

Instituto de Computación, Facultad de Ingeniería
Universidad de la República



Generación de Modelos Digitales de Edificios a partir de Imágenes Satelitales Estereoscópicas

Informe final
Proyecto de Grado P2012_0040

Autores

Matías Davyt
Mario Paulsen

Tutores

Dr. Ing. Andrés Almansa
Ing. Javier Preciozzi

Cliente

Dr. Ing. Gabriel Usera

Diciembre 2013, Montevideo, Uruguay

Resumen

Las imágenes satelitales estereoscópicas consisten en un par de imágenes que corresponden a dos puntos de vista diferentes de una zona de interés, lo que permite la simulación de una visión en 3 dimensiones (3D) de los objetos, de igual forma que se comporta la visión del ser humano.

Este proyecto forma parte de una línea de investigación del Grupo de Mecánica de los Fluidos Computacional del Instituto de Mecánica de los Fluidos e Ingeniería Ambiental (IMFIA). Dicha línea plantea como uno de sus objetivos la realización de simulaciones de viento en la zona donde se ubica la Facultad de Ingeniería, para evaluar la factibilidad de instalar un molino de viento en la Facultad con la finalidad del autoabastecimiento energético de la misma.

Para inferir cómo se comporta el viento en la zona de interés, se necesita previamente simular las construcciones que existen alrededor, a partir de esto es que surge la necesidad de crear un modelo 3D del lugar. Este proyecto pretende aportar a la línea de investigación, brindando la posibilidad de calcular la altura de los edificios a partir de dos imágenes satelitales estereoscópicas y así conformar el modelo tridimensional del área.

Con el fin de evaluar la factibilidad de dicho cálculo y la posterior implementación en una aplicación que cumpla con los requerimientos, se realiza un estudio conceptual y de antecedentes relacionados al tema de tratamiento de imágenes. Posteriormente se investiga acerca del estado del arte, que arroja como resultado las dos formas de calcular la altura en este tipo de problemas: *baseline* y modelo RPC. Con la información disponible para este proyecto la alternativa de *baseline* se descarta y la de modelo RPC surge como la adecuada para el tipo de imágenes que se adquirieron. En el proceso desde la obtención de la imagen hasta la obtención de la altura y la georreferenciación de un edificio, se pueden identificar claramente dos grandes etapas: la correlación y el cálculo de altura, cada una con más de un método de resolución.

El resultado final es una aplicación implementada en Matlab, que recibe como entrada las imágenes satelitales estereoscópicas y al seleccionar los *outlines* (edificios) permite el procesamiento de las imágenes y la obtención de la latitud, longitud y altura de cada edificio. La aplicación desarrollada realiza la tarea requerida con buena precisión, de acuerdo a comparaciones realizadas contra medidas tomadas con GPS y aplicaciones similares disponibles en el mercado. Además, el sistema agrega valor al permitir obtener el modelo 3D de cualquier ciudad a partir de sus imágenes satelitales estereoscópicas.

Palabras claves: tratamiento de imágenes, estereoscopia, correlación, modelos digitales de edificios, Matlab.

Índice

1. Introducción.....	6
1.1. Descripción del problema	6
1.2. Objetivos	7
1.3. Resultados y aportes del proyecto	7
1.4. Organización del informe	8
2. Marco Conceptual.....	9
2.1. Tratamiento de imágenes	9
2.1.1. Estereoscopía.....	9
2.1.2. Geometría Epipolar	9
2.1.3. Rectificación de imágenes	11
2.1.4. Propiedades de las imágenes satelitales.....	11
2.2. Disparidad	14
2.2.1. Tipos de algoritmos	14
2.2.2. Taxonomía de algoritmos para cálculos estéreo.....	15
2.3. Cálculo de alturas y georreferenciación	16
2.3.1. Definiciones de las diferentes alturas	16
2.3.2. Modelo RPC	16
2.3.3. Baseline	20
3. Análisis.....	21
3.1. Descripción del sistema.....	21
3.2. Requerimientos del sistema	22
4. Planteo de la solución.....	25
4.1. Descripción General.....	25
4.2. Disparidad	25
4.2.1. IPOL.....	26
4.2.2. StereoMatch	27
4.2.3. Orfeo toolbox	27
4.3. Modelado de techos.....	28
4.4. Cálculo de alturas y georreferenciación	29
4.5. Importación y exportación de outlines	30
4.6. Exportación del modelo 3D.....	31
5. Diseño e implementación	32

5.1. Arquitectura	32
5.2. Diagrama de clases de diseño	34
5.3. Tecnologías utilizadas	35
5.3.1. Matlab	35
5.3.2. Mksqlite.....	35
5.3.3. Orfeo toolbox	35
5.4. Implementación.....	36
5.5. Dificultades encontradas.....	41
5.6. Limitaciones	42
6. Validaciones	43
6.1. Validaciones de disparidad.....	43
6.2. Validaciones modelado de techos	44
6.3. Validaciones realizadas al programa.....	44
6.3.1. CityModeler.....	44
6.3.2. GPS	46
6.3.3. Stereog3mas	48
6.3.4. Cuadro comparativo de los resultados	50
7. Conclusiones.....	51
8. Trabajo a futuro.....	52
9. Bibliografía	54
10. Anexos	54

Índice de figuras

Figura 1 - Estereoscopia	9
Figura 2 - Geometría epipolar.....	10
Figura 3 - Proceso de rectificación.....	11
Figura 4 - Precisión sin corrección del terreno.....	12
Figura 5 - Geometría de la imagen adquirida	13
Figura 6 - Block Matching	14
Figura 7 - Altura elipsoidal y ortométrica.....	16
Figura 8 - Extracción de características estéreo.....	17
Figura 9 - Generación RPC	19
Figura 10 - Cálculo mediante baseline	20
Figura 11 - IPOL.....	26
Figura 12 - Diferentes métodos para el modelado de los techos.....	29
Figura 13 - Diagrama de arquitectura	32
Figura 14 - Arquitectura de tubos y filtros	34
Figura 15 - Diagrama de clases de diseño.....	34
Figura 16 - Proyecto CityModeler zona Facultad de Ingeniería	45
Figura 17 - Puntos medidos visualizados en Google Earth	47
Figura 18 - Edificios tomados para validar	47
Figura 19 - Stereog3mas	49
Figura 20 - Modelo 3D de la zona de Facultad de Ingeniería.....	49
Figura 21 - Métodos de obtención de aristas	52

Índice de tablas

Tabla 1 – Comparación de algoritmos de disparidad, entre paréntesis se muestra el error y se resaltan los mejores resultados.....	43
Tabla 2 – Comparación de métodos para el cálculo de los techos	44
Tabla 3 – Alturas ortométricas.....	47
Tabla 4 – Medidas GPS.....	48
Tabla 5 – Comparación de alturas	50
Tabla 6 – Comparación de georreferenciación	50

1. Introducción

El Grupo de Mecánica de los Fluidos Computacional del Instituto de Mecánica de los Fluidos e Ingeniería Ambiental (IMFIA) está embarcado en la iniciativa de crear un modelo digital de edificios de la ciudad de Montevideo, a partir del procesamiento de imágenes. En su investigación acerca de los productos existentes en el mercado que permitieran obtener la altura de los edificios a partir de imágenes satelitales, encontraron que estos productos tienen un costo elevado, licencias de más de 10.000 euros (CityModeler¹ y Tridicon), fundándose la iniciativa de invertir recursos en un proyecto de grado, brindando la posibilidad a estudiantes de desarrollar un producto propio utilizando bibliotecas de código abierto.

Esta iniciativa es un proyecto de una línea de investigación del IMFIA en la temática de eólica urbana, financiado por la Agencia Nacional de Investigación e Innovación² (ANII). La finalidad de este proyecto es realizar un modelado tridimensional de los edificios ubicados en la zona de Facultad de Ingeniería, para realizar simulaciones de viento y poder evaluar la factibilidad de colocar un molino de viento para el autoabastecimiento energético de la Facultad de Ingeniería.

En este marco surge el proyecto, que se propone el desarrollo de una herramienta basada en bibliotecas *open source*³ así como en desarrollos propios, que genere modelos digitales de edificios a partir de imágenes satelitales estereoscópicas de alta resolución. A partir de la identificación de objetos (techos de edificios) en las mismas, pretende hallar la altura del edificio, su posición georreferenciada, tipología del techo, además de permitir la visualización de los edificios en 3D.

1.1. Descripción del problema

Ante la necesidad de obtener la altura de los edificios cercanos a la zona de facultad, se procede a investigar qué organismo podría tener esta información. A partir de la visita a la Intendencia, Catastro y el Servicio Geográfico Militar del Uruguay⁴ (SGM), se descubre la no existencia de una base centralizada al respecto. Una solución consiste en la medición de las alturas con GPS, lo que no es viable porque implicaría realizar la medición en cada edificio lo cual sería muy costoso. La otra opción es la utilización de fotografías satelitales, que es la opción seleccionada para este proyecto.

¹ (2013, Noviembre) CityModeler [Online] <http://citymodeler.com/index.html>

² (2013, Noviembre) Agencia Nacional de Investigación e Innovación [Online] <http://www.anii.org.uy/web/>

³ Open source: término con el que se conoce al software distribuido y desarrollado libremente.

⁴ (2013, Noviembre) Servicio Geográfico Militar del Uruguay [Online] <http://www.sgm.gub.uy/>

1.2. Objetivos

El objetivo principal del proyecto es crear un modelo 3D de la zona de la Facultad de Ingeniería (FING), para lo cual se plantean las siguientes metas intermedias:

- Evaluar la factibilidad de hallar las alturas de edificios a partir de imágenes satelitales, para generar un modelo digital de edificios de una zona de interés.
- Analizar las distintas alternativas que existen para calcular las alturas y obtener la solución más adecuada para el problema.
- Desarrollar una aplicación que genere modelos digitales de edificios, automatizando la mayor cantidad de pasos posibles.
- Comparar la calidad de los resultados obtenidos con respecto a aplicaciones similares existentes en el mercado.

1.3. Resultados y aportes del proyecto

- Se evaluó la factibilidad de hallar la altura de edificios a partir de imágenes satelitales, para generar un modelo digital de edificios de una zona de interés.

Se estudió el estado del arte respecto a la obtención de alturas a partir de imágenes estereoscópicas, encontrando diferentes métodos para calcularlas. Se procedió a hallar la altura de un punto, cuya factibilidad permitió derivar la de un edificio completo.

- Se analizaron las distintas alternativas que existen para calcular la altura y se obtuvo la solución más adecuada para el problema.

El problema de hallar la altura puede ser dividido en problemas de más corto alcance. Analizando en detalle cada uno de ellos, se encontraron programas que permiten probarlos; comparando los resultados se escogió la mejor solución.

- Se desarrolló una aplicación que genera modelos digitales de edificios, automatizando la mayor cantidad de pasos posibles.

Se implementó un software a medida para generar modelos de edificios tridimensionales en Matlab. Este *software* permite elegir el algoritmo a utilizar, entre las distintas alternativas para calcular las alturas. El producto desarrollado tiene el valor agregado de poder obtener un modelo digital de una zona de interés de cualquier ciudad. Además, se entregan los manuales de usuario y de instalación, los cuales permiten instalar el programa sin problemas y utilizarlo sin la necesidad de ser un especialista.

- Se comparó la calidad de los resultados obtenidos con respecto a aplicaciones similares existentes en el mercado.

Se eligió la zona de Facultad de Ingeniería como área de interés, que fue el móvil del proyecto. Se midieron varios edificios conocidos con la aplicación desarrollada, GPS y una aplicación del mercado para comparar las medidas, arrojando resultados muy similares entre sí.

1.4. Organización del informe

En el capítulo 2 se presenta el marco conceptual, el cual describe los conceptos necesarios para una clara interpretación del problema. En las diferentes secciones se presentan: tratamiento de imágenes, disparidad, cálculo de alturas y georreferenciación.

En el capítulo 3 se analiza el problema planteado por el cliente y se describe el sistema a construir. Además, se enumeran los requerimientos funcionales y no funcionales del sistema, así como la documentación a entregar.

En el capítulo 4 se plantea la solución. Para los aspectos más importantes del proyecto se detallan las decisiones tomadas indicando el algoritmo elegido para hallar la disparidad, método para el modelado de los techos y procedimiento usado para el cálculo de altura.

En el capítulo 5 se explica cómo se diseñó e implementó el producto, explicando la arquitectura elegida en el sistema y en sus componentes. Se explican brevemente las tecnologías utilizadas. Finalmente, se describen las dificultades encontradas en el transcurso del proyecto y se enumeran algunas de las limitaciones que presenta la solución desarrollada.

En el capítulo 6 se presentan las validaciones realizadas a cada etapa y al producto generado. Se muestran las mediciones de edificios conocidos realizadas con GPS, una aplicación del mercado y la aplicación desarrollada. Se comparan los resultados.

En los capítulos 7 y 8 se presentan las conclusiones y el trabajo a futuro que se podría realizar manteniendo la línea de este proyecto.

2. Marco Conceptual

El objetivo de este apartado es describir y explicar brevemente algunos conceptos relevantes para el proyecto. Se ordenan los temas según el orden de procesamiento que reciben las imágenes. Queda fuera del alcance de este documento un análisis conceptual profundo, quedando a disposición del lector las referencias que son citadas para obtener más información.

2.1. Tratamiento de imágenes

2.1.1. Estereoscopía

La fotografía estereoscópica tradicional consiste en crear una ilusión 3D a partir de dos imágenes 2D. La forma más sencilla de crear en el cerebro la percepción de profundidad, es proporcionando a los ojos del espectador dos imágenes diferentes que representan dos perspectivas del mismo objeto, con una pequeña desviación (visión estéreo). Esto es análogo a las perspectivas que de forma natural reciben los ojos en la visión binocular.

La figura 1 muestra dos cámaras sacando una foto de una escena desde dos posiciones diferentes.

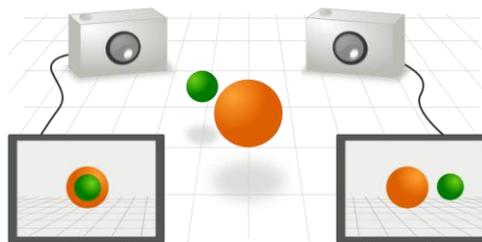


Figura 1 - Estereoscopía

2.1.2. Geometría Epipolar

La geometría epipolar describe matemáticamente la estereoscopía, define una relación entre un punto tridimensional y su proyección en dos dimensiones [1, 6, 7].

La figura 2 muestra dos cámaras mirando al punto X . O_L y O_R representan el centro de proyección del punto X en los planos de las imágenes. X representa el punto de interés en ambas cámaras. Los puntos X_L y X_R son las proyecciones de X en los planos de las imágenes.

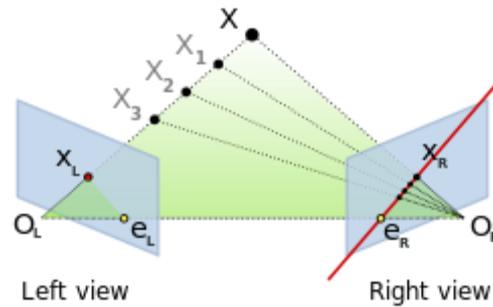


Figura 2 - Geometría epipolar

Como los centros de proyecciones de las cámaras son distintos, cada centro tiene su proyección en el plano de la imagen del otro. Estos dos puntos son e_L y e_R y se llaman puntos epipolares o epipolos. Como se observa en la imagen, los puntos O_L , O_R , e_L y e_R forman una recta.

La recta ($O_L - X$) que se ve como un punto en el plano de la cámara izquierda, se ve en la cámara derecha como una recta ($e_R - X_R$). Esta línea se llama línea epipolar; si hacemos la analogía con la cámara derecha, también se puede notar que existe una recta ($e_L - X_L$).

Como se aprecia, una línea epipolar es una función de un punto tridimensional X . Si se varía X en todo el espacio 3D, se obtiene un conjunto de líneas epipolares en ambas imágenes que hacen corresponder a cualquier X . Como $O_L - X$ pasa por O_L , la línea epipolar correspondiente en la imagen derecha pasa por el epipolo e_R . Esto significa que todas las líneas epipolares en la imagen derecha intersecan el epipolo e_R . Como consecuencia, todas las líneas que intersecan el epipolo son líneas epipolares que derivan de un punto X del espacio.

Otra forma de ver esto, es considerar los puntos X , O_L y O_R que forman un plano llamado plano epipolar. Si miramos la intersección de los dos planos de las cámaras con el plano epipolar obtenemos las líneas epipolares. Todos los planos epipolares y líneas epipolares intersecan el epipolo, no importa donde se encuentre X .

Viendo estos resultados se pueden sacar dos conclusiones útiles:

1) Si la proyección del punto X_L es conocida, entonces la recta ($e_R - X_R$) es conocida también y el punto X se proyecta en la imagen derecha en un punto X_R , el cual cae en una línea epipolar. Esto significa que para todo punto en una imagen, se debe poder observar su punto correspondiente en una línea epipolar conocida en la otra imagen. Esta hipótesis nos permite comprobar si dos puntos corresponden al mismo punto de la realidad.

2) Si los puntos X_L y X_R son conocidos, sus líneas de proyección son también conocidas. Si los dos puntos corresponden al mismo punto X de la realidad, las líneas de proyección deberían intersecarse en el punto X . Esto significa que X puede ser calculado a partir de las coordenadas de los puntos en ambas imágenes, este proceso se llama triangulación.

2.1.3. Rectificación de imágenes

Es una transformación que se ejecuta sobre las imágenes para alinear horizontalmente las líneas epipolares.

Como el estudio de imágenes estéreo usa triangulación basada en geometría epipolar para obtener la distancia de un objeto entre dos cámaras, se necesita encontrar los puntos que se corresponden en cada imagen (correlación). Dicha búsqueda hay que hacerla en una dimensión, sobre la línea epipolar. Al rectificar las imágenes, esta dimensión pasa a ser el eje horizontal.

En la figura 3, se ve un sistema con geometría epipolar sin rectificación (1), y el sistema rectificado (2). Este resultado, acelera mucho la búsqueda de la disparidad, por eso es recomendable que las imágenes estén rectificadas [8].

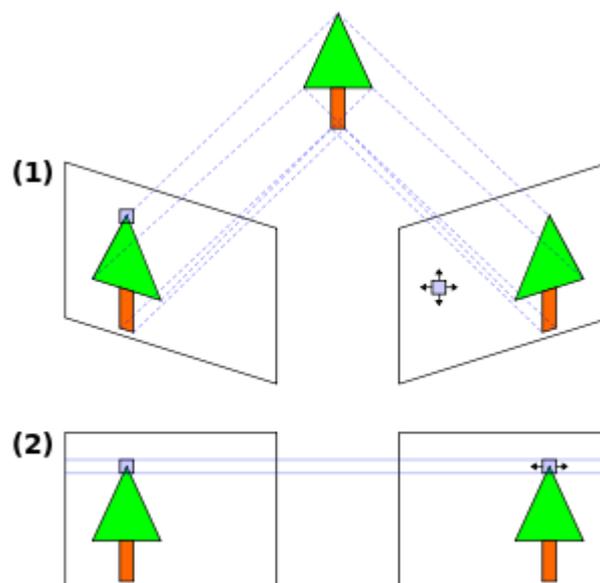


Figura 3 - Proceso de rectificación

2.1.4. Propiedades de las imágenes satelitales

A continuación se presentan las propiedades más relevantes relacionadas a las imágenes satelitales [13]:

Resolución espacial: Describe el nivel de detalle de una imagen. Por ejemplo, si las imágenes son de un metro de resolución, significa que cada píxel representa una distancia terrestre de un metro cuadrado. La desventaja de tener una alta resolución es que el archivo que contiene la imagen puede ser muy grande, lo que lo hace difícil de manejar. Para contrarrestar este problema se reduce el área de interés (*Area Of Interest, AOI*).

[4, 11] *Precisión espacial:* Es la precisión que se consigue al georreferenciar una imagen. Es una medida de la localización de un objeto en una imagen, con su localización verdadera. Para aumentar la precisión se hacen mejoras a las imágenes satelitales mediante estrategias como el rectificado u ortorrectificado de las imágenes, entre otras. La ortorrectificación corrige los desplazamientos de terreno, que generan muchos problemas

en zonas de relieve alto. Para determinar la precisión se compara un punto conocido de la tierra, llamado *ground control point* (GCP), con su correspondiente en la imagen. En la figura 4 se muestra la ubicación real de un GCP. El píxel que representa el punto en la imagen, se proyecta desde el satélite a la altura conocida del GCP, calculando así la ubicación aparente del píxel. La diferencia entre la ubicación real y la ubicación aparente es el error para ese píxel.

Para expresar la precisión se puede usar CE90 (*Circular Error of 90th percentile*), LE90 (*Linear Error of 90th percentile*), o RMSE (*Root Mean Square Error*). CE90 significa que existe una probabilidad del 90% de que el punto tenga un error horizontal de georreferenciación menor al valor dado en CE90. LE90 representa el error vertical de georreferenciación, análogo a CE90. RMSE se calcula como la raíz cuadrada del promedio de los errores al cuadrado.

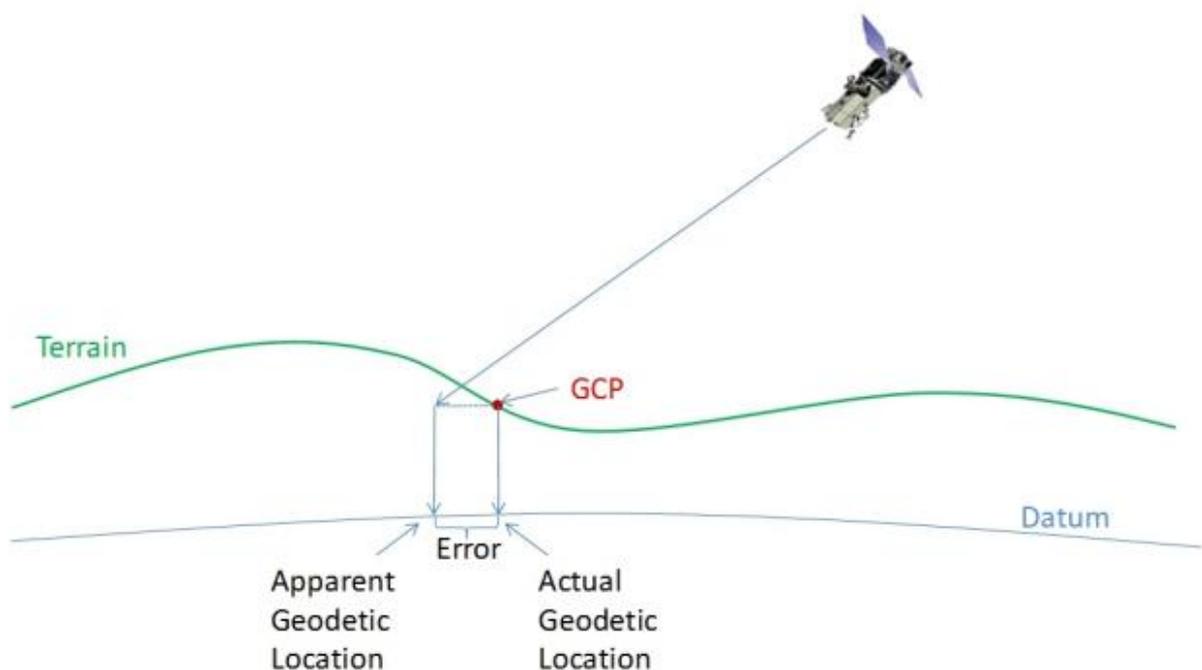


Figura 4 - Precisión sin corrección del terreno

[16] *Ángulo de nadir/Ángulo de elevación*: Obtener una imagen en nadir sería sacar una foto exactamente sobre un objeto (90 grados “*elevation angle*” o 0 grados “*Off-nadir*”). El ángulo de elevación típico es de 60° (lo que corresponde a un ángulo *off-nadir* de 30°), un mayor ángulo de elevación es deseable para minimizar las oclusiones, es decir edificios que no aparecen en las imágenes, por ejemplo cuando el área de interés contiene relieve alto o edificios altos. Generalmente, las imágenes estereoscópicas son tomadas sobre la misma órbita, una hacia adelante (*forward*) y la otra hacia atrás (*backward*). El resultado es que ambas mejoran la calidad de la imagen, ya que un corto intervalo de tiempo entre las dos imágenes resulta en las mismas condiciones de iluminación, contenido de la escena y mejor precisión espacial. En adelante, se denomina *imagen de referencia* a la imagen más próxima al nadir e *imagen secundaria* a la otra. En la figura 5 se muestran el ángulo de elevación así como el ángulo azimuth, estos dos definen la posición exacta del satélite.

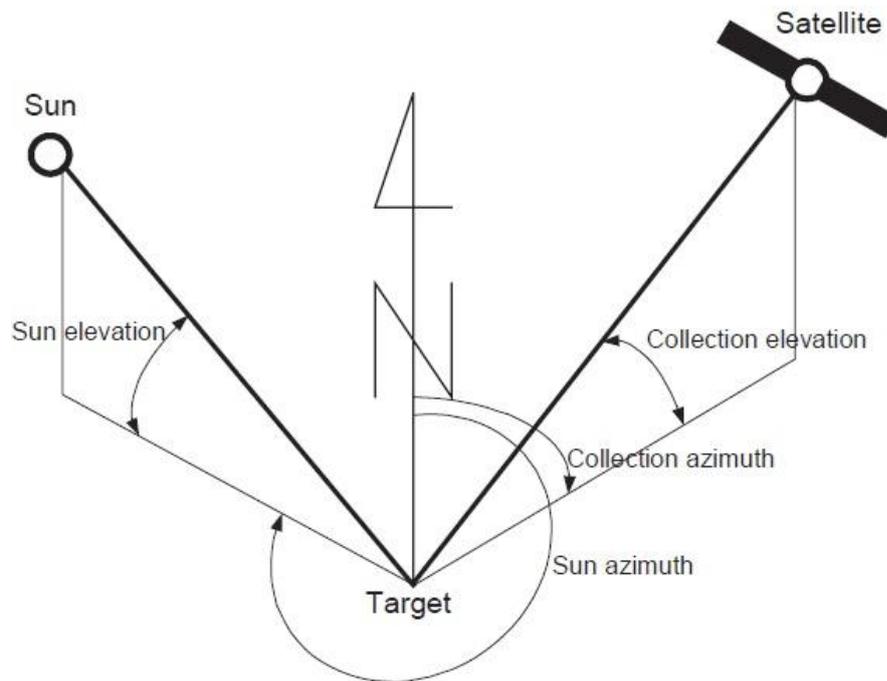


Figura 5 - Geometría de la imagen adquirida

Elevación del sol: La posición del sol es relevante, porque si éste se encuentra con poca altitud la imagen sale oscura; una posición típica es de 30 grados de elevación. El otro problema que genera el sol son las sombras, las cuales dan problemas para hallar la correspondencia de los píxeles en cada imagen.

[12] *Información espectral (bandas, bits de profundidad, ajustes de bandas):* Cada banda de una imagen corresponde a diferente longitud de onda de la luz. La mayoría de los satélites captan imágenes pancromáticas con una alta resolución. Este tipo de imágenes captan toda la parte visible e infrarrojo cercano del espectro; los datos pancromáticos se visualizan en escala de grises. Además, se captan imágenes multiespectrales de cuatro bandas (rojo, verde, azul e infrarrojo) con una resolución menor a las anteriores. Las tres bandas (rojo, verde y azul) son para ver los verdaderos colores. La infrarroja sirve, por ejemplo, para el análisis de vegetación. Los bits de profundidad de la mayoría de los satélites es de 11 bits, esto significa que los datos alcanzan un rango entre 0 y 2047 por banda, también se ofrecen de 8 bits. Los ajustes de bandas sirven para calibrar el contraste o el brillo. Un algoritmo conocido que se usa para ajustar es *dynamic range adjustment* (DRA).

2.2. Disparidad

El término disparidad fue inicialmente introducido para explicar la visión del ojo humano y describe el desplazamiento de un punto físico cuando se ve con el ojo izquierdo y luego con el derecho. La disparidad se puede dividir en horizontal y vertical [8].

2.2.1. Tipos de algoritmos

Los algoritmos para el cálculo de disparidad se pueden dividir en dos: locales y globales. Todos los algoritmos tratan de buscar la correspondencia de un píxel en una imagen de referencia con un píxel de la otra imagen. Los algoritmos que son locales, se basan en buscar la correspondencia en un pequeño número de píxeles alrededor del píxel de interés. En cambio, si se busca la correspondencia en una línea o en toda la imagen se dice que es del tipo global.

Algunos algoritmos locales son:

- *Block Matching*
- *Gradient Based Optimization*
- *Feature Matching*

Algunos algoritmos globales son:

- *Dynamic Programming*
- *Intrinsic Curves*
- *Graph Cuts*
- *Nonlinear Diffusion*
- *Belief Propagation*
- *Correspondenceless Method*

Para profundizar en estos algoritmos ver referencia [2].

Los métodos locales son eficientes, pero tienen como desventaja su sensibilidad a las ambigüedades de las regiones, por ejemplo en caso de haber oclusiones. Los métodos globales toman más información del resto de la imagen, lo que hace que sean mucho más costosos computacionalmente.

En el trabajo se explica el algoritmo “*Block Matching*”, ya que es el que se utiliza en el sistema. Este método busca estimar la disparidad en un punto de una imagen, comparando una región pequeña alrededor de ese punto con una serie de pequeñas regiones en la otra imagen. Como se mencionó anteriormente, agregar la restricción de que las imágenes estén en geometría epipolar reduce la búsqueda a una dimensión [10].

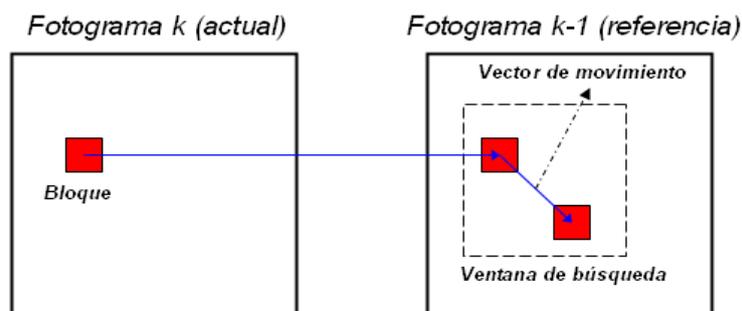


Figura 6 - Block Matching

2.2.2. Taxonomía de algoritmos para cálculos estéreo

La taxonomía definida en la publicación “*A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*” [3], se basa en la observación de que los algoritmos estéreo generalmente recorren un subconjunto de los siguientes cuatro pasos:

1. *Se elige el algoritmo de costo (matching cost computation)*
Estos pueden ser por ejemplo: *squared intensity differences (SD)*, *absolute intensity differences (AD)* y *normalized cross-correlation (NCC)*.
2. *Se elige un algoritmo de agregación (cost aggregation)*
Los algoritmos de agregación pueden ser locales o basados en ventanas, estos le dan soporte a los algoritmos de costos sumando o promediando los valores en una región.
3. *Se halla la disparidad (disparity computation / optimization)*
Los métodos de disparidad se dividen en dos: los locales y los globales. Los locales simplemente eligen para cada píxel la disparidad asociada con el valor mínimo del costo. Una limitación de este enfoque (y muchos otros algoritmos de correspondencia) es que la unicidad de correspondencia sólo se aplica para una imagen (la imagen de referencia), mientras que un punto en la otra imagen podría quedar emparejado a múltiples puntos. Por otro lado, la mayoría de los métodos globales tienen como objetivo encontrar una función de disparidad que minimice una energía global. Una vez encontrada la función, se utiliza algún algoritmo para hallar el mínimo local.
4. *Se refinan los resultados (disparity refinement)*
La mayoría de los algoritmos estéreo computan un conjunto de disparidades estimadas en un espacio discreto. Sin embargo, para imágenes, es mejor tener un resultado más visible, para lo cual se usa algún tipo de refinamiento. Los más comunes son: *sub-pixel*, *cross-checking* (comparar la imagen de referencia con la secundaria y al revés) y filtro por mediana para limpiar valores que son erróneos.

Para ver esta taxonomía se ejemplifica con el algoritmo SSD:

1. La función de costo, es la diferencia al cuadrado de los valores de intensidad para una disparidad dada.
2. La operación de agregación está dada por la suma de las funciones de costo en una ventana cuadrada con disparidad constante.
3. La disparidad se obtiene seleccionando el mínimo de los valores agregados en cada píxel.

2.3. Cálculo de alturas y georreferenciación

Esta sección define los tipos de alturas y las diferentes formas encontradas para el cálculo de las mismas. Se evalúan dos formas de obtención: una mediante el modelo RPC; y la otra mediante el dato del *baseline*.

2.3.1. Definiciones de las diferentes alturas

Altura elipsoidal (h): Es la distancia vertical de un punto de la superficie terrestre con respecto a la superficie del elipsoide de referencia (en este caso WGS 84⁵).

Altura ortométrica (H): Es la distancia vertical entre un punto de la superficie terrestre y la superficie del geoide, llamada también altura sobre el nivel del mar. Geoide es una aproximación a la forma real de La Tierra, la cual es difícil de describir matemáticamente debido a las irregularidades de las superficies locales y las variaciones en el lecho marino. La palabra geoide significa “forma de la Tierra”.

En la figura 7, se muestra gráficamente los conceptos de altura elipsoidal y ortométrica. A lo largo del informe, cuando se menciona altura se refiere a altura elipsoidal y se encuentra expresada en metros (m).

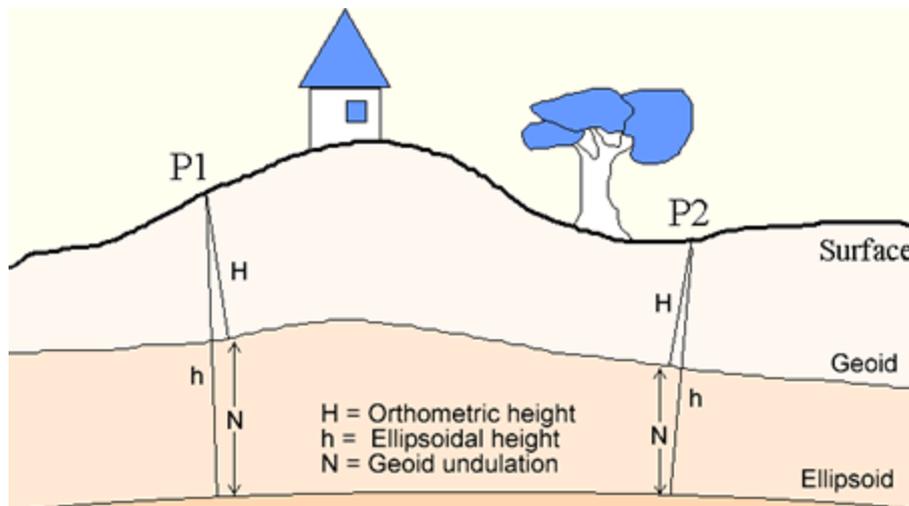


Figura 7 - Altura elipsoidal y ortométrica

2.3.2. Modelo RPC

El modelo *Rational Polynomial Camera* (en adelante RPC) relaciona las coordenadas del espacio del objeto (latitud, longitud y altura), con las coordenadas del espacio de la imagen (*line* y *sample*). Para comprenderlo mejor, *sample* corresponde al eje x y *line* corresponde al eje y [5,16].

La relación objeto-imagen para cada imagen estéreo, es expresada por un modelo RPC *sample* y un modelo RPC *line*. Para determinar las coordenadas 3D en el espacio

⁵ WGS84 es un sistema de coordenadas geográficas, por sus siglas en inglés **World Geodetic System 84**

objeto de un punto dado, se necesita medir sus coordenadas *sample* y *line* en ambas imágenes (ver figura 8).

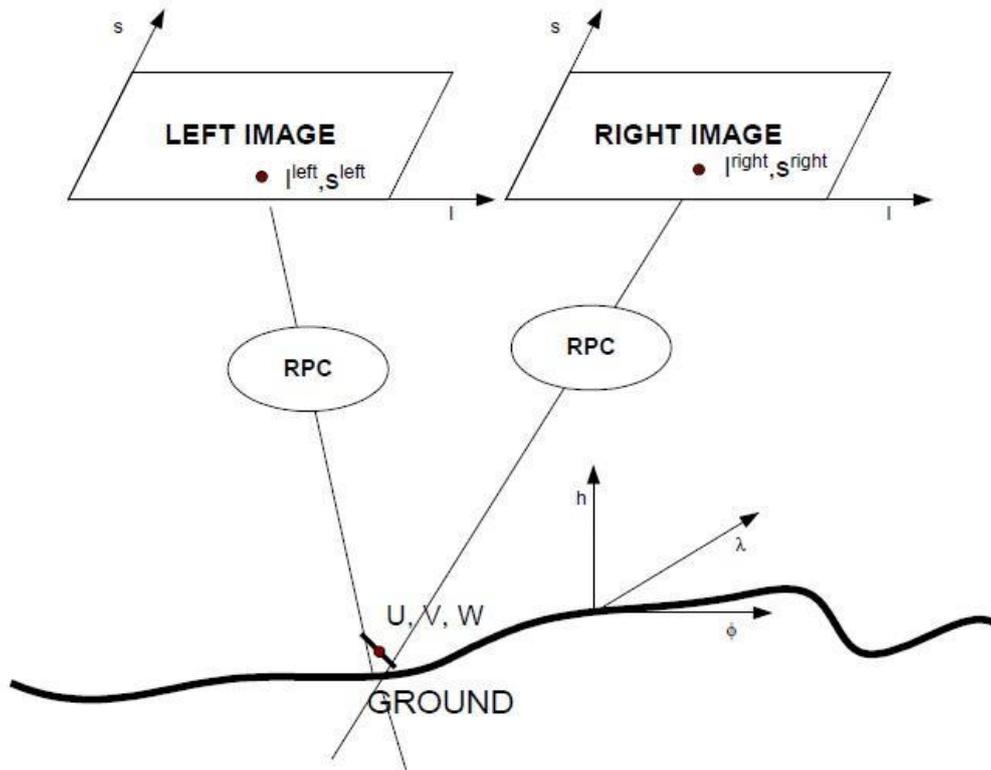


Figura 8 - Extracción de características estereó

U, V, W corresponden a los valores normalizados de latitud, longitud y altura respectivamente (φ, λ, h) , para desnormalizar se utiliza:

$$U = (\varphi - O_{\varphi})/SF_{\varphi}$$

$$V = (\lambda - O_{\lambda})/SF_{\lambda}$$

$$W = (h - O_h)/SF_h$$

Donde $SF_{\lambda}, SF_{\varphi}, SF_h$ son los factores de escala y $O_{\lambda}, O_{\varphi}, O_h$ son los valores promedios de latitud, longitud y altura respectivamente:

$$SF_{\lambda} = \max(|\lambda_{\max} - O_{\lambda}|, |\lambda_{\min} - O_{\lambda}|)$$

$$SF_{\varphi} = \max(|\varphi_{\max} - O_{\varphi}|, |\varphi_{\min} - O_{\varphi}|)$$

$$SF_h = \max(|h_{\max} - O_h|, |h_{\min} - O_h|)$$

$$O_{\lambda} = \frac{1}{n} \sum \lambda$$

$$O_{\varphi} = \frac{1}{n} \sum \varphi$$

$$O_h = \frac{1}{n} \sum h$$

Se definen l, s como los valores normalizados en las coordenadas del espacio de la imagen (L,S) donde L,S corresponden a las coordenadas de *line* y *sample*, respectivamente:

$$l = (L - O_L)/SF_L$$

$$s = (S - O_S)/SF_S$$

Donde SF_L, SF_S son los valores de escala y O_L, O_S son los valores promedio de *line* y *sample* respectivamente:

$$SF_L = \max(|L_{\max} - O_L|, |L_{\min} - O_L|)$$

$$SF_S = \max(|S_{\max} - O_S|, |S_{\min} - O_S|)$$

$$O_L = \frac{1}{n} \sum L$$

$$O_S = \frac{1}{n} \sum S$$

El modelo funcional RPC es el cociente de dos funciones cúbicas de las coordenadas del objeto. Se utilizan dos funciones racionales para expresar las relaciones entre el espacio del objeto y las coordenadas *line* (1) y *sample* (2).

El modelo RPC para *line* está dado por:

$$(1) l = \frac{Num_L(U,V,W)}{Den_L(U,V,W)}$$

Donde:

$$Num_L(U,V,W) = a_1 + a_2 \cdot V + a_3 \cdot U + a_4 \cdot W + a_5 \cdot V \cdot U + a_6 \cdot V \cdot W + a_7 \cdot U \cdot W + a_8 \cdot V^2 + a_9 \cdot U^2$$

$$+ a_{10} \cdot W^2 + a_{11} \cdot U \cdot V \cdot W + a_{12} \cdot V^3 + a_{13} \cdot V \cdot U^2 + a_{14} \cdot V \cdot W^2 + a_{15} \cdot V^2 \cdot U + a_{16} \cdot U^3 + a_{17} \cdot U \cdot W^2$$

$$+ a_{18} \cdot V^2 \cdot W + a_{19} \cdot U^2 \cdot W + a_{20} \cdot W^3$$

$$Den_L(U,V,W) = b_1 + b_2 \cdot V + b_3 \cdot U + b_4 \cdot W + b_5 \cdot V \cdot U + b_6 \cdot V \cdot W + b_7 \cdot U \cdot W + b_8 \cdot V^2 + b_9 \cdot U^2$$

$$+ b_{10} \cdot W^2 + b_{11} \cdot U \cdot V \cdot W + b_{12} \cdot V^3 + b_{13} \cdot V \cdot U^2 + b_{14} \cdot V \cdot W^2 + b_{15} \cdot V^2 \cdot U + b_{16} \cdot U^3 + b_{17} \cdot U \cdot W^2$$

$$+ b_{18} \cdot V^2 \cdot W + b_{19} \cdot U^2 \cdot W + b_{20} \cdot W^3$$

Y el modelo RPC para *sample* está dado por:

$$(2) s = \frac{Num_S(U,V,W)}{Den_S(U,V,W)}$$

Donde:

$$Num_S(U,V,W) = c_1 + c_2 \cdot V + c_3 \cdot U + c_4 \cdot W + c_5 \cdot V \cdot U + c_6 \cdot V \cdot W + c_7 \cdot U \cdot W + c_8 \cdot V^2 + c_9 \cdot U^2$$

$$+ c_{10} \cdot W^2 + c_{11} \cdot U \cdot V \cdot W + c_{12} \cdot V^3 + c_{13} \cdot V \cdot U^2 + c_{14} \cdot V \cdot W^2 + c_{15} \cdot V^2 \cdot U + c_{16} \cdot U^3 + c_{17} \cdot U \cdot W^2$$

$$+ c_{18} \cdot V^2 \cdot W + c_{19} \cdot U^2 \cdot W + c_{20} \cdot W^3$$

$$Den_S(U,V,W) = d_1 + d_2 \cdot V + d_3 \cdot U + d_4 \cdot W + d_5 \cdot V \cdot U + d_6 \cdot V \cdot W + d_7 \cdot U \cdot W + d_8 \cdot V^2 + d_9 \cdot U^2$$

$$+ d_{10} \cdot W^2 + d_{11} \cdot U \cdot V \cdot W + d_{12} \cdot V^3 + d_{13} \cdot V \cdot U^2 + d_{14} \cdot V \cdot W^2 + d_{15} \cdot V^2 \cdot U + d_{16} \cdot U^3 + d_{17} \cdot U \cdot W^2$$

$$+ d_{18} \cdot V^2 \cdot W + d_{19} \cdot U^2 \cdot W + d_{20} \cdot W^3$$

Los coeficientes del modelo RPC a_i , b_i , c_i y d_i son conocidos y vienen dados en la metadata de las imágenes.

Esto resulta en el siguiente conjunto de ecuaciones:

$$l^{\text{left}} = \frac{\text{Num}_L^{\text{left}}(U, V, W)}{\text{Den}_L^{\text{left}}(U, V, W)}$$

$$s^{\text{left}} = \frac{\text{Num}_S^{\text{left}}(U, V, W)}{\text{Den}_S^{\text{left}}(U, V, W)}$$

$$l^{\text{right}} = \frac{\text{Num}_L^{\text{right}}(U, V, W)}{\text{Den}_L^{\text{right}}(U, V, W)}$$

$$s^{\text{right}} = \frac{\text{Num}_S^{\text{right}}(U, V, W)}{\text{Den}_S^{\text{right}}(U, V, W)}$$

Donde s^{left} , l^{left} , s^{right} , l^{right} son medidas normalizadas de un punto en la imagen de referencia y secundaria respectivamente, en el espacio de coordenadas de la imagen. U , V , W son las incógnitas normalizadas en el espacio de coordenadas del objeto. Como son 4 ecuaciones y 3 incógnitas, el sistema de ecuaciones queda sobredeterminado. El criterio de mínimos cuadrados puede ser utilizado para encontrar la solución.

Estimación de coeficientes RPC del satélite IKONOS

Una aproximación por mínimos cuadrados se utiliza para determinar los coeficientes del modelo RPC a_i , b_i , c_i y d_i . La grilla 3D de puntos objeto se genera proyectando los puntos desde la grilla 2D de puntos imagen e intersecando con un número de planos de elevación constantes (ver figura 9). El proceso de estimación, fue ejecutado independientemente para cada uno de los modelos RPC (*line* y *sample*).

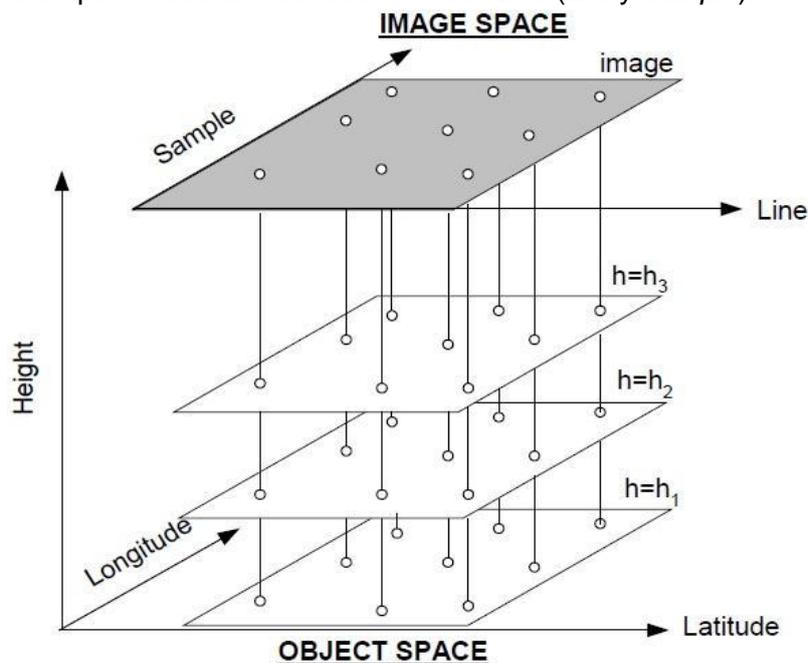


Figura 9 - Generación RPC

2.3.3. Baseline

Un proceso para obtener información de altura de los objetos en una escena es mediante el *baseline*, el cual se define como la distancia que hay entre dos centros ópticos. Este proceso involucra varios pasos, el más difícil es hacer corresponder un punto de una imagen con la otra (*matching*), luego de esto con un simple cálculo geométrico se obtiene la altura [8].

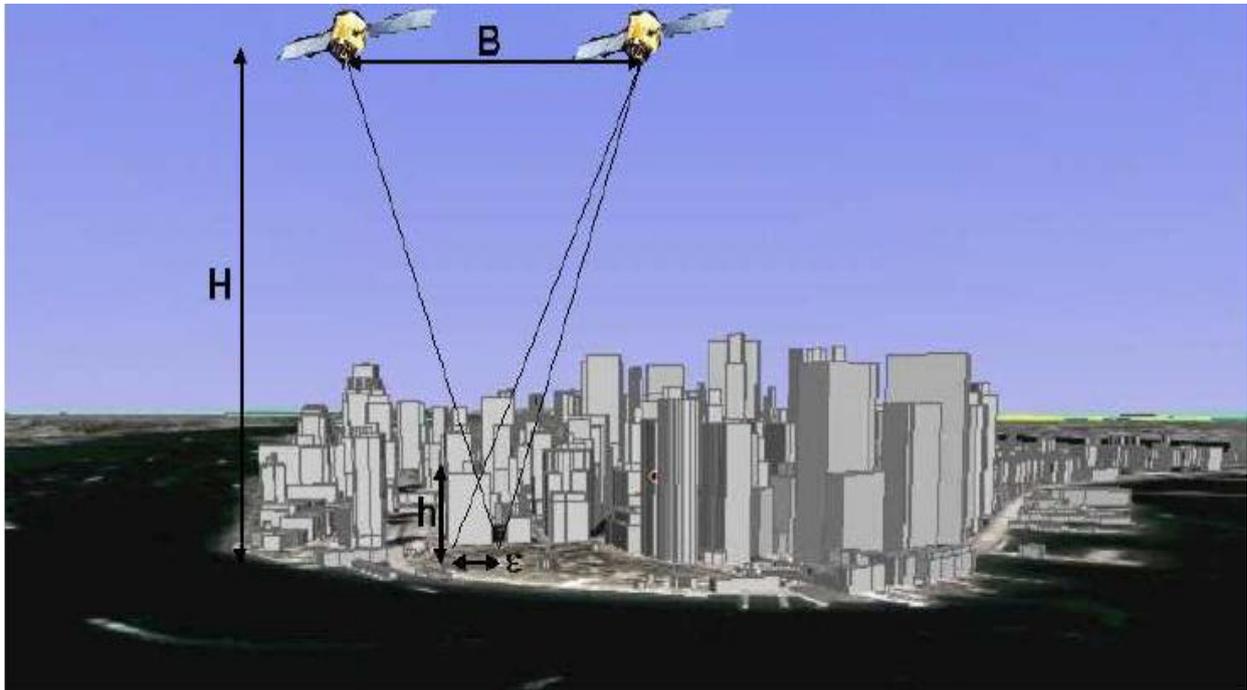


Figura 10 - Cálculo mediante baseline

Viendo la figura 10 de arriba se puede obtener esta ecuación:

$$\epsilon = \frac{B}{H} h$$

Donde ϵ es la disparidad (mencionado en la sección 2.2), B es el baseline y H es la altura de las cámaras. Viendo esta fórmula, se puede apreciar que la relación B/H tiene una gran influencia en la precisión con la que se puede obtener la altura. Como H es constante y es muy grande, conviene tener un B grande para obtener un resultado más preciso. Sin embargo, esto genera otros problemas, por ejemplo: al transcurrir unos minutos por el movimiento del satélite entre una toma y la otra, los objetos móviles como autos o sombras también se mueven; el otro gran problema es que hay objetos que se pueden visualizar en una imagen pero en la otra no y viceversa, esto se debe a las oclusiones.

3. Análisis

Al comienzo del proyecto fueron necesarias varias reuniones con el cliente para definir claramente los requerimientos de la aplicación a implementar, una vez definidos se obtuvo el documento de especificación de requerimientos (Anexo I).

Parte del proyecto consistió en la realización de pruebas, evaluación de las distintas alternativas y evaluación de su factibilidad de llevarse a cabo, para posteriormente sugerir al cliente la mejor solución. Se encontró una aplicación similar en el mercado: CityModeler. De las pruebas con esta aplicación, se destaca el diseño de interacción que posee como programa de procesamiento de imágenes. Esto derivó en la adquisición por parte del cliente de una versión, que fue utilizada como referencia para la interfaz de usuario y validaciones.

Por otra parte se cuenta con imágenes satelitales estereoscópicas del satélite IKONOS de la empresa DigitalGlobe⁶, adquiridas de Landinfo⁷. Las especificaciones de dicho satélite y otros que hay en el mercado se pueden ver en el anexo IX. Las imágenes son de la línea GeoStereo, pancromáticas y de 16 bits de profundidad, de los cuales sólo los 11 bits menos significativos tienen datos [9].

3.1. Descripción del sistema

Se espera que el sistema a implementar tenga las siguientes funcionalidades imprescindibles:

1. Visualización de las imágenes satelitales, reconociendo tanto la información gráfica contenida, así como la metadata que acompaña a las mismas.
2. Extracción de la altura de un edificio individual a partir del señalamiento de su contorno (en adelante *outline*) en ambas imágenes por parte del usuario. Es decir, que la herramienta debe ser capaz de deducir la altura del edificio, asumiendo que la ubicación indicada por el usuario en ambas imágenes es correcta. Para esta funcionalidad, la herramienta debe tener una interfaz que permita al usuario delinear el *outline* del edificio sobre la imagen mediante polilíneas u otras entidades como arcos.
3. Generación y visualización del modelo 3D generado a partir de los edificios procesados.
4. Exportación de los datos calculados en formatos adecuados, que sean compatibles con otras aplicaciones.

Adicionalmente, es deseable que el sistema incorpore las siguientes funcionalidades, listadas en orden decreciente de prioridad y orden creciente de dificultad esperada en su implementación:

- A. A partir del señalamiento del *outline* de un edificio en la imagen de referencia la herramienta identifica automáticamente en la imagen secundaria el edificio correspondiente.

⁶ (2013, Noviembre) Digital Globe [Online] <http://www.digitalglobe.com/>

⁷ (2013, Noviembre) Land Info [Online] <http://www.landinfo.com/>

- B. La herramienta es capaz de leer archivos de *outlines* de edificios, permitiendo la carga masiva de *outlines*.
- C. La herramienta es capaz de identificar en forma semi-automática *outlines* de edificios en la imagen de referencia mediante técnicas de detección de bordes.

3.2. Requerimientos del sistema

A continuación se detallan los requerimientos funcionales y no funcionales del sistema, junto con la documentación requerida. Estos fueron relevados y validados con el cliente.

3.2.1. Requerimientos funcionales

Los requerimientos funcionales refieren a las funciones que el sistema debe proveer. Describen la interacción entre el sistema y un actor, este último puede ser un usuario o un sistema externo, independientemente de su implementación.

Se enumeran los requerimientos funcionales del sistema:

a) Cargar par de imágenes

Se debe permitir cargar el par de imágenes (imagen de referencia y secundaria) y visualizarlas, además de mantener una relación visual entre ellas.

b) Seleccionar contorno (*outline*)

Se debe permitir seleccionar el contorno de un edificio y automáticamente sugerir el contorno en la imagen secundaria.

c) Obtener altura de un edificio seleccionado

Dado un contorno seleccionado, el programa debe permitir calcular la altura del mismo, así como también latitud y longitud.

d) Generar modelo 3D completo

Se debe permitir generar un modelo 3D de todos los edificios seleccionados.

e) Visualizar modelo

Se debe permitir visualizar el modelo 3D generado.

f) Exportar modelo 3D generado

Se debe permitir exportar el modelo en algún formato compatible con programas de tipo CAD, se elige como formato: STL.

g) Importar contornos

Se debe permitir importar contornos a partir de un archivo con formato predefinido.

h) Exportar contornos

Se debe permitir exportar los contornos a partir de un archivo con formato predefinido.

3.2.2. Requerimientos no funcionales

Los requerimientos no funcionales son las restricciones que el sistema debe contemplar. Estas restricciones limitan las elecciones a la hora de construir la solución. Como lo indica su nombre, estos requerimientos no se refieren directamente a las funciones del sistema. Algunos de estos pueden ser: rendimiento, lenguaje, interoperabilidad, entre otros.

A continuación se detallan los requeridos por el cliente:

a) Tiempos de ejecución de la solución final

Dependiendo del tamaño de las imágenes a procesar y la capacidad de procesamiento del computador (o clúster) donde se ejecute, los tiempos de ejecución estarán acotados. Para toda la ciudad de Montevideo, el tiempo de ejecución deberá ser menor a 1 mes. En caso de ser necesario se dispone de un clúster que cuenta con 72 GB de memoria RAM, para el procesamiento de las imágenes.

b) Métodos de procesamiento extensibles fácilmente

Permitirá incluir nuevos métodos de procesamiento, los cuales deben ser fácilmente incorporados al sistema.

c) Sistemas operativos soportados por la aplicación

Soportará Windows 7 32/64 bits. Si se decide utilizar el clúster el sistema debe poder ejecutarse en Linux.

d) Lenguaje

Se debe programar en Matlab, debido a que el cliente tiene un alto conocimiento en este lenguaje. El sistema puede ser un *toolbox*⁸ para Matlab o una aplicación de escritorio.

⁸ Un *toolbox* es un módulo que extiende una aplicación existente sin la necesidad de modificarla.

e) Código claro, legible

La implementación deberá tener código claro y legible para su fácil mantenimiento y escalabilidad.

f) Error

Dada la precisión de 1 m. de las imágenes, se tolerará un error de hasta 10% en el cálculo de la altura.

3.2.3. Requerimientos de documentación

En esta sección se especifica el tipo de documentación que se requiere, el contenido y formato de la misma.

a) Documentación de los métodos

Se deberá brindar una documentación clara de cómo configurar los métodos a utilizar y los parámetros necesarios de cada uno de los mismos.

b) Manual de usuario

Se deberá brindar un manual de usuario correspondiente a la última versión liberada del sistema, detallando las distintas funcionalidades u opciones que provee, indicando qué hace y cómo funciona cada una de ellas.

c) Manual de instalación

Se deberá brindar un manual de instalación que indique precisa y detalladamente los pasos que se deben seguir para instalar la aplicación en los distintos sistemas operativos (sección 3.2.2.c). Además, se debe indicar los parámetros a nivel de sistema que se requieran configurar.

4. Planteo de la solución

Este capítulo describe la solución propuesta. Se brinda una descripción general de la solución, para luego analizar detalladamente aquellos aspectos que tomaron mayor relevancia en el transcurso del proyecto. Se detallan las decisiones tomadas, indicando el método elegido para hallar la disparidad, procedimiento para el modelado de los techos, algoritmo usado para el cálculo de altura y formatos de exportación e importación; considerando las que mejor se adecúan al problema dadas las condiciones.

4.1. Descripción General

Se decide implementar una aplicación de escritorio en Matlab⁹, llamada **Stereog3mas**¹⁰ (sg3), que permita cargar el par de imágenes satelitales estereoscópicas y seleccionar el *outline* de un edificio para calcular su altura, latitud y longitud. Para lograr esto, se divide el problema en subproblemas. El primero, es el cálculo de la disparidad, para el cual se decide utilizar el método NCC. El segundo, es el modelado de techos, en el cual se aproxima el techo de un edificio a un plano horizontal, se opta por aplicar la mediana de los puntos internos del *outline*. Por último, se elige el modelo RPC para el cálculo de las alturas y la georreferenciación.

Se decide permitir exportar e importar los *outlines* y exportar el modelo 3D generado. Para el primer caso, se elige el formato XML, por ser fácil de manipular en distintos programas. En el segundo caso, para la exportación del modelo 3D, se utiliza el formato STL por ser compatible con programas de tipo CAD¹¹ del mercado.

4.2. Disparidad

El mapa de disparidad muestra el desplazamiento de cada píxel de una imagen a la otra. Si se considera que sólo existe disparidad horizontal (como se menciona en sección 2.1.3), dada la imagen de referencia I y la secundaria I' , sea D el mapa de disparidad, entonces:

$$I(i,j) = I'(i + d_{ij}, j) \text{ donde } d_{ij}=D(i,j)$$

Las herramientas encontradas para el cálculo de la disparidad fueron: IPOL¹², Stereomatch y Orfeo toolbox (OTB). A continuación se presentan:

⁹ Matlab es una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio.

¹⁰ <http://stereog3mas.webs.com/>

¹¹ CAD se refiere a Diseño Asistido por Computadora (por sus siglas en inglés)

¹² (2013, Noviembre) Image Processing On Line, Block Matching [Online]
http://dev.ipol.im/~facciolo/ipol_demo/f_stereo_pipe/

4.2.1 IPOL

Image Processing On Line (IPOL) es una revista online de procesamiento y análisis de imágenes que describe diferentes algoritmos y sus implementaciones. Esta herramienta es muy útil, ya que no requiere de la instalación de ninguna aplicación ni extensión en el navegador web. Además, es muy ágil a la hora de procesar las imágenes para obtener el mapa de disparidad. Brinda la posibilidad de probar los diferentes métodos para posteriormente compararlos y analizar cuál de ellos se adapta mejor al tipo de imágenes que se poseen.

Procedimiento para obtener mapa de disparidad correspondiente a un par de imágenes

En primer lugar se debe elegir un par de imágenes para procesar (existe un límite en el tamaño de cada archivo de 300 MB). Luego se deben indicar los parámetros de entrada y seleccionar los métodos con los cuales se desea procesar las imágenes. Por ejemplo, se debe indicar un rango de disparidad y el tamaño de la ventana (*window size*). Para “*Block Matching*” se puede elegir entre: SSD, SAD, SOD, NCC, entre otros. En “*Decision Methods*” se puede activar: LR - RL, LR Difference, RL Occlusions, MinDiff, entre otros. Para “*Filtering*” se puede activar: min-filter on non-rejected, min-filter, Subpix refinement y RAFA DAR refinement. Posteriormente, se presiona el botón “*run*” y comienza la ejecución. En un máximo de 30 segundos se obtiene como resultado el mapa de disparidad. En la figura 11, se pueden ver los parámetros de entrada gráficamente.

← → ↻ dev.ipol.im/~facciolo/ipol_demo/f_stereo_pipe/input_select?Corridor_bonn.x=88&Corridor_bonn.y=48#

Block matching stereo - development

[article](#) [demo](#) [archive](#)

This is a preprint. It may change before it is accepted for publication.

If you use this algorithm's code or demo results, please cite it.

Preprocess images

Add noise to input σ Convert input to monochrome

Parameters of the pair

Disparity range [,]px Image noise σ (0.28 for quantization)

Block matching

- SSD [?]
- SSD exhaust. [?]
- SAD [?]
- SOD [?]
- SSD - mean [?]
- SSD/Mean [?]
- SSD/Std [?]
- SSD/Norm [?]
- SSD - mean, shiftable win. [?]
- NCC [?]
- SSD-BT [?]
- PCA match [?]
- MARC2 [?]

Subpixel [?] 1 1/2 1/4 1/8

Window size

Decision methods

- LR - RL (reciprocity) [?]
- LR - RL (TWO PASS) [?]
- LR Difference [?]
- RL Occlusions [?]
- MinDiff (min-filter diffs) [?]
- MinDiff dilated 1px [?]
- MinMax [?]
- MinDiff subpix [?]
- Flat patches [?]
- Flat(Hor.) patches [?]
- Flat patches dilated [?]
- Flat(Hor.) patches dilated [?]
- Disparity error estimation [?]
- 1/2 pixel shift [?]
- s/2 pixel shift (subpix) [?]
- 2nd minima (strobe) [?]
- Self-Similarity [?]
- Symmetric Self-Similarity [?]
- A-Contrario (ACBM) [?]
- ACBM+SelfSim (ACBMSS) [?]
- NANS [?]

Filtering

- min-filter on non-rejected (with checked decision methods)
- min-filter
- Subpix refinement (SSD)
- RAFA DAR refinement (with checked decision methods)

Figura 11 - IPOL

4.2.2. StereoMatch

Es un programa implementado por estudiantes de la Universidad de Middlebury¹³, como ejemplo de aplicación de la taxonomía explicada en la sección 2.2.2, para el cálculo de mapa de disparidad presentada en el artículo publicado por los mismos autores. Recibe como entrada un par de imágenes de ejemplo que se pueden descargar del mismo sitio donde se descargan los archivos fuentes, se elige el método que se desea ejecutar en conjunto con otros parámetros y la salida es un mapa de disparidad. Al igual que el software de IPOL se utilizó para comparar los diferentes algoritmos de *block matching* existentes.

4.2.3. Orfeo toolbox

Orfeo es una biblioteca desarrollada por la agencia espacial francesa CNES¹⁴, la misma provee un conjunto de algoritmos los cuales permiten procesar imágenes espaciales. OTB es distribuido como una biblioteca *open source* [14, 15]. Algunas operaciones que proporciona Orfeo son las siguientes:

- *StereoRectificationGridGenerator*: Esta operación crea la matriz de deformación de las imágenes, para construir una imagen epipolar rectificada. Se debe conocer la geometría de la adquisición.
- *GridBasedImageResampling*: Esta operación permite deformar una imagen, un ejemplo de uso sería: en conjunto con la operación anterior sirve para obtener las imágenes rectificadas.
- *BlockMatching*: Esta operación calcula la correlación entre bloques de las dos imágenes, para esto se puede usar una serie de medidas de similitud como pueden ser: NCC, SSD, Lp pseudo-norm (Lp), entre otros. Algunos parámetros que recibe esta operación y fueron utilizados son:
 - i. Minhd, maxhd: Es el rango de la disparidad horizontal, esto permite que el algoritmo sea más eficiente, ya que busca hacer la correspondencia solamente en el rango ingresado.
 - ii. Outmetric: Cuando se activa este parámetro hace que el algoritmo devuelva el grado de confiabilidad para cada cálculo.
 - iii. Radius: Este parámetro corresponde a la mitad del tamaño de la ventana en donde se hace la búsqueda, en píxeles.
- *DisparityMapToElevationMap*: Esta operación permite pasar de un modelo de disparidad a uno de elevación mediante triangulación.

Algoritmo de disparidad elegido

En la sección 2.2 se enumeraron los distintos métodos para hallar el valor de la disparidad. Luego de experimentar con estos algoritmos en la herramienta OTB, recibiendo como entrada el área de interés de las imágenes IKONOS, NCC es el que da mejores resultados comparado con la disparidad real (por más información ver sección 6.1). NCC es una medida de similitud, la cual se elige como función de costo para el algoritmo *block matching*, se basa en el medida SSD pero con una ventaja, SSD usa como valor el nivel de

¹³ (2013, Noviembre) Universidad de Middlebury [Online] <http://vision.middlebury.edu/stereo/>

¹⁴ (2013, Noviembre) Centro Nacional de Estudios Espaciales, Francia [Online] <http://www.cnes.fr>

gris de las imágenes, lo que hace que sea sensible a las diferencias radiométricas entre las dos imágenes (las mediciones son afectadas por el cambio de contraste). Para corregir este problema, NCC utiliza valores normalizados.

4.3. Modelado de techos

Dado un contorno en la imagen de referencia, el problema que se plantea es hallar el contorno correspondiente en la otra imagen. Para esto se utiliza el mapa de disparidad, el inconveniente que se presenta es que al ser un mapa no denso, no está definida la disparidad para todos los puntos. Cuando esto pasa se dice que el mismo tiene ruido. Por lo tanto, no se puede definir un plano de forma inmediata, para solucionar este inconveniente lo que se hace es aproximar el techo a un plano horizontal, estimando las disparidades que tienen ruido.

Métodos para mejorar *outline* propuesto en la imagen secundaria

Los siguientes métodos toman como hipótesis que los techos se aproximan a un plano horizontal.

- 1) *Calcular correlación de los vértices del polígono vértice por vértice.*
Se calcula para cada vértice del polígono la disparidad y se dibuja el vértice en la imagen secundaria con el desplazamiento calculado, luego se unen los vértices en el mismo orden indicado por el usuario en la imagen de referencia.
- 2) *Considerar media de los vértices.*
Se calcula para cada vértice su disparidad, luego se realiza un promedio de las disparidades halladas. Dicho promedio será la disparidad de cada vértice para poder dibujar el polígono en la imagen secundaria.
- 3) *Considerar la media de todos los puntos internos del polígono.*
Se calcula la disparidad para cada punto interior del polígono, luego se calcula el promedio, se toma el mismo como disparidad de cada vértice y se dibuja el polígono en la imagen secundaria.
- 4) *Considerar mediana de todos los puntos internos del polígono.*
Se calcula la disparidad para cada punto interior del polígono, luego se calcula la mediana, se toma la misma como disparidad de cada vértice y se dibuja el polígono en la imagen secundaria.

Para todos los métodos se debe tener en cuenta que los puntos sean válidos (por ejemplo que no sea en una oclusión). Para los casos 1) y 2), si la disparidad para el vértice seleccionado no es válida se busca una disparidad válida en los puntos contiguos. Se considera que la disparidad es válida cuando la confianza (salida del algoritmo de correlación, ver 4.2.3, parámetro *outmetric*) es mayor a cierto umbral definido.

En la figura 12, se muestra el resultado de seleccionar un edificio en la imagen de referencia y aplicar los métodos mencionados anteriormente para hallar el contorno de un edificio en la imagen secundaria. Como se aprecia, el peor de los métodos es "Por vértices", que es muy susceptible a las oclusiones, ya que si uno de los vértices seleccionados tiene mucho ruido, el contorno queda deformado. Por otro lado, el mejor de los métodos es

“Mediana Polígono”, porque la mediana ayuda a descartar los valores que contienen mayor error.

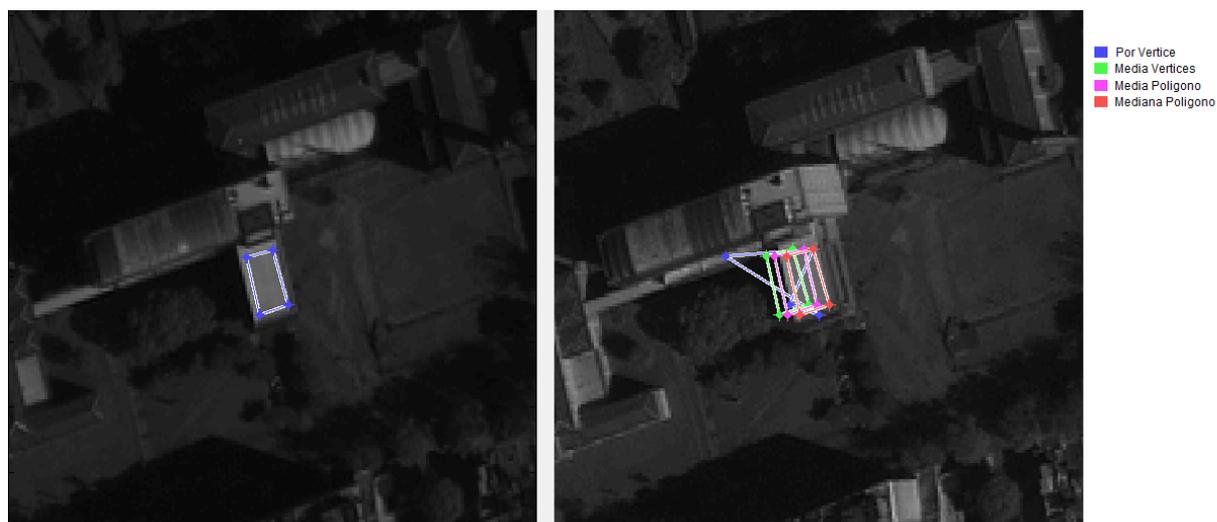


Figura 12 - Diferentes métodos para el modelado de los techos

En la sección 6.2 se presentan las validaciones realizadas a esta etapa del proceso. Se presenta un cuadro comparativo de los resultados obtenidos al ejecutar los distintos métodos para edificios conocidos.

4.4. Cálculo de alturas y georreferenciación

Como se vio en el marco conceptual hay dos formas de obtener las alturas: *baseline* y modelo RPC. Al adquirirse las imágenes de IKONOS, se acotan las soluciones ya que estas imágenes sólo vienen con la información para aplicar el modelo RPC.

Imágenes IKONOS y modelo RPC

Las imágenes estereoscópicas de los satélites IKONOS son adecuadas para generación de modelos 3D. Con el fin de poder ejecutar la extracción de características estéreo, con suficiente precisión, se utiliza el modelo RPC [5]. Este modelo cumple esa tarea con gran eficiencia y sin discernir pérdida de precisión. Es lo suficientemente genérico para comunicar fácilmente con la mayoría de las aplicaciones en el mercado, incluso aquellas de tipo COTS¹⁵.

De los diferentes tipos de productos proporcionados por IKONOS, el cliente adquirió la línea GeoStereo, estas imágenes tienen una resolución de 11 bits, capacidad multiespectral de 4 bandas (roja, verde, azul, infrarroja) y vienen rectificadas.

¹⁵ COTS proviene de *Commercial Off-The-Shelf* por sus siglas en inglés, refiere a productos comercialmente disponibles y que pueden ser comprados “as is”, es decir que no se pueden modificar.

Metadata de las imágenes IKONOS

Estas imágenes vienen con metadata que permite ejecutar el algoritmo RPC, citado anteriormente. Algunos datos interesantes para nuestro programa:

- Tipo de imágenes pancromáticas: (*Image Type: PAN*)
- Los datos vienen dados con el sistema de coordenadas WGS84 (*Datum: WGS84*)
- La resolución de las imágenes son de un píxel por metro (*Product Order Pixel Size: 1.0 meters*)
- Los archivos vienen en formato tiff sin ningún tipo de compresión (*File Format: TIFF*)
- Cada píxel tiene 11 bits de información (*Bits per Pixel per Band: 11 bits per pixel*)
- La posición del satélite (*Nominal Collection Azimuth: 203.7414 degrees y Nominal Collection Elevation: 82.36797 degrees*)
- Parámetros del RPC:
 - Desplazamientos para *line*, *sample*, altura, latitud y longitud: *LINE_OFF*, *SAMP_OFF*, *LAT_OFF*, *LONG_OFF* y *HEIGHT_OFF*. Se corresponden con O_L , O_S , O_φ , O_λ y O_h respectivamente.
 - Factores de escala para *line*, *sample*, altura, latitud y longitud: *LINE_SCALE*, *SAMP_SCALE*, *LAT_SCALE*, *LONG_SCALE* y *HEIGHT_SCALE*. Se corresponden con SF_L , SF_S , SF_φ , SF_λ y SF_h respectivamente.
 - Coeficientes del numerador y denominador para *line*, *sample*: *LINE_NUM_COEFF*, *LINE_DEN_COEFF*, *SAMP_NUM_COEFF* y *SAMP_DEN_COEFF*. Se corresponden con a_i , b_i , c_i y d_i , respectivamente.

4.5. Importación y exportación de outlines

Archivo XML

XML, por sus siglas en inglés de *eXtensible Markup Language*, es un lenguaje de marcas desarrollado por el World Wide Web Consortium¹⁶ (W3C), utilizado para almacenar datos en forma legible. XML se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Es una tecnología sencilla, que tiene a su alrededor otras que la complementan y la hacen más poderosa. Tiene un papel muy importante en la actualidad, ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

En la aplicación, se utiliza el formato xml para la exportación e importación de polígonos, esto logra compatibilidad con otros sistemas y le da flexibilidad al cliente quien solicitó este formato, por ser fácil implementar un script en Matlab que permita cargar los

¹⁶ (2013, Noviembre) World Wide Web Consortium [Online] <http://www.w3.org/>

polígonos. Por otra parte, los lenguajes siempre proveen una biblioteca para trabajar con este tipo de archivos lo cual es otro aliciente para su uso.

4.6. Exportación del modelo 3D

Formato STL

El formato STL (STereoLithography) es un formato que permite guardar objetos en 3D. Para exportar, se elige este formato por ser soportado por muchos programas del mercado, como por ejemplo, los programas CAD conocidos y utilizados por el cliente. Los archivos STL describen sólo la superficie geométrica de un objeto 3D sin representar ni el color ni la textura. Los dos formatos para representar estos archivos son ASCII y binarios. Los archivos binarios son más compactos, sin embargo los ASCII permiten visualizar lo que se guarda. Este último es el elegido por facilitar las pruebas, permitiendo validar el archivo a simple vista por ser un formato legible.

5. Diseño e implementación

Este capítulo está dedicado a los aspectos de diseño y de implementación del sistema. Se comienza describiendo la arquitectura general del sistema, para luego detallar las arquitecturas de cada capa. Se presenta el diagrama de clases de diseño. Luego se hace una breve reseña de las tecnologías utilizadas. Finalmente, se describen las dificultades encontradas en el transcurso del proyecto y se enumeran algunas de las limitaciones que presenta la solución desarrollada.

5.1. Arquitectura

Se utilizó una arquitectura de 3 capas: presentación, lógica y persistencia. Se elige este tipo de arquitectura porque permite separar el código de una manera simple que permite el escalamiento de un proyecto, además de poder implementar y probar las capas una a una en las etapas tempranas de un proyecto.

A continuación, en la figura 13 se presenta el diagrama de arquitectura:

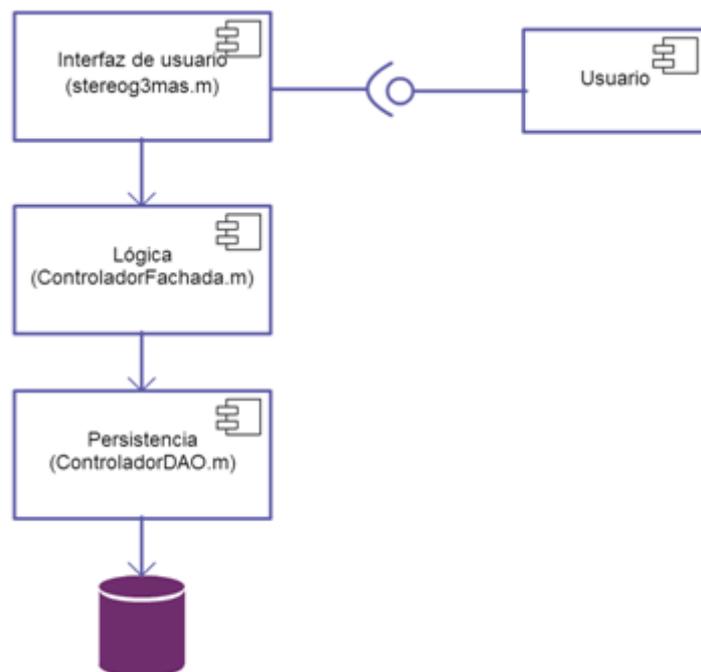


Figura 13 - Diagrama de arquitectura

Capa presentación

En esta capa se implementa la interfaz de usuario, llamada **stereog3mas**. Se utiliza la funcionalidad GUIDE de Matlab que permite diseñar una interfaz para aplicaciones de escritorio arrastrando componentes (*drag and drop*).

Capa lógica

En esta capa se encuentra la lógica de negocio, así como los objetos de negocio. Se implementa un controlador fachada que publica las operaciones necesarias. Dicho controlador llamado “ControladorFachada”, implementa el patrón de diseño Singleton, el cual provee una única instancia de acceso global. Además, tiene memoria, logrando recordar valores entre operaciones.

Capa persistencia

En esta capa se centraliza el acceso a la base de datos, mediante la clase “ControladorDAO”, se encarga de establecer y cerrar conexión con la base de datos así como alta, baja, modificación y consulta de datos. Se utiliza la biblioteca Mksqlite (se detalla en la sección 5.3.2) para persistir los datos.

Arquitectura en capa lógica

El tipo de aplicación que el cliente precisa tiene claramente un caso de uso principal, el cálculo de la altura de un edificio. Éste es imprescindible para que la aplicación sea útil y tiene una alta complejidad. Para definir la arquitectura hay que tener en cuenta que este caso de uso se basa en una sucesión de pasos, los cuales hay diferentes formas de calcularlos, por esto se optó por una **arquitectura de “tubos y filtros”** que permite intercambiar fácilmente el método utilizado en cada etapa. Pensando en el futuro sería sencillo que el sistema evolucione agregando nuevas transformaciones y/o nuevos métodos. En la figura 14, se pueden ver los distintos filtros que se le aplican a un par de imágenes satelitales estereoscópicas. Los filtros son los siguientes:

1. *Rectificación*: Este filtro es opcional ya que depende de las imágenes compradas. Por ejemplo las imágenes que se adquirieron de IKONOS ya vienen rectificadas (ver sección 2.1.3).
2. *Outline*: Este filtro se encarga de obtener los contornos de los edificios. Debido a que este procedimiento es muy complicado, se decidió simplificarlo y señalarlos de forma manual. El sistema también permite ingresar los *outlines* mediante un archivo xml (ver sección 4.5)
3. *Calcular Correlación*: Se encarga de hallar la disparidad de las imágenes (ver sección 2.2) y en conjunto con el contorno seleccionado en la imagen de referencia busca el edificio en la segunda imagen. Como alternativa, el sistema permite cargar un mapa de disparidad completo de la imagen, esto acelera el proceso de correlación.
4. *Calcular Altura/Latitud/Longitud*: Este procedimiento toma como parámetro los contornos de los edificios obtenidos por el filtro de “Outline” y el mapa de disparidad obtenido en “Calcular Correlación”. Luego se procesan los datos de entrada y se

hace el cálculo de altura, latitud y longitud para cada vértice del edificio seleccionado.

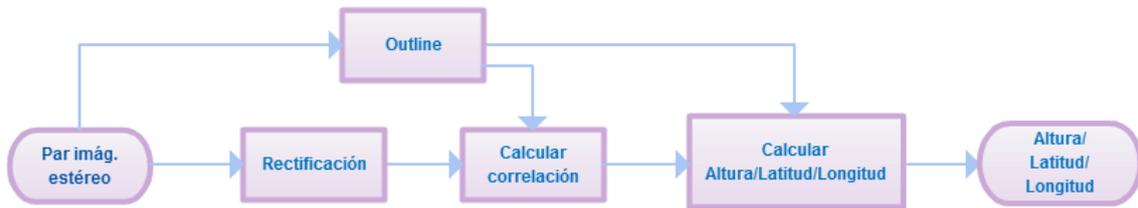


Figura 14 - Arquitectura de tubos y filtros

5.2. Diagrama de clases de diseño

La siguiente figura muestra el diagrama de clases de diseño (DCD), de las clases más importantes del sistema.

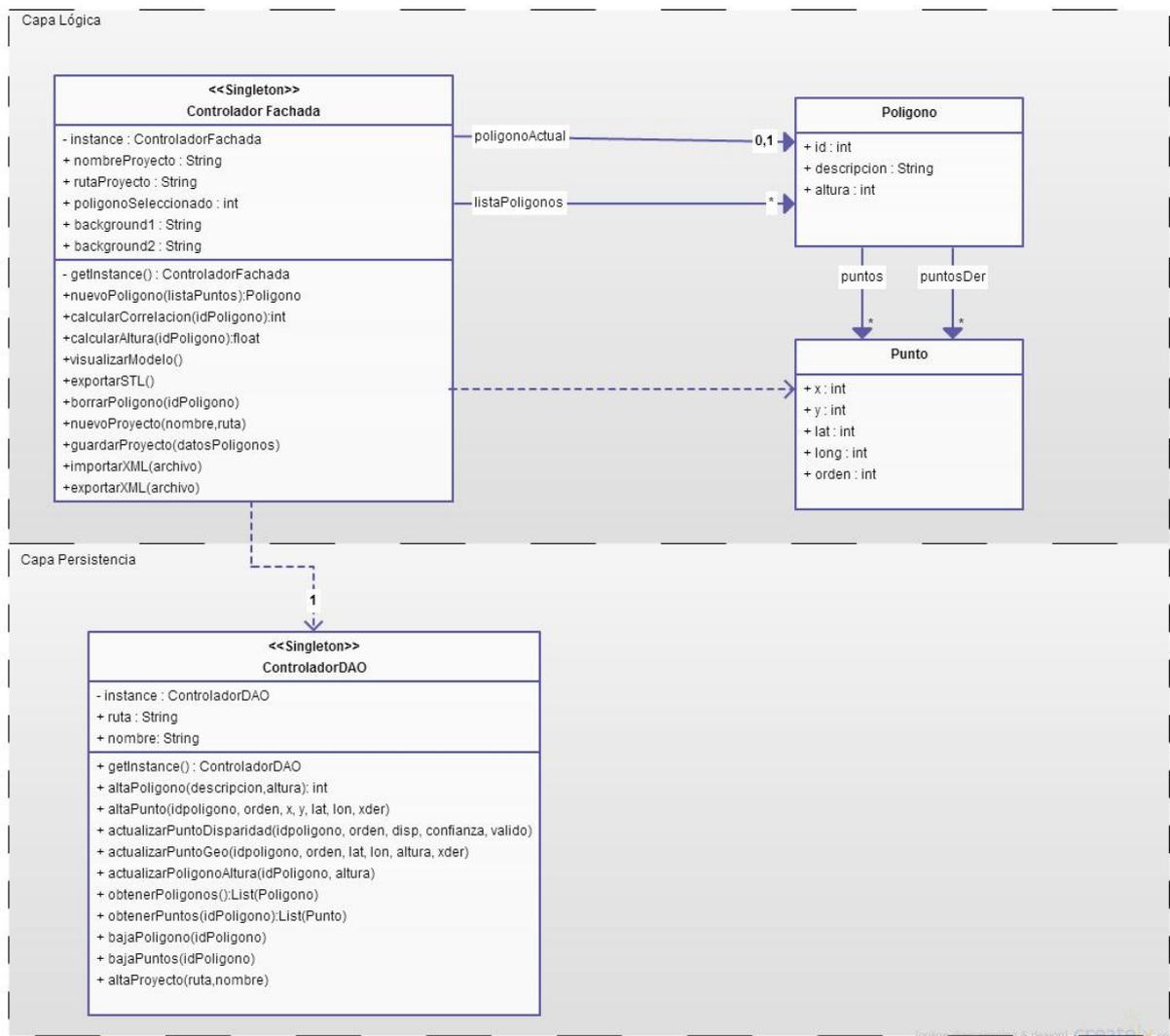


Figura 15 - Diagrama de clases de diseño

5.3. Tecnologías utilizadas

A continuación se detallan las principales tecnologías utilizadas en el sistema.

5.3.1. Matlab

Matlab es una herramienta de software matemático con un lenguaje de programación propio llamado lenguaje M. Ofrece un entorno de desarrollo integrado (IDE) que facilita todo el proceso de desarrollo. Es altamente recomendado para implementar algoritmos de alta complejidad en cálculos como lo es el procesamiento de imágenes, que demanda elevado poder de cómputo.

5.3.2. Mksqlite

Mksqlite es una biblioteca *open source* que permite comunicar MATLAB con SQLite. SQLite es un motor de base de datos también *open source*. Este tipo de base de datos se almacena en un archivo, lo cual la hace una base de datos muy liviana, ideal para este desarrollo. Mediante la instalación de una extensión para el navegador Firefox llamado "SQLite Manager" se puede administrar estas bases y consultar los datos.

5.3.3. Orfeo toolbox

Como se explica en la sección 4.2.3 se utiliza OTB para hallar la disparidad en las imágenes seleccionadas. Se elige esta biblioteca porque permite integrarla con Matlab y posee una gran variedad de operaciones útiles para el procesamiento de las imágenes, esto permite que el sistema a implementar sea escalable. Además posee documentación actualizada que detalla su uso y una comunidad que brinda soporte constantemente.

5.4. Implementación

Para explicar la implementación del software se hará por las etapas definidas en la sección 5.1 de “Arquitectura en capa lógica”.

Para la implementación de la **primera etapa** “Rectificación”, se va a utilizar el software que provee Orfeo (ver sección 4.2.3). Como primer paso se va a utilizar la operación StereoRectificationGridGenerator, la cual genera dos grillas de deformación para rectificar el par de imágenes estéreo.

Para ejecutar el comando se utilizan los siguientes parámetros:

- *Left input image*: La imagen de referencia de entrada para rectificar.
- *Right input image*: La imagen secundaria de entrada para rectificar.
- *Left output deformation grid*: Nombre del archivo de salida correspondiente a la grilla de deformación para rectificar la imagen de referencia.
- *Right output deformation grid*: Nombre del archivo de salida correspondiente a la grilla de deformación para rectificar la imagen secundaria.
- *DEM directory*: Este parámetro nos permite seleccionar el directorio en donde se encuentra el modelo digital de elevación (DEM).
- *Geoid File*: En caso de no pasar el DEM o no tener valor de la elevación para un determinado punto se le puede pasar el geode.
- *Default elevation*: Este parámetro permite ingresar la altura por defecto sobre el elipsoide cuando no se ingresa DEM ni Geoid para un determinado punto.
- *Step of the deformation grid (in nb. of pixels)*: Este parámetro sirve para estimar el desplazamiento en la grilla.

Los parámetros de salida son los siguientes:

- *Rectified image size X/Y*: Representa las dimensiones de la imagen de salida en geometría epipolar.
- *Mean baseline ratio*: Es el valor medio del *baseline*, en $(\text{píxeles.metro})^{-1}$.
- *Left deformation grid*: Grilla de deformación izquierda.
- *Right deformation grid*: Grilla de deformación derecha.

Luego de obtener las grillas de deformación, se utiliza la operación de Orfeo GridBasedImageResampling, esta permite combinar las grillas de deformación con las imágenes para rectificarlas.

Los parámetros para ejecutarla son los siguientes:

- *Input image*: La imagen para rectificar.
- *Output Image*: Nombre de imagen de salida.
- *Input resampling grid*: La grilla para usar.
- *Grid Type*: El tipo de grilla que se le pasa como parámetro, para rectificar se pasa “*Deformation grid*”.
- *Upper Left X*: Coordenada X del píxel superior izquierdo de la imagen de salida.
- *Upper Left Y*: Coordenada Y del píxel superior izquierdo de la imagen de salida.

- *Size X*: Tamaño del eje X de la imagen de salida, expresada en píxeles.
- *Size Y*: Tamaño del eje Y de la imagen de salida, expresada en píxeles.
- *Pixel Size X*: Tamaño de cada píxel en el eje de las X.
- *Pixel Size Y*: Tamaño de cada píxel en el eje de las Y.
- *Default value*: El valor por defecto para darle a los píxeles que caen afuera de la imagen de entrada.
- [Opcional] *Interpolation*: Este grupo de parámetros permiten definir como la imagen de entrada será interpolada durante el reconstrucción de la imagen.
 - *Nearest Neighbor interpolation*: Recomendado para imágenes de poca resolución, es muy rápido.
 - *Linear interpolation*: Recomendado para imágenes de calidad promedio, pero no es tan rápido como el anterior.
 - *Bicubic interpolation*: Recomendado para imágenes de buena calidad, es el más lento de los algoritmos.

Para la **segunda etapa** “Outline”, el software permite mediante la interfaz del programa, dibujar un polígono sobre las imágenes estéreo. Para esto se utiliza la biblioteca de Matlab *impoly*. Otra alternativa es cargar los *outlines* mediante un archivo xml siguiendo el formato definido en la sección 4.5, para lo que se utiliza la operación de Matlab *xmlread*, el cual permite leer un documento XML devolviendo el DOM (*Document Object Model*) con sus respectivos nodos.

La **tercera etapa** “Calcular Correlación”, también se realiza con el software que provee Orfeo, para esto se utiliza la operación “BlockMatching”, como su nombre lo indica esta aplicación permite ejecutar algoritmos de *block matching* para estimar la disparidad de un pixel entre dos imágenes.

Los parámetros que recibe son los siguientes:

- *Left input image*: La imagen de referencia
- *Right input image*: La imagen secundaria
- *The output disparity map*: Nombre del archivo de salida que contiene el mapa de disparidad, así como los valores métricos si están activados.
- *Output optimal metric values as well*: Si se activa, la imagen de salida tiene un segundo componente que es el valor métrico óptimo.
- *Block-matching metric*: Algoritmo de disparidad (SSD, NCC, Lp)
- *Radius of blocks*: Radio en píxeles
- *Minimum horizontal disparity*: Disparidad mínima horizontal para explorar
- *Maximum horizontal disparity*: Disparidad máxima horizontal para explorar
- *Minimum vertical disparity*: Disparidad mínima vertical para explorar
- *Maximum vertical disparity*: Disparidad máxima vertical para explorar
- *Median filtering*: Este filtro permite tener un mapa de disparidad más suave
 - *Radius of median filtering*: El radio para “*median filter*”
 - *Incoherence threshold*: Umbral de incoherencia entre la disparidad original y filtrada

En la **cuarta etapa** “Calcular Altura/Latitud/Longitud”, se utiliza el modelo RPC. Este modelo requiere de coeficientes llamados coeficientes RPC, los cuales vienen en la metadata de las imágenes. A grandes rasgos, lo que se hace es:

Dado un polígono pol seleccionado,

Para cada punto p

[latNorm, lonNorm, altNorm] = calcularUnPunto(p.x, p.y, p + correlacion(p))

[lat, lon, alt] = desnormalizar(latNorm, lonNorm, altNorm)

Fin

La función “calcularUnPunto” encapsula todos los cálculos, recibe como parámetros:

1. Coordenada x del punto en la imagen de referencia
2. Coordenada y del punto en la imagen de referencia
3. Coordenadas (x,y) del punto en la imagen secundaria, calculable como el punto en la imagen de referencia más la disparidad del punto, obtenida en la cuarta etapa “Calcular Correlación”.

Entrando más en detalle, la función “calcularUnPunto”, utiliza la función *lsqnonlin* de Matlab para resolver problemas de mínimos cuadrados no lineales. La misma resuelve problemas de la forma:

$$\min_x \|f(x)\|_2^2 = \min_x (f_1(x)^2 + f_2(x)^2 + \dots + f_n(x)^2)$$

La función *lsqnonlin* recibe como parámetros:

1. Una función $f(x) = [f_1(x); f_2(x); f_3(x); f_4(x)]$ que corresponden a las 4 ecuaciones del modelo RPC mencionadas en la sección 2.3.2. Dichas ecuaciones tienen 3 incógnitas (U, V, W) que corresponden a latitud, longitud y altura normalizadas y utilizan los coeficientes RPC.
2. Un punto inicial $x_0 = (0.0, 0.0, 0.0)$
3. Un conjunto de opciones (este parámetro es opcional), en este caso se elige el algoritmo Levenberg-Marquardt. También se podría activar la opción Jacobiana, que permite pasarle la matriz Jacobiana (derivadas parciales de f_i con respecto a x siendo $x = [U; V; W]$) para optimizar la ejecución, acelerando la convergencia a la solución.

Como salida de la función “calcularUnPunto” se obtiene la latitud, longitud y altura, todos estos valores normalizados. Por lo tanto, posteriormente, es necesario invocar a la función “desnormalizar” que recibe como parámetros de entrada estos tres valores y retorna los mismos desnormalizados, siendo el resultado esperado por esta etapa.

Importación y exportación de outlines

En la aplicación, se utilizan archivos en formato xml para la exportación e importación de polígonos. La estructura es la siguiente:

```
<poligono>
  <descripcion>techo fing</descripcion>
  <altura>38.1</altura>
  <puntos>
    <punto>
      <orden>1</orden>
      <x>139</x>
      <y>111</y>
      <xder>146</xder>
    </punto>
    <punto>
      <orden>2</orden>
      <x>154</x>
      <y>105</y>
      <xder>161</xder>
    </punto>
    <punto>
      <orden>3</orden>
      <x>155</x>
      <y>117</y>
      <xder>162</xder>
    </punto>
  </puntos>
</poligono>
```

Es decir, que existe un nodo “poligono” por cada polígono. De cada polígono se guarda su descripción, altura (en caso de que ya esté calculada) y lista de puntos. Cada punto tiene: orden (de unión de los puntos), coordenadas x, y, x de la imagen secundaria (en caso de que ya esté calculada la disparidad). Al exportar un archivo, el mismo sirve posteriormente para ser importado por la aplicación.

Exportación del modelo 3D

Para la exportación del modelo se utiliza un archivo en formato STL, que describe la triangulación de una superficie mediante el vector unitario de la normal y los vértices de la triangulación, usando coordenadas cartesianas de tres dimensiones. La estructura es la siguiente:

```
solid Edificio1
  facet normal nx ny nz
    outer loop
      vertex v1x v1y v1z
      vertex v2x v2y v2z
      vertex v3x v3y v3z
    endloop
  endfacet
endsolid Edificio1
```

Es decir que se construye un solo objeto llamado “Edificio1” con un solo triángulo descrito por la palabra “facet”. Se aprecia que su normal es (n_x, n_y, n_z) y los vértices son (v_{1x}, v_{1y}, v_{1z}) , (v_{2x}, v_{2y}, v_{2z}) y (v_{3x}, v_{3y}, v_{3z}) .

5.5. Dificultades encontradas

Una de las primeras dificultades encontradas fue la búsqueda del *baseline* de las imágenes, este dato no venía dado y se intentó calcular mediante trigonometría. Al no tener éxito, se buscó que el proveedor de las imágenes proporcionara dicho dato o indicara una alternativa para hallar el mismo. No pudiendo facilitar dicha información, la alternativa sugerida por su parte fue obtener las alturas mediante el modelo RPC.

Otro inconveniente surgió a partir de las imágenes adquiridas en una primera instancia. Éstas no fueron útiles por estar en formato color (RGB) por ser éste incompatible con los programas utilizados. Fue necesario comprar imágenes pancromáticas a un costo adicional (5 dólares por km²).

Otra de las dificultades encontradas fue el lenguaje requerido para la realización del programa, debido a que, si bien éste fue usado en la carrera, nunca se utilizó para la creación de una aplicación de escritorio de mediano porte. Se encontraron dificultades a la hora del diseño de la interfaz y manejo de imágenes. Para la primera, se utilizó *guide editor* de Matlab, pero éste contiene un conjunto muy acotado de objetos, limitando las opciones a elegir. Para la segunda, Matlab no proporciona ninguna librería para el manejo de imágenes grandes, haciendo que la carga y visualización de las mismas sea muy lenta.

Dada la falta de experiencia en el área de procesamiento de imágenes, el aprendizaje lleva su tiempo. Fue necesaria la incorporación de nuevos conceptos, tomando como referencia publicaciones existentes sobre el tema y pruebas experimentales con los aplicativos mencionados en la sección 4.2. Con respecto a los programas y sus manuales de uso, existe poca información o no es fácil acceder a ella, por lo que se vuelve necesario probarlos empíricamente.

El inconveniente de mayor magnitud se presenta debido a la falta de medidas exactas de las edificaciones, lo que dificulta poder validar la calidad de los datos obtenidos por el software. Este problema se ve reducido por el uso de GPS, sin embargo no es viable la medición de todos los edificios de interés.

5.6. Limitaciones

Algunas de las limitaciones del producto implementado son:

1. Al momento de seleccionar el *outline* de un edificio se podrían contemplar más facilidades como arcos, círculos o líneas ortogonales, de manera de permitir indicar el *outline* lo más exacto posible.
2. Este tipo de aplicaciones que procesan imágenes satelitales requieren manejo de imágenes grandes. Como no existe en Matlab una biblioteca que facilite la carga, se debería implementar un desarrollo propio que insumiría mucho tiempo. Una opción sería tener precalculados distintos zoom y/o subdivisiones (tiles) de las imágenes, estos sirven para no levantar toda la imagen sino una parte de ella, lo que aceleraría la visualización y ahorraría memoria.
3. Sería conveniente que la herramienta sea capaz de identificar distintas tipologías de techos a partir de las imágenes (techo horizontal, techo a dos aguas, entre otros). Para esto, se debe aproximar en la etapa de correlación mediante una transformación afin, lo que sería sencillo de adaptar. No se realizó esta mejora por un tema de tiempo, debido a que las validaciones realizadas quedarían obsoletas y se debería comenzar nuevamente el proceso de validación.

6. Validaciones

En este capítulo se presentan las validaciones realizadas para determinar los métodos con mejor precisión en cada etapa, así como la validación al sistema implementado.

6.1. Validaciones de disparidad

Para comparar los distintos métodos de disparidad se utilizó la herramienta OTB. Para cada uno de ellos se calculó el mapa de disparidad, que es ingresado a **Stereog3mas** para ser utilizado en los cálculos. Como altura verdadera se considera la del GPS, CityModeler o **Stereog3mas**, priorizando en ese orden. A continuación se presentan las alturas calculadas en metros, entre paréntesis se muestra el error y se resaltan los mejores resultados:

Edificio	Altura verdadera	Altura calculada NCC	Altura calculada SSD	Altura calculada Lp
IIE 1	47.50	47.50 (0.00)	52.90 (5.40)	51.10 (3.60)
IIE 2	38.70	38.70 (0.00)	42.20 (3.50)	42.20 (3.50)
IIE 3	47.50	47.50 (0.00)	49.30 (1.80)	49.30 (1.80)
pasillo IIE	67.90	68.70 (0.08)	68.70 (0.80)	68.70 (0.80)
techo entrada fing	48.79	49.30 (0.51)	49.30 (0.51)	49.30 (0.51)
techo inco	71.70	67.00 (4.70)	67.00 (4.70)	67.00 (4.70)
pasillo IMFIA	50.60	49.30 (1.30)	51.00 (0.40)	51.00 (0.40)
bandejas IMFIA	40.00	42.20 (2.20)	42.20 (2.20)	44.00 (4.00)
aulario1	35.10	33.30 (1.80)	31.60 (3.50)	33.30 (1.80)
aulario2	35.50	35.10 (0.40)	35.10 (0.40)	35.10 (0.40)
galpón fing	32.90	33.40 (0.50)	31.60 (1.30)	31.60 (1.30)
torre patria	103.78	45.80 (57.88)	0.00 (103.78)	5.10 (98.68)
edificio al lado TP	59.90	59.90 (0.00)	59.90 (0.00)	63.40 (3.50)
edificio EF	61.96	61.70 (0.26)	63.40 (1.44)	63.40 (1.44)
EB	63.34	59.90 (3.44)	91.80 (28.46)	59.90 (3.44)
IIE 4	42.20	47.50 (5.30)	38.70 (3.50)	35.20 (7.00)

Tabla 1 – Comparación de algoritmos de disparidad, entre paréntesis se muestra el error y se resaltan los mejores resultados

Como se aprecia en la tabla, la altura calculada es más precisa con el mapa de disparidad hallado con el algoritmo NCC, siendo uno de los mejores resultados en catorce de los dieciséis (14/16) casos calculados.

6.2. Validaciones modelado de techos

Para validar los diferentes métodos para modelar los techos, se usa el sistema **Stereog3mas** indicando el método a utilizar en el parámetro de configuración *metodoDisparidad* (ver anexo XI). A continuación se muestra una tabla en donde se detallan los resultados obtenidos de las pruebas realizadas:

Edificio	Altura verdadera (m)	Método	Altura Calculada (m)	Error (m)
Techo entrada FING	48.79	Mediana polígono	49.30	0.51
		Media polígono	47.50	1.29
		Media vertices	56.40	7.61
		Vértices	56.40	7.61
Techo Inco	71.70	Mediana polígono	67.00	4.70
		Media polígono	67.00	4.70
		Media vertices	63.40	8.30
		Vértices	63.00	8.70
Edificio Frente	61.96	Mediana polígono	61.70	0.26
		Media polígono	54.60	7.36
		Media vertices	44.00	17.96
		Vértices	44.00	17.96

Tabla 2 – Comparación de métodos para el cálculo de los techos

Para comparar la precisión de cada método, se toman como valores verdaderos los obtenidos con GPS y CityModeler (datos que se presentan en la sección 6.3 con mayor detalle). En todos los casos, la mediana es la que mejor se aproxima al valor verdadero. En la altura del techo de la entrada de la FING es de 49.3 m, con un error de 0.51 m, para el techo del Inco da 67 m, con un error de 4.7 m, en el último caso da 61.7 m con un error de 0.26 m. Este resultado es el esperado ya que la mediana es una medida robusta, debido a que descarta los valores erróneos.

6.3. Validaciones realizadas al programa

En esta sección se presentan las validaciones hechas a la aplicación desarrollada. Al no existir datos reales de las alturas de los edificios, lo que se hizo fue tomar medidas con un programa comercial que adquirió el cliente, llamado CityModeler. También se realizaron medidas con GPS adquirido por la facultad. De esta manera, se pueden comparar los resultados obtenidos.

6.3.1. CityModeler

Metria CityModeler es una aplicación comercial disponible en el mercado, implementada por la empresa Revisor AB. Este software permite crear modelos 3D de la ciudad a partir de dos imágenes satelitales de alta resolución y opcionalmente un modelo digital de elevación. Los modelos generados pueden ser exportados a Google Earth

(formato kml) y a formato *shapefile*¹⁷. El cliente adquirió una licencia de 2 años. Fue una gran referencia para temas de usabilidad y funcionalidades requeridas. Además, se aprovechó tomar nota de los resultados obtenidos (latitud, longitud y altura) para puntos conocidos dentro de la ciudad de Montevideo para luego cotejar con los resultados del programa implementado. En la figura 16 se muestra la interfaz del programa.

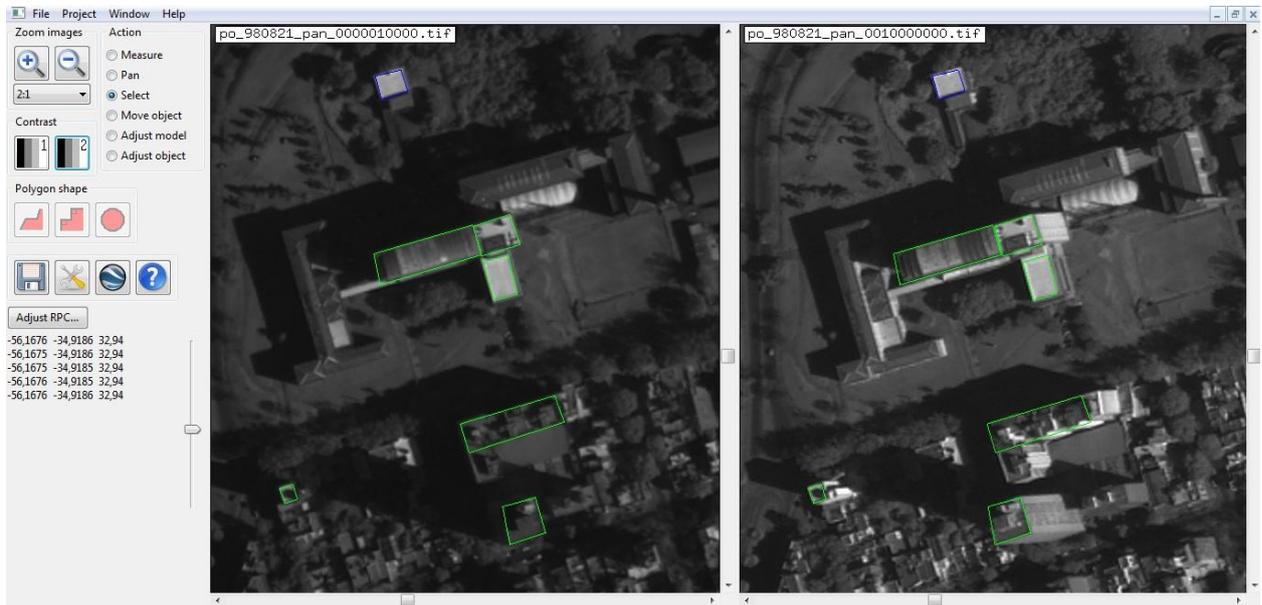


Figura 16 - Proyecto CityModeler zona Facultad de Ingeniería

¹⁷ (2013, Noviembre) Formato shapefile [Online]
<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

6.3.2. GPS

A pedido del cliente, profesionales de Agrimensura tomaron medidas sobre el techo de Facultad de Ingeniería, más precisamente en el techo del Instituto de Computación (InCo), basados en la tecnología Ground Truth. Ground Truth es un término utilizado para la georreferenciación de objetos, se refiere a la información recogida en el lugar permitiendo relacionar los datos de la imagen con características reales y materiales sobre el terreno. La recolección de datos en el lugar permite calibrar los datos obtenidos por el GPS y ayuda a la interpretación y análisis de lo que está siendo detectado.

Este relevamiento fue realizado con GPS, utilizando el método "tiempo real". El GPS consiste de dos bases: una fija y otra móvil. A la base fija se le configura una coordenada inicial. La base móvil se encarga de obtener una coordenada instantánea del sistema GPS. Estas bases se comunican entre sí, de forma que en la base móvil se realicen correcciones, que básicamente consisten en realizar la diferencia entre la coordenada inicial y la coordenada instantánea. Este procedimiento permite minimizar errores que se producen al tomar datos GPS, obteniendo muy buena precisión relativa entre estos puntos. Para obtener buena precisión absoluta (en el sistema global), basta con que la coordenada inicial de la base haya sido obtenida con buena precisión absoluta. Esto se logra, con varias horas de toma de datos GPS y la realización de un post-proceso con alguna de las bases fijas del SGM, tarea realizada previamente por el Instituto de Agrimensura.

A continuación, se muestra una tabla con los puntos y valores de alturas ortométricas que se obtuvieron:

Id de punto	Latitud	Longitud	Alt. ortométrica (m)
1	34 55 5.323098 S	56 10 0.370978 W	54.877
2	34 55 5.304679 S	56 10 0.023881 W	54.978
3	34 55 5.305436 S	56 10 0.013856 W	54.784
4	34 55 5.296482 S	56 9 59.855888 W	54.718
5	34 55 5.961850 S	56 10 0.317339 W	54.715
6	34 55 5.974136 S	56 9 59.721948 W	54.696
7	34 55 5.877117 S	56 9 59.966517 W	55.247
8	34 55 5.864733 S	56 9 59.754651 W	55.266
9	34 55 5.533092 S	56 9 59.782174 W	55.288
10	34 55 5.540915 S	56 9 59.985079 W	55.244

11	34 55 5.502807 S	56 9 59.769062 W	54.758
12	34 55 5.514047 S	56 9 59.995017 W	54.739
RTCM-Ref 0000	34 55 5.891389 S	56 9 59.984659 W	57.365

Tabla 3 – Alturas ortométricas

En la figura 17, se puede ver la ubicación de los puntos tomados con el GPS sobre una imagen de Google Earth. Estos puntos se encuentran desfasados ya que la imagen está en perspectiva. De todas formas dejan una idea de su ubicación en el techo. Las coordenadas se encuentran en grados, minutos, segundos.

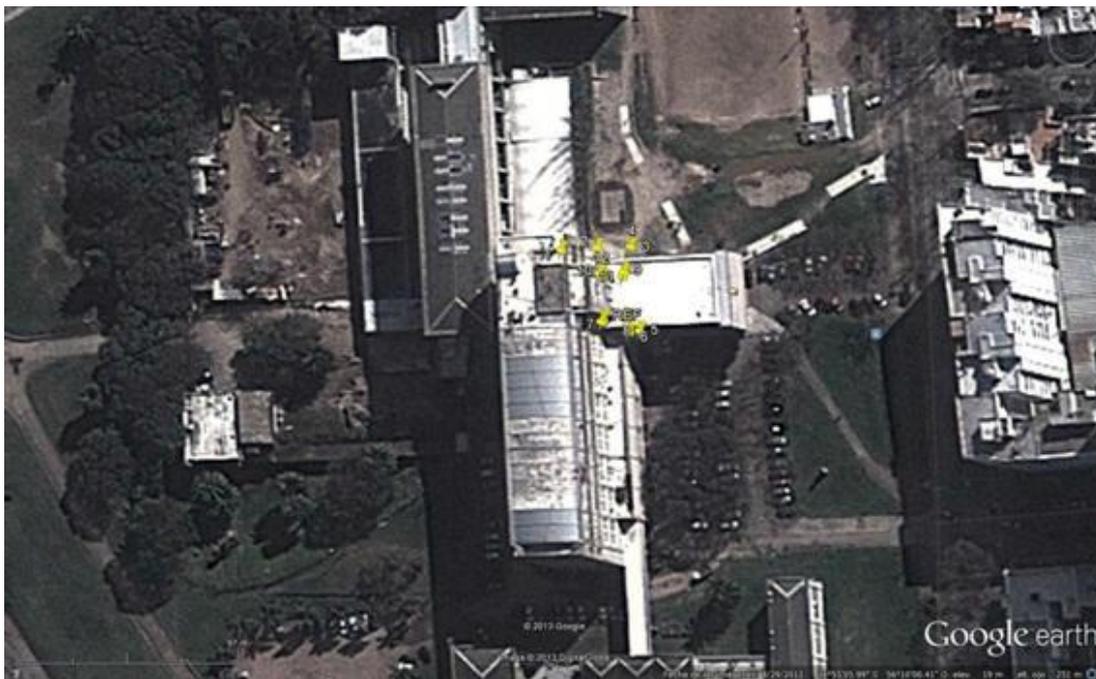


Figura 17 - Puntos medidos visualizados en Google Earth

En la figura 18, se presentan los edificios tomados como referencia para las validaciones: Torre Patria (TP), edificio frente a la facultad (EF) y edificio blanco frente al club de Golf (EB).



Figura 18 - Edificios tomados para validar

Para realizar este relevamiento, se midieron ángulos y distancias. Con esto se obtuvieron coordenadas relativas, que después se pasaron a UTM (*Universal Transverse Mercator*). El procedimiento realizado fue el siguiente:

- a. Medir el ángulo horizontal entre la estación de referencia del Servicio Geográfico Militar que está en el Cerro de Montevideo (del cual se conocen sus coordenadas con alta precisión) y el punto de interés (TP, EF, EB), con centro en el punto de Facultad de Ingeniería (del cual se conocen sus coordenadas).
- b. La distancia medida es entre la Facultad de Ingeniería y el punto de interés.
- c. Entre estos puntos también es necesario medir el ángulo vertical.
- d. Teniendo esos tres datos (ángulo horizontal, distancia y ángulo vertical) se puede obtener la coordenada de los edificios de interés.

Se tomaron dos medidas del edificio TP, dos de EF y una de EB. Las coordenadas se encuentran en sistema UTM. A continuación, se presentan los resultados obtenidos:

	E (m)	N (m)	H elipsoidal (m)	H ortométrica (m)
FI	576119,87	6135699,86	71,70	57,37
TP1	576262,32	6135669,56	103,82	89,49
TP2	576260,65	6135689,84	103,78	89,45
EF1	576214,48	6135665,61	61,97	47,64
EF2	576211,36	6135726,02	61,96	47,63
EB1	576230,18	6135575,62	63,34	49,01

Tabla 4 – Medidas GPS

6.3.3. Stereog3mas

Por último, se tomaron las medidas en la misma zona de interés con la aplicación desarrollada. A continuación, en la figura 19, se muestra una captura de pantalla donde se pueden apreciar los *outlines* de edificios seleccionados, así como los resultados obtenidos.

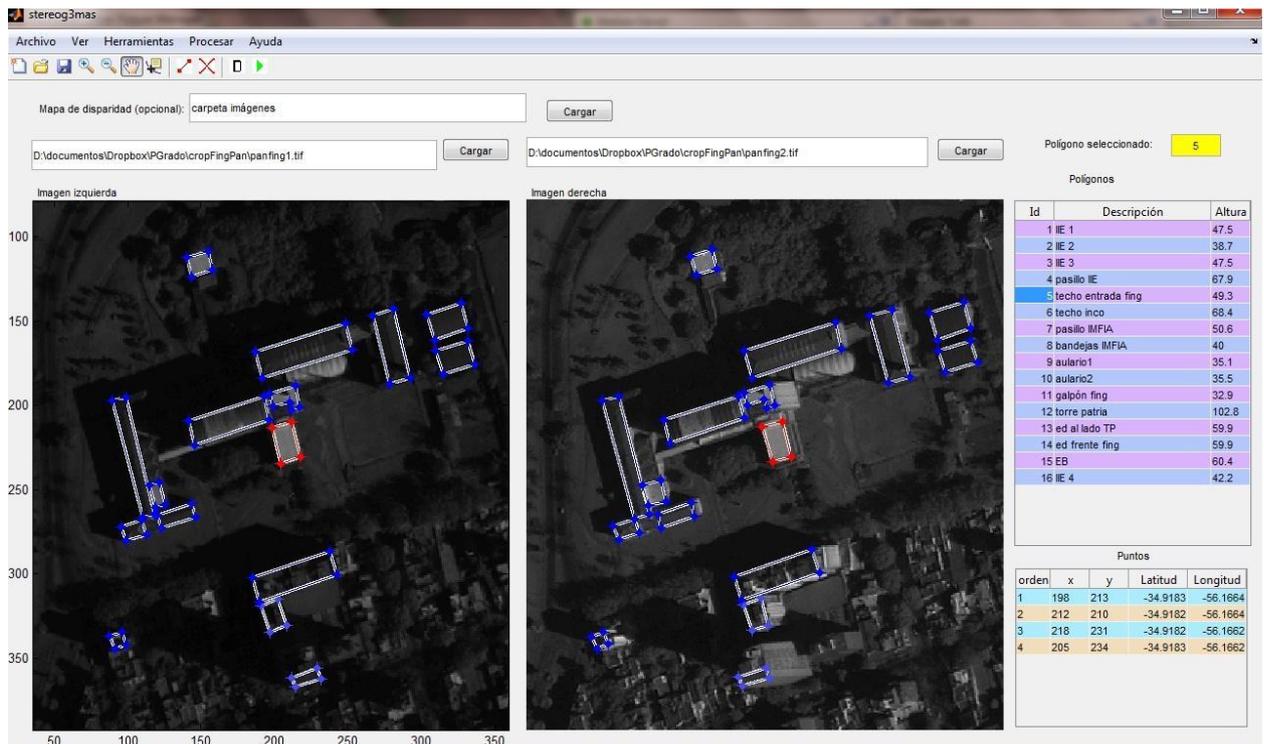


Figura 19 - Stereog3mas

Una vez finalizado el trabajo en la zona de interés, el programa permite visualizar un modelo 3D de la zona con las alturas elipsoidales calculadas. Dicho modelo se puede apreciar en la figura 20.

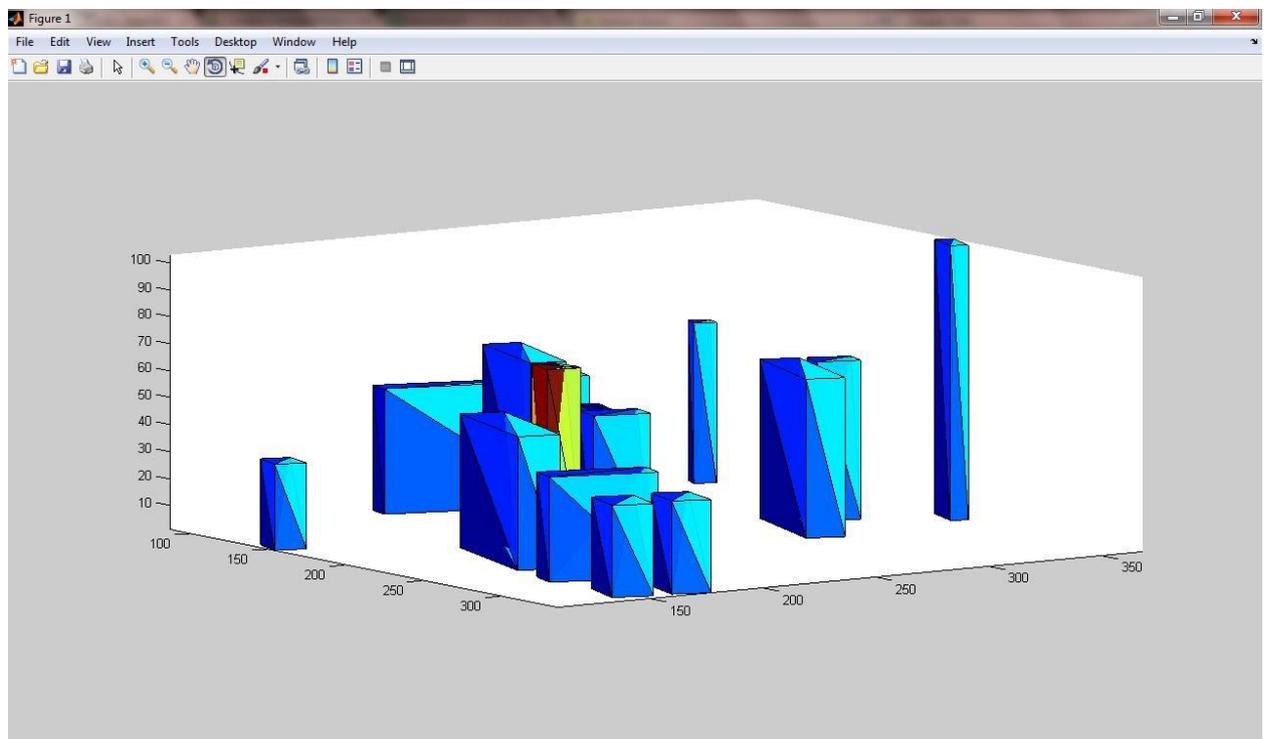


Figura 20 - Modelo 3D de la zona de Facultad de Ingeniería

6.3.4. Cuadro comparativo de los resultados

A continuación, se presenta un cuadro comparativo entre las medidas tomadas con las distintas herramientas. Los puntos de los polígonos (edificios) seleccionados son los mismos para **Stereog3mas** y CityModeler.

CM = CityModeler

sg3 = Stereog3mas

h = altura, las mismas se encuentran especificadas en metros

dif GPS = diferencia en metros de altura entre sg3 y GPS

dif CM = diferencia en metros de altura entre sg3 CM

S/V = sin valor

Edificio	h CM	h sg3	h GPS	dif GPS	dif CM
galpón fing	32.94	32.90	S/V	S/V	0.04
techo inco	67.89	68.40	71.70	3.30	0.51
techo entrada fing	48.79	49.30	S/V	S/V	0.51
torre patria	102.62	102.80	103.78	0.98	0.18
edificio EB	59.80	60.40	63.34	2.94	0.60
edificio EF	59.19	59.90	61.96	2.06	0.71

Tabla 5 – Comparación de alturas

En la tabla 5, se puede apreciar que los resultados obtenidos con sg3 no difieren de los de CM en más de 0.71 m., como es el caso de “edificio EF”, derivando que los errores de los métodos basados en imágenes son similares. Con respecto al GPS, las diferencias son un poco mayor, varía de 0.98 - 3.3 m. Vale destacar que la altura de sg3 está más cercana a la altura de GPS comparado con CM ($CM < sg3 < GPS$).

Edificio	lat CM	lon CM	lat sg3	lon sg3	lat GPS	lon GPS
galpón fing	-34.9186	-56.1676	-34.9186	-56.1676	S/V	S/V
techo inco	-34.9183	-56.1667	-34.9183	-56.1667	-34.9183	-56.1667
techo entrada fing	-34.9183	-56.1664	-34.9183	-56.1664	S/V	S/V
torre patria	-34.9186	-56.1649	-34.9186	-56.1649	-34.9186	-56.1651
edificio EB	-34.9196	-56.1654	-34.9196	-56.1654	-34.9194	-56.1654
edificio EF	S/V	S/V	-34.9181	-56.1656	-34.9186	-56.1656

Tabla 6 – Comparación de georreferenciación

En la tabla 6, la diferencia en la georreferenciación de las tres formas calculadas no es significativa, deduciendo que la generación de datos de sg3 es satisfactoria.

7. Conclusiones

La propuesta de este proyecto de grado fue evaluar la factibilidad de crear una aplicación que permita obtener las alturas de los edificios a partir de imágenes satelitales, se puede afirmar que se cumplió el objetivo porque se implementó un software que recibe las imágenes y obtiene un modelo 3D de los edificios señalados. Para lograr esto, se implementaron los requerimientos funcionales que demandó el cliente, así como los requerimientos no funcionales. Dentro de los no funcionales, se destacan el tiempo de procesamiento de la solución, el error de las alturas calculadas y la extensibilidad del sistema. En cuanto al plazo del tiempo de procesamiento, inicialmente era en días, pero luego de realizado el programa se obtienen alturas en cuestión de minutos, por ende, en este aspecto el sistema cumple con lo requerido. Viendo la calidad de los datos de salida también se puede afirmar que son confiables. Era requerido un error en las alturas menor a 10% (a modo de ejemplo, para un edificio que mide 100 m, el error debía ser menor a 10 m). Según las validaciones, el margen de error de **Stereog3mas** no supera el 4.7%. En cuanto al diseño, la arquitectura de tubos y filtros, otorga al programa la cualidad de ser extensible. Teniendo en cuenta el constante desarrollo de nuevos métodos para el cálculo de alturas, especialmente para calcular la disparidad, la arquitectura elegida permite agregarlos fácilmente. En cuanto a la usabilidad del aplicativo, es oportuno destacar que un usuario sin conocimientos técnicos en el área puede utilizarlo sin dificultades, porque luego de haber configurado las imágenes en el sistema el uso es intuitivo.

El abordaje del proyecto resultó difícil por ser un tópico que no se trató en el transcurso de la carrera, lo que hizo necesario incorporar los conceptos inherentes a este tema, haciendo que el tiempo de aprendizaje sea extenso. Para mitigar este problema, se resuelve dividir el problema en etapas que, a grandes rasgos, son dos: cálculo de disparidad, y obtención de la altura y georreferenciación. Para el primero, se utilizó la biblioteca Orfeo Toolbox la cual provee la implementación de los algoritmos de disparidad. Para la segunda, se utilizó el modelo RPC para el cual las imágenes de IKONOS proveen la información necesaria.

Cabe destacar que el sistema agrega valor al permitir calcular el modelo 3D de cualquier ciudad si se cuenta con las imágenes correspondientes.

Los aspectos mencionados anteriormente permitieron lograr la satisfacción del cliente, entregando un software a medida, con la misma calidad de sus semejantes en el mercado.

8. Trabajo a futuro

En este capítulo se enumeran las funcionalidades, mejoras o ideas que se podrían considerar para continuar este trabajo. Por razones de tiempo y complejidad no pudieron ser consideradas, pero sería pertinente dejarlas plasmadas como punto de partida para trabajos futuros.

Se evaluó con el cliente, agregar a la herramienta la capacidad de identificar, de forma semi-automática, los *outlines* de los edificios. Se investigaron algunos algoritmos de detección de aristas y se decide comparar los métodos: Sobel, Prewitt, Roberts, Laplacian of Gaussian, Zero-Cross y Canny.

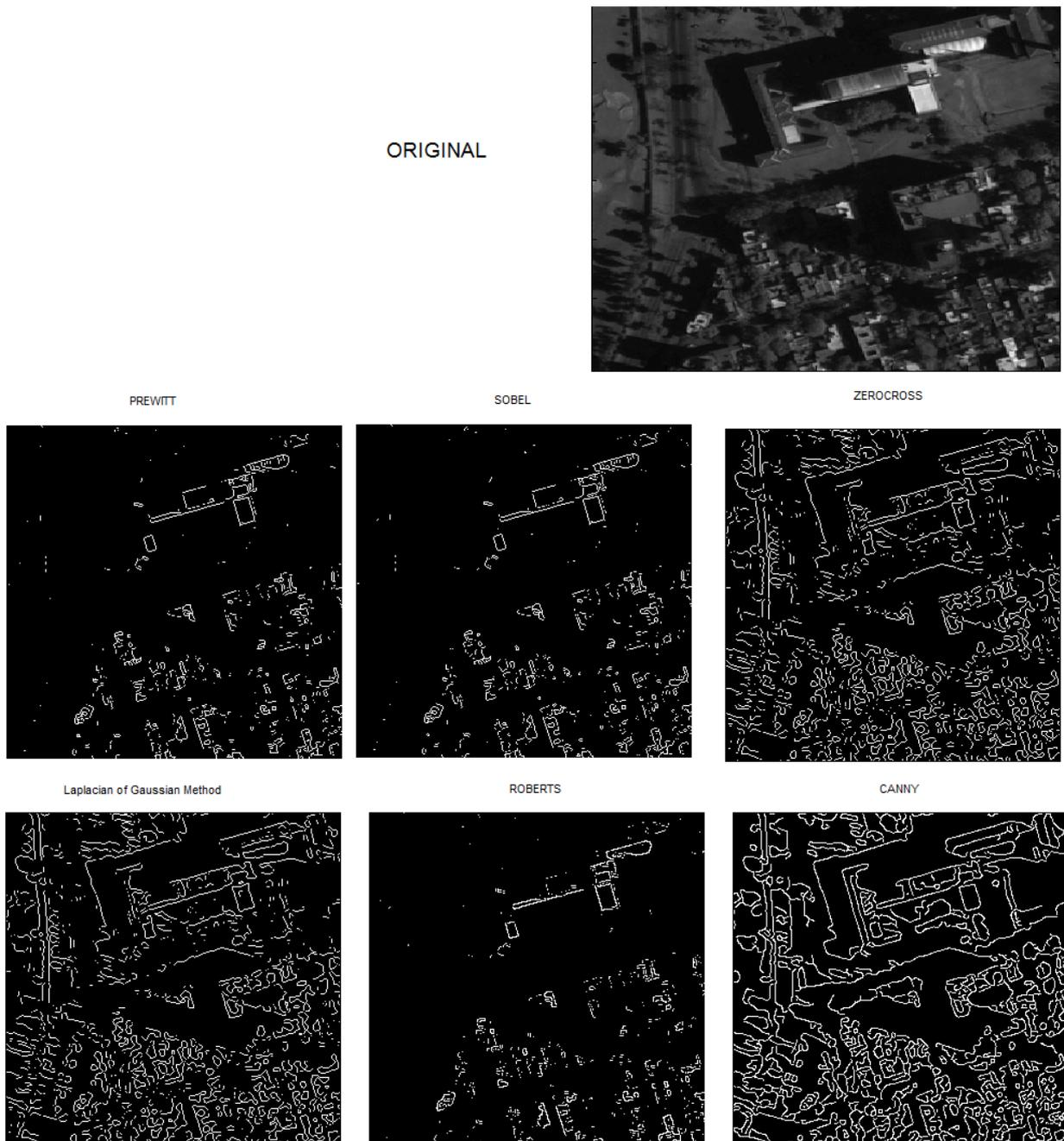


Figura 21 - Métodos de obtención de aristas

La figura 21 muestra el resultado de aplicarlos en la zona de interés (alrededores de la Facultad de Ingeniería). Se puede apreciar que el mejor método es el Canny, el cual detecta mejor los bordes de los edificios. Otro método a considerar es Line Segment Detector (LSD)¹⁸. Debido a la naturaleza de las imágenes, éstas contienen sombras que hacen que la detección no sea perfecta. Por este motivo, se decidió no abarcar la detección automática de los contornos porque requiere un post-procesamiento complicado y no es de relevancia en este proyecto.

Sería conveniente implementar la exportación en nuevos formatos. Como por ejemplo, permitir compatibilidad con Google Maps y Google Earth, ya que estos son fáciles de usar para un usuario con poca experiencia en el tema. Otro formato podría ser *shapefile* para Sistemas de Información Geográfica, debido a que estos permiten más control para usuarios especializados.

Otro trabajo a futuro podría ser, dado que los métodos en cada etapa de la implementación son intercambiables y extensibles, implementar nuevos métodos para el cálculo de disparidad, así como nuevos algoritmos para el cálculo de latitud, longitud y altura. En particular, se podría continuar con el método del *baseline* y agregar la opción al programa como una nueva alternativa.

Como última sugerencia, se podría migrar la aplicación a un lenguaje más adecuado para el manejo de imágenes grandes, además de permitir dibujar y manipular polígonos sobre la interfaz gráfica de usuario. Sería conveniente que el lenguaje elegido provea bibliotecas para estos fines.

Sería bueno también, comparar cuánto mejora la precisión en los cálculos si se obtienen imágenes de mejor resolución, y de esta manera, analizar si es oportuna una inversión mayor.

¹⁸ (2013, Noviembre) Image Processing On Line, LSD [Online] <http://www.ipol.im/pub/art/2012/gjmr-lsd/>

9. Bibliografía

- [1] Jaehong Oh. Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-based Georeferencing of Aerial Images, presented for the Degree Doctor of Philosophy, in the Graduate School of The Ohio State University, 2011
- [2] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in Computational Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(8):993–1008, 2003.
- [3] D. Scharstein, R. Szeliski, A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, International Journal of Computer Vision, Volume 47 Issue 1-3, April-June 2002
- [4] (2013, Noviembre) Geolocation Accuracy of WorldView Products [Online] http://www.digitalglobe.com/sites/default/files/WorldView_Geolocation_Accuracy.pdf
- [5] Jacek Grodecki, IKONOS stereo feature extraction - RPC approach, 2001
- [6] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision 2ed
- [7] Olivier Faugeras and Quang-Tuan Luong. The geometry of multiple images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications. The MIT Press, Cambridge, MA, USA, 2001
- [8] Javier Preciozzi. Dense Urban Elevation Models from Stereo Images by an Affine Region Merging Approach, Master Thesis, Universidad de la República, Montevideo, Uruguay, September 2006
- [9] (2013, Noviembre) IKONOS Imagery Products Guide [Online] http://glcf.umd.edu/library/guide/IKONOS_Product_Guide_jan06.pdf
- [10] A. Almansa, G. Blanchet, A. Chambolle, T. Coll, J. Delvit, Y. Gousseau, R. Keriven, Y. Meyer, L. Moisan, J. Morel, B. Roug, L. Rudin, Reliability and accuracy in stereovision Application to aerial and satellite high resolution images, presented for the Degree Doctor of Philosophy, Escuela Normal Superior de Cachan, France, December 2009
- [11] Sandra Eckert, 3D-Building Height Extraction from Stereo IKONOS Data, 2008
- [12] (2013, Noviembre) SRGIS, Guía básica sobre Imágenes Satelitales y sus productos [Online] http://www.srgis.cl/pdf/guia_basica_imagenes_satelitales.pdf
- [13] (2013, Noviembre) Nick Hubing, Top 10 things to consider to buy satellite imagery [Online] <http://landinfo.com/EIJ.Optical-Article.pdf>
- [14] (2013, Noviembre) The Orfeo ToolBox Cookbook, a guide for non-developers, 2012 [Online] <http://www.orfeo-toolbox.org/packages/OTBCookBook.pdf>
- [15] (2013, Noviembre) The Orfeo ToolBox Software Guide, 2013 [Online] <http://www.orfeo-toolbox.org/packages/OTBSoftwareGuide.pdf>
- [16] J. Grodecki and G. Dial, IKONOS Geometric Accuracy, 2001

10. Anexos

- I. Especificación de requerimientos
- II. Alcance del sistema
- III. Descripción del proyecto
- IV. Análisis de riesgos
- V. Cronograma
- VI. Arquitectura
- VII. Manual de instalación
- VIII. Manual de usuario
- IX. Satélites en el mercado
- X. Actas de las reuniones con el tutor
- XI. Parámetros de configuración