollo del Sistema, aquí expuesta se basa en el denominado ***modelo en espiral,*** propuesto originalmente por Boehm [BOE88].

El modelo en espiral es un modelo de proceso de software evolutivo que acompaña la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Se proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas de ingeniería del sistema.

El ciclo de vida en espiral, añade el análisis de riesgo en el proyecto de software y considera 4 fases: Planificación, análisis de riesgo, ingeniería y evaluación de proyectos:

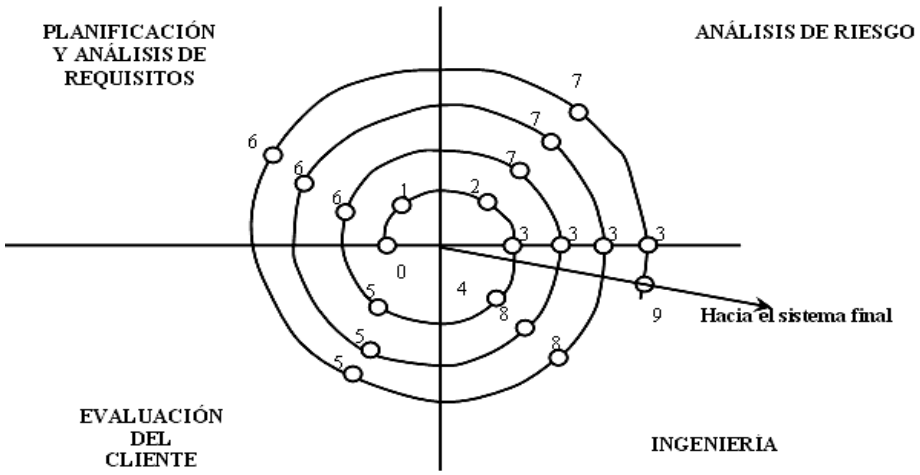

Figura Nº 16. Fases de la metodología en Espiral

Cada Fase de la metodología es una combinación de los ítems mostrados a continuación:

0.- Decisión de desarrollar un determinado software
1.- Recolección de requisitos iniciales, planificación del proyecto
2.- Análisis de riesgo según los requisitos iniciales.
3.- ¿Continúa el proyecto?
4.- Prototipo inicial.
5.- Evaluación del cliente.
6.- Replanificación según comentarios del cliente.
7.- Análisis del riesgo según la reacción del cliente.
8.- Prototipo del siguiente nivel.

9.- Sistema final.

Con cada iteración alrededor de la espiral, se construyen sucesivas versiones del software, cada vez más completas. El paradigma del Modelo en Espiral es actualmente el enfoque más realista para el desarrollo de software y de sistemas en gran escala.

Este método demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto y si, se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

2.5 *Unified Modeling Language* (UML).

Para el modelado de la aplicación, Lenguaje Unificado de Construcción de Modelos (UML**)** es el aplicado y el mas apropiado considerando que la programación de nuestro sistema ha sido desarrollada bajo la filosofía POO (Programación Orientada a Objetos).

Unified Modeling Language (UML), Lenguaje Unificado de Construcción de Modelos se define como un "lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de Software". UML es un sistema rotacional estándar incipiente de la industria para construir modelos orientados a objetos. Surgió de la combinación de las 3 más famosas metodologías de desarrollo como lo son Booch, OMT y OOSE.

Por tales razones es el estándar aceptado por las empresas y desarrolladores a nivel mundial, pues envuelve bajo si, lo mejor de las técnicas de orientación por objetos más elegantes y robustas. Surge de la necesidad de crear un estándar único e incipiente dentro del mundo del desarrollo de sistemas programados y la Ingeniería del Software.

El UML se caracteriza por describir mediante diagramas todas las vistas, partes, componentes, etc. de un sistema programado, con lo cual se persigue un entendimiento y conocimiento detallado del sistema que se está generando (por parte del desarrollador) y permite también a los clientes o usuarios conocer internamente como funciona y como fue concebido (empresas, clientes, etc.). Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación. En la **tabla Nº 1** se muestran los diagramas con los cuales se puede modelar eficazmente un sistema programado.

Diagrama	Descripción	Prioridad
Actividad	Representa los procesos de alto nivel del negocio, incluyendo el flujo de datos, o modela la lógica de la lógica compleja dentro de un sistema.	Alto
Clases	Muestra una colección de elementos del modelo estático por ejemplo clase y tipos, su contenido, y sus relaciones	Alto
Comunicación	Muestra las instancias de las clases, sus interrelaciones, y los mensajes de flujo entre ellos. Los Diagramas de comunicación se enfocan típicamente en los objetos que envían y reciben mensajes dentro de la organización. Llamado también Diagramas de Colaboración.	Bajo
Componentes	Representa los componentes que componen un uso, un sistema, o una empresa. Se representan los componentes,	Medio

	sus correlaciones, las interacciones, y sus interfaces públicas.	
Composición de la Estructura	Representa la estructura interna de un clasificador (tal como una clase, un componente, o un caso de uso), incluyendo los puntos de la interacción del clasificador con las otras partes del sistema.	Bajo
Despliegue	Muestra la arquitectura de ejecución de los sistemas. Esto incluye nodos, los ambientes de la ejecución del hardware o del software, así como el middleware que los conecta.	Medio
Descripción de la Interacción	Una variante de un diagrama de actividad que describe el flujo del control dentro de un proceso del sistema o del negocio. Cada nodo/actividad dentro del diagrama puede representar otro diagrama de la interacción.	Bajo
Objetos	Representa objetos y sus relaciones en un punto en el tiempo, típicamente un caso especial de un diagrama de la clase o de un diagrama de comunicación.	Bajo
Paquetes	Demuestra cómo los elementos del modelo se organizan dentro de los paquetes así como las dependencias entre los paquetes.	Bajo
Secuencia	Modela la lógica secuencial, en el efecto de ordenamiento temporal de los mensajes	Medio

	entre los clasificadores.	
Estado de la Máquina	Describe los estados de un objeto o la interacción que puede estar dentro del mismo, así como las transiciones entre los estados. Designado antes un diagrama de estado-transición.	Medio
Medida del tiempo (timing)	Representa el cambio en estado o la condición de una instancia del clasificador o en un cierto plazo. Típicamente demostraban el cambio en estado de un objeto en un cierto plazo en respuesta a acontecimientos externos.	Bajo
Casos de Uso	Muestra los casos de uso, los actores y las relaciones entre ellos.	Alto

Tabla Nº 1. Diagramas del UML

CAPITULO 3
CONSTRUCCION DEL SISTEMA

En este capítulo se describe detalladamente la metodología para el diseño, desarrollo y futura implementación del sistema.

3.1 Modelado Funcional del Sistema

Esta sección corresponde a la elaboración del proceso de desarrollo de software empleado para este proyecto y se describen todos los artefactos y disciplinas obtenidos del desarrollo.

3.1.1 Requerimientos Funcionales

Un proyecto no puede ser exitoso sin una especificación correcta o exhaustiva de los requerimientos. A continuación se muestra una tabla (**Tabla Nº 2)** donde se presentan los requerimientos funcionales del proyecto, cada requerimiento se encuentra identificado con una clave, a la cual se hará referencia de aquí en adelante dentro de este proyecto.

Ref #	Función
R1	Selecciona 2 fotografías
R2	Despliega ambas fotografías
R3	Captura los parámetros asociados al cálculo de las coordenadas y/o pertenecientes a las fotografías.
R4	Posiciona el sistema de coordenadas
R5	Posiciona punto homologo
R6	Crea el anaglifo
R7	Ajusta sobre-posición de las fotografías
R8	Registra coordenadas del punto homologo
R9	Almacena coordenadas en medio persistente

Tabla Nº 2. Requerimientos funcionales del proyecto

3.1.2 Requerimientos No Funcionales

Los requerimientos no funcionales del sistema representan aquellos aspectos del sistema, que no cumplen una función específica, pero contribuyen a la interacción entre el sistema y los actores.

- Interfaz grafica amigable e intuitiva. El uso de la herramienta Qt de TrollTech facilita la creación de interfaces de usuario sencillas e intuitivas.
- Sistema robusto e independiente que permita la interpretación de las fotografías aéreas sin ayuda de sistemas externos.
- El mínimo de requerimientos posibles que permita su uso en computadoras de bajo desempeño.
- Desarrollo bajo la filosofía de software libre y código abierto. TrollTech permite el uso de Qt para desarrollar software bajo esta filosofía.

3.1.3 Diagramas de casos de uso

Este formato muestra una descripción para ayudar a comprender mejor los casos de uso, también hace referencia a los requerimientos (que fueron especificados en el apartado de requerimientos dentro de este mismo punto) con los cuales guarda relación. Las descripciones para cada caso de uso se encuentran detalladas en el **Anexo A**.

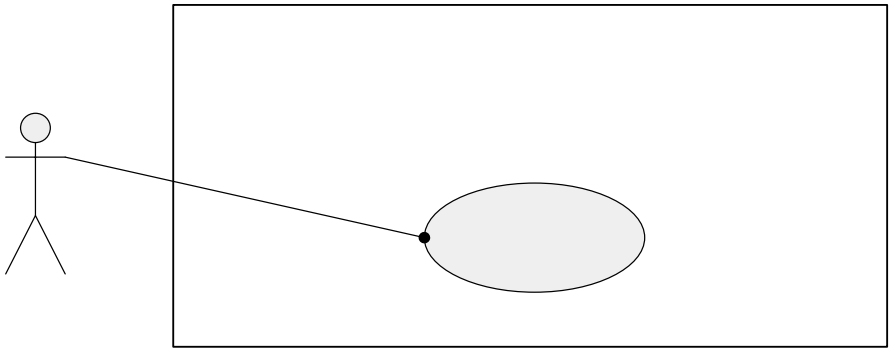

Figura Nº 17. Caso de Uso Visualizar Anaglifo

Actores:	Operador
Función:	El caso de uso *Visualizar Anaglifo,* permite al Operador visualizar la imagen anaglífa creada por el sistema a través de las fotografías aéreas del terreno.
Descripción:	1. El Operador selecciona modo Anaglifo. 2. El sistema muestra una caja de dialogo que contiene el visor anaglifo. 3. El sistema el sistema crea el anaglifo y lo muestra en el visor.
Referencia:	R6

Tabla Nº 3. Descripción del Caso de Uso Visualizar Anaglifo

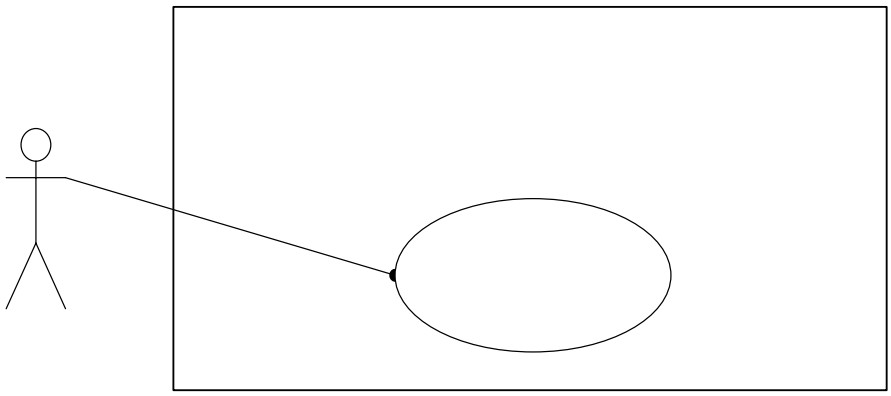

Figura Nº 18. Caso de Uso Asignar grado de solapamiento o separación horizontal

Actores:	Operador
Función:	El caso de uso *Asignar grado de solapamiento o separación horizontal,* permite al Operador modificar el grado de solapamiento *o separación horizontal* con el cual se construirá el anaglifo.
Descripción:	1. El Operador posiciona el cursor sobre la barra deslizadora o slider. 2. El Operador arrastra la barra deslizadora o mueve el scroll hacia arriba o abajo según sea el caso. 3. El sistema crea el anaglifo con el nuevo ajuste.
Referencia:	R6, R7

Tabla Nº 4. Descripción del Caso de Uso Asignar grado de solapamiento o separación horizontal

Operador

3.1.3 Desarrollo

Se especifican cada una de las iteraciones en las que se llevo a cabo el desarrollo del proyecto.

3.1.3.1 Planificación y Análisis Inicial de Requisitos

Los requerimientos funcionales fueron especificados en el apartado de requerimientos dentro de este mismo capitulo.

3.1.3.2 Prototipo inicial

1. Constaba de una Ventana principal.
2. Todas las operaciones, se hacían sobre la ventana principal.
3. Las Operaciones abarcadas se resumían en:
   a. Carga de una de las fotografía en la ventana principal.
   b. Colocación de la marca o punto en la fotografía, sin referencia alguna dentro del plano cartesiano (XY).
   c. Visor anaglifo, que no permitía la visualización estereoscópica correcta.
   d. El grado o porcentaje de sobre-posición de las fotografías era asociado al movimiento de la rueda de desplazamiento vertical del Mouse (scroll) como también, a un widget de tipo slider localizado en la parte inferior del visor.
   e. La opción de escoger entre el uso de un solo buffer de procesamiento (single buffer) o un buffer doble (double buffer) para mejorar el rendimiento y la visión del anaglifo.

3.1.3.3 Evaluación

1. Mejorar el visor anaglifo.

2. Implementar la construcción del anaglifo, con 2 fotografías.
3. Implementar el Sistema de Coordenadas.
4. Implementar la colocación del punto o marca dentro del anaglifo.

3.1.3.4 Segundo Prototipo

1. Consta de dos ventanas, la principal que contiene los menús, opciones principales y el visor de la fotografía, la secundaria que contiene el visor anaglifo.
2. En el estado de visión anaglifo, la ventana secundaria se muestra *fullscreen* o modo de ventana completa.
3. El control *slider* (barra deslizadora) ya no se encuentra en la ventana principal sino en la secundaria, donde se encuentra el visor anaglifo.
4. Se implemento el sistema de coordenadas dentro de la aplicación (en escala de píxeles).
5. La colocación del origen del plano cartesiano en la fotografía es asociado a los eventos *click* y doble *click* del mouse sobre el área de la fotografía, una vez seleccionada el modo sistema de coordenadas.
6. La colocación de la marca o punto dentro del plano cartesiano en la fotografía es asociado a los eventos *click* y doble *click* del mouse sobre el área de la fotografía, una vez seleccionada el modo punto homologo.
7. El anaglifo es construido con dos fotografías o pares estereoscópicos del mismo terreno a la misma escala.

3.1.3.5 Evaluación

1. Definir la escala del sistema de coordenadas con respecto a la escala y la resolución de la fotografía.

2. Sistema de Almacenamiento de los datos extraídos del anaglifo (lista de coordenadas de las marcas y el origen del sistema de coordenadas).

3.1.3.6 Producto Final

Constituye la aplicación con toda su funcionalidad y todos los requerimientos funcionales y no funcionales cumplidos. La descripción del sistema programado se manifiesta en el apartado **3.2 El Sistema Programado.**

3.1.4 Diagramas de Clases

El siguiente modelo muestra las asociaciones entre los objetos que conforman la aplicación, las especificaciones formales de cada clase se encuentran en el **Anexo C**.

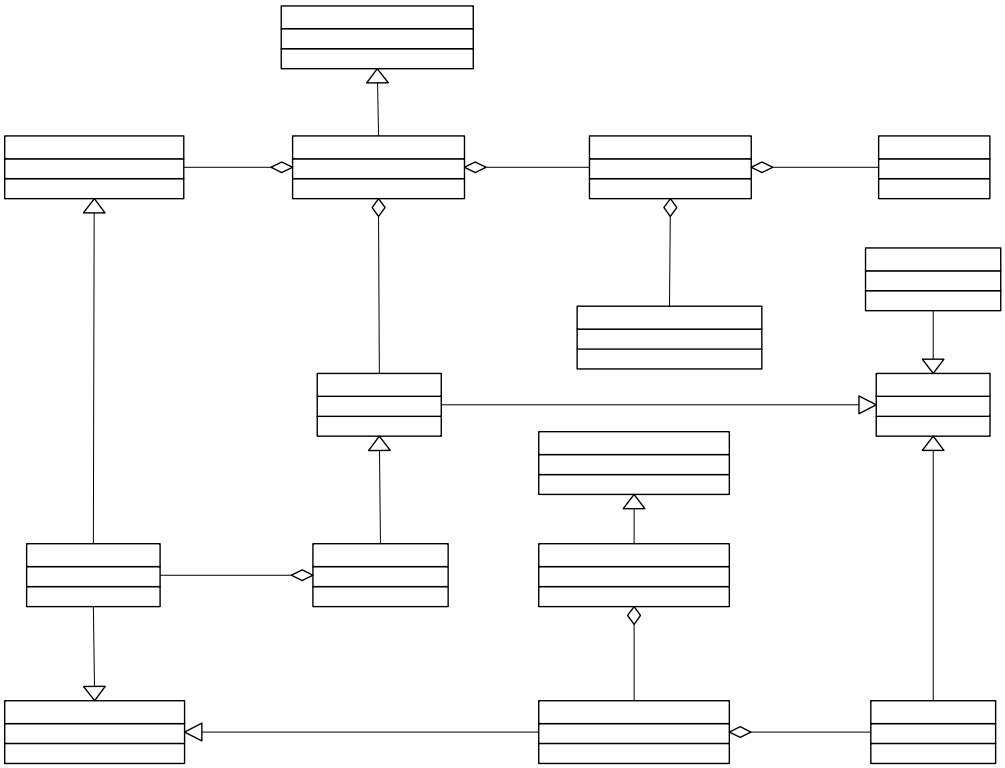

Figura Nº 19. Diagrama de clases

Puesto que todas las clases que conforman el sistema heredan directa o indirectamente de las clases pertenecientes a las bibliotecas de Qt, los métodos e interfaces heredadas de dichas dependencias no serán especificadas formalmente en TDSO, solo se especificarán los métodos y atributos de las clases herederas, la documentación de las clases base se encuentra en el **Anexo D.**

3.1.4 Diagramas de Interacción

Los diagramas a continuación denominados de secuencia y colaboración muestran las interacciones del Operador con el sistema programado además del intercambio de mensajes entre los objetos y el Operador. Los diagramas de interacción para cada caso de uso se encuentran en el **Anexo B** del presente proyecto.

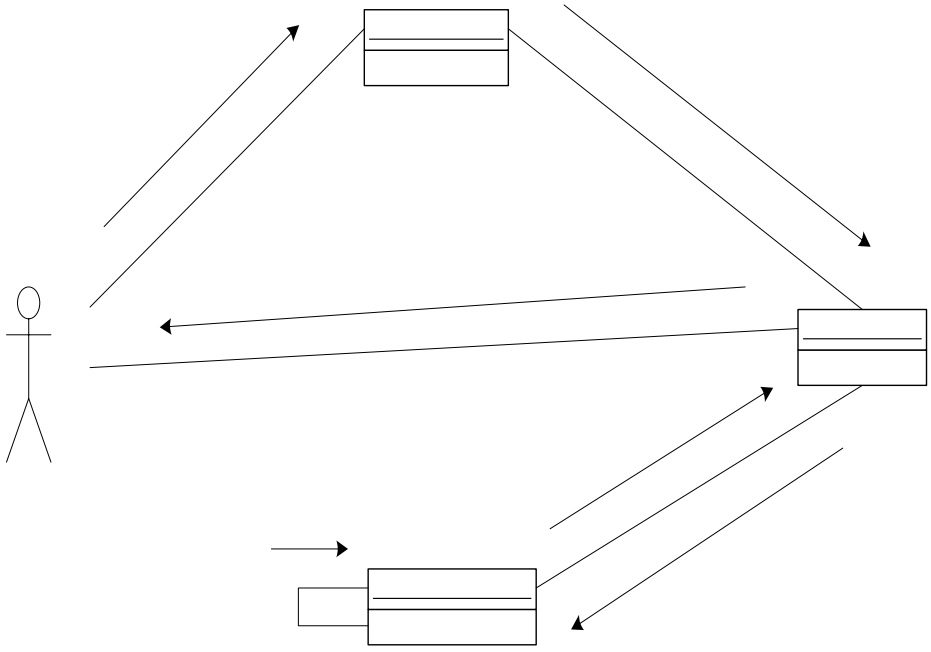

Figura Nº 20. Diagrama de colaboración del Caso de Uso Visualizar Anaglifo

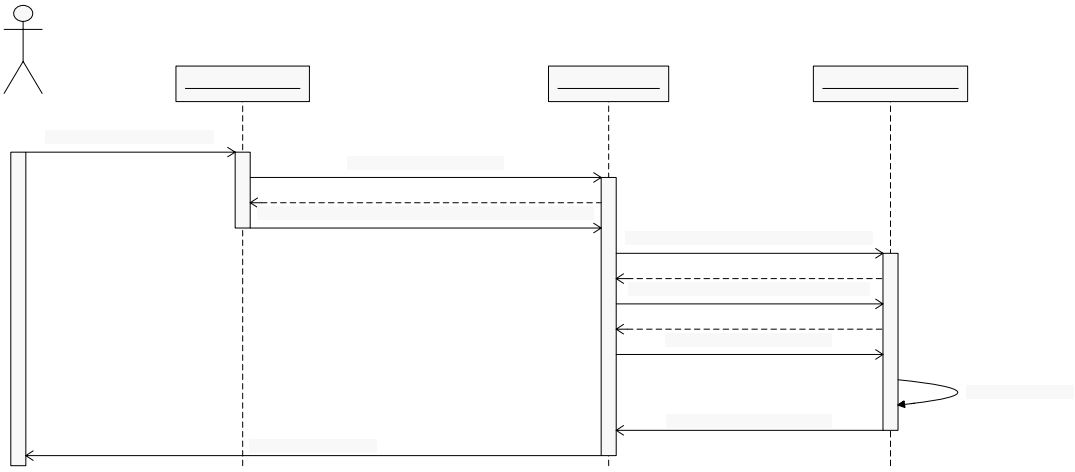

Figura Nº 21. Diagrama de secuencia del Caso de Uso Visualizar Anaglifo

3.1.5 Diagrama de Componentes

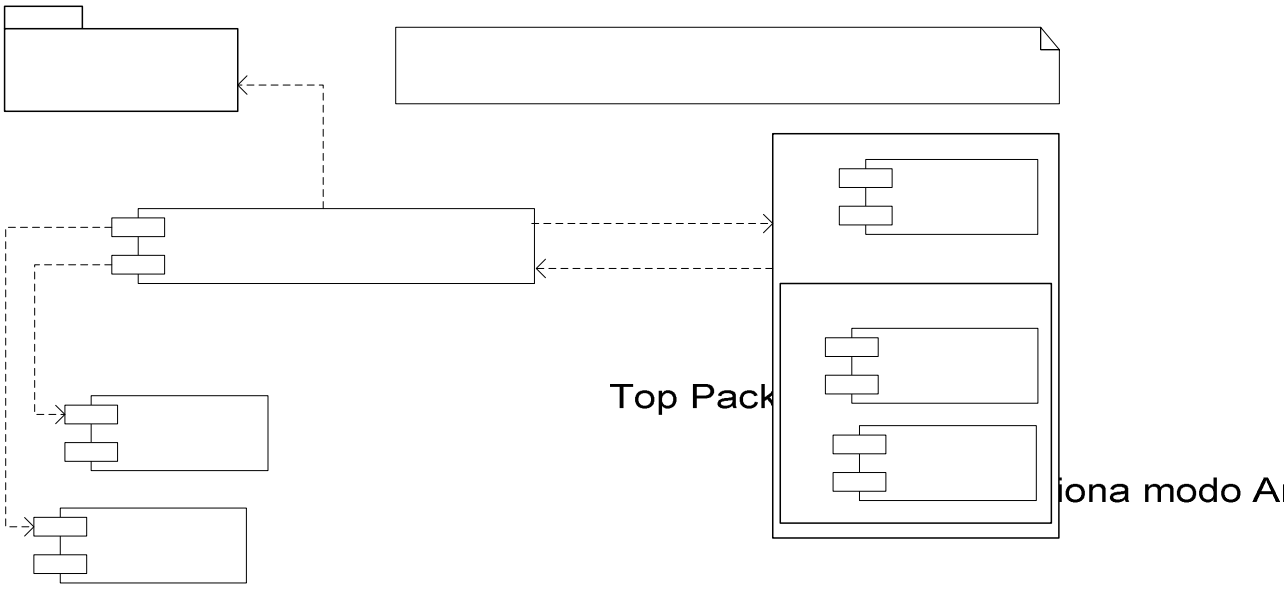

Figura Nº 22. Diagrama de componentes.

La figura mostrada en la parte superior (**Figura Nº 22**) pone de manifiesto las dependencias entre los componentes a los cuales hace uso el sistema**.** A continuación se hace una breve descripción de cada uno de ellos:

- Bibliotecas dinámicas de Qt: Constituyen todas y cada una de los ficheros dll, de los cuales hace uso la aplicación para levantar la interfaz grafica.
- Sistema de visualización de Imágenes tridimensionales con uso de marca flotante: Es la aplicación como tal, necesita de los componentes "Imagen Izquierda" e "Imagen derecha" para crear el anaglifo.
- Imagen Izquierda: Es el fichero de imagen que contiene la fotografía izquierda como tal y sirve de artefacto para construir el anaglifo.
- Imagen Derecha: Es el fichero de imagen que contiene la fotografía derecha como tal y sirve de artefacto para construir el anaglifo.
- Archivo binario: Es el fichero donde se almacenan cada una de las coordenadas registradas o extraídas del anaglifo, como también los parámetros utilizados para calcular dichas coordenadas.

3.2 El Sistema Programado

La aplicación consta de una ventana principal (**Figura Nº 23**), en la parte superior de la ventana se encuentra la barra de menú, a través de ella se accede a las acciones principales, como crear un nuevo proyecto, abrir un proyecto ya creado, asignar parámetros, acceder a datos registrados, etc.

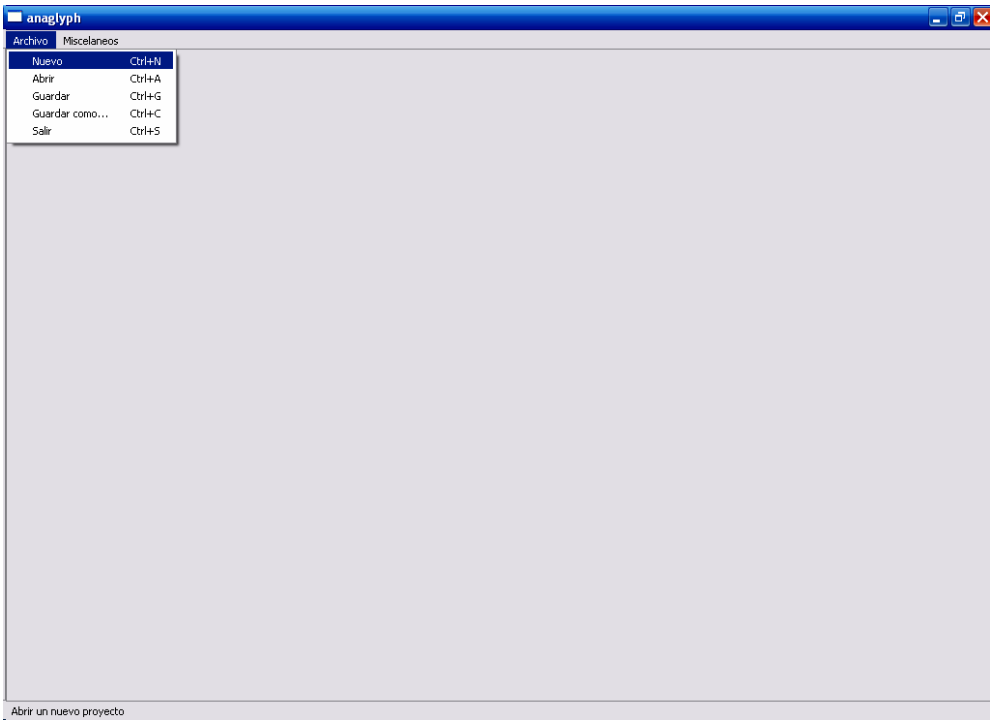

Figura Nº 23. Ventana principal

En la parte central de la ventana principal se encuentra localizado el visor fotográfico, donde se muestra la fotografía izquierda una vez cargada (solo se muestra la fotografía izquierda puesto que es a esta a la cual se le posará el sistema de coordenadas de referencia y el punto homologo). El visor cuenta con un menú de contexto, que contiene las acciones esenciales de la aplicación, como el activar el modo anaglifo, el sistema de coordenadas de referencia, el punto homologo y asignar los parámetros (**Figura Nº 24**).

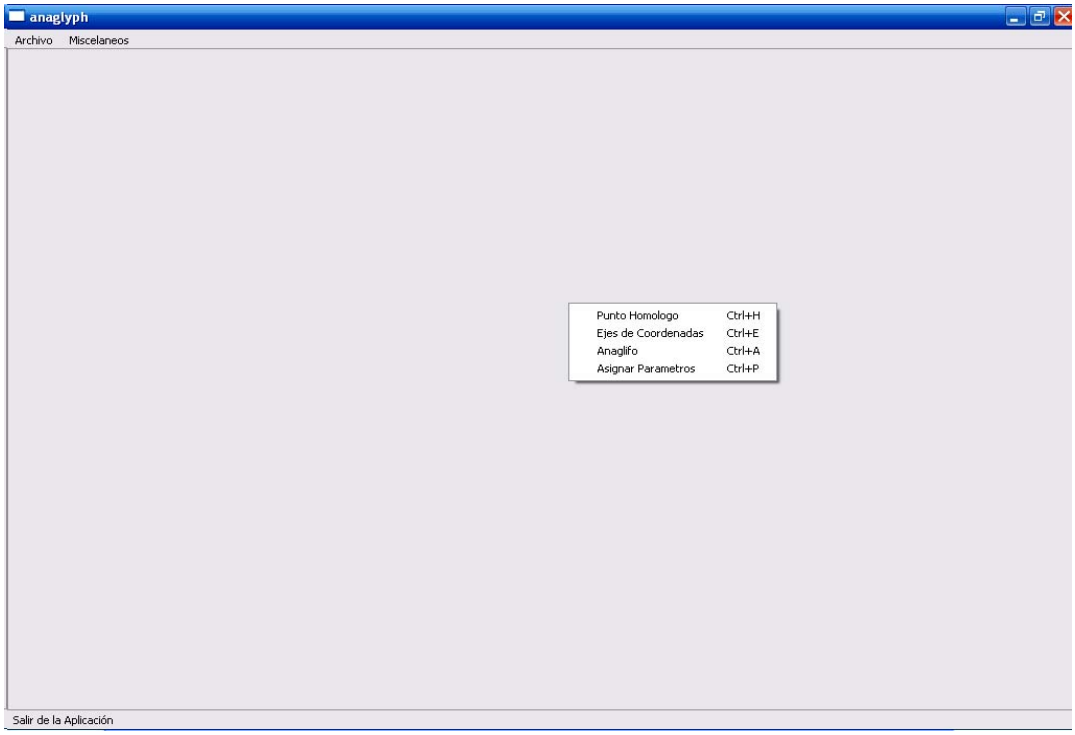

Figura Nº 24. Ventana principal con Menú de Contexto

Una vez en la ventana principal el Operador, tiene 3 posibilidades iniciales:

1. Abrir un nuevo proyecto. Si el Operador selecciona esta operación se desplegará un caja de dialogo, en el cual se le pedirá seleccionar las fotografías y el nombre del proyecto (**Figura Nº 25**). Si decide "Aceptar", se cierra la caja de dialogo y se carga la fotografía derecha sobre el visor fotográfico en la ventana principal, de lo contrario se cierra la caja de dialogo y la aplicación vuelve a su estado inicial.
2. Abrir un proyecto ya creado. Si el Operador selecciona esta operación se desplegará una caja de dialogo donde podrá seleccionar un proyecto (**Figura Nº 26**). El Operador tiene la posibilidad de explorar cualquier directorio del disco duro en busca de un proyecto en especial (los proyectos tienen extensión pya), una vez seleccionado el Operador decide si seleccionar "Aceptar" para continuar con la operación o "Cancelar" para abandonar en cualquier

momento la acción. Si se decide por “Aceptar”, se cierra la caja de dialogo y se carga la fotografía derecha sobre el visor fotográfico en la ventana principal, de lo contrario se cierra la caja de dialogo y la aplicación vuelve a su estado inicial.

3. Salir de la aplicación. Simplemente se cerrará la aplicación.

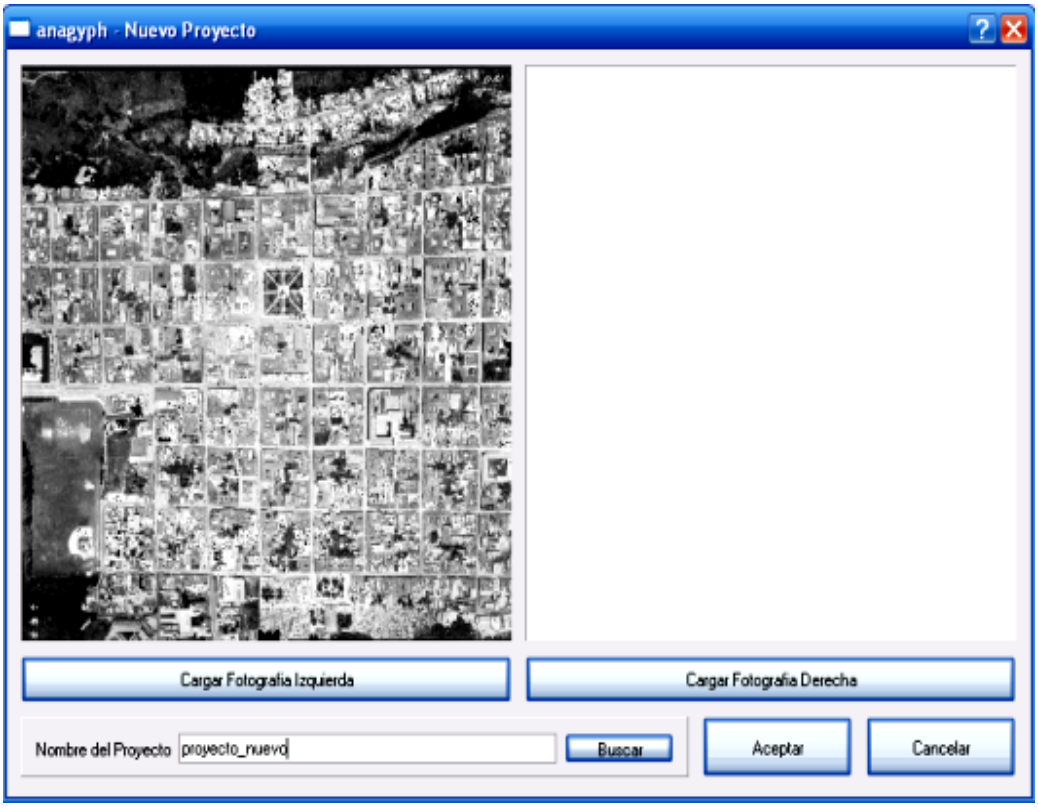

Figura Nº 25. Caja de dialogo para un Nuevo proyecto