

ESCUELA UNIVERSITARIA DE INGENIERIA. VITORIA-GASTEIZ

GRADO EN INGENIERÍA EN GEOMÁTICA Y TOPOGRAFÍA

Curso 2016/2017

TRABAJO DE FIN DE GRADO

APLICACIONES DE LOS LENGUAJES DE PROGRAMACIÓN A LA INTELIGENCIA ARTIFICIAL, LOS CÁLCULOS GEOFÍSICOS COMPLEJOS Y LA REALIDAD AUMENTADA.

Autora: Saiz Álvaro, Fátima

Director: García Gracianteparaluceta, Beñat



Fecha: Vitoria, a 6 de junio de 2017

Contenido

Índice de ilustraciones.....	4
Índice de ecuaciones	5
Índice de tablas.....	6
Agradecimientos	7
1. Resumen	8
2. Introducción	10
3. Objetivo.....	12
4. Estado del arte.....	14
5. Metodología	18
5.1. Planificación.....	18
5.2. Recorrido	19
5.3. Experimentos desarrollados.....	21
5.3.1 Educativo: Caja de arena topográfica	21
5.3.2. Big Data: Reconocimiento de objetos	24
5.3.3. Científico: Cálculo de la gravedad en superficie a partir de armónicos esféricos y corrección de anomalías gravimétricas.....	25
6. Valoración económica.....	32
7. Conclusiones	33
8. Líneas futuras de trabajo	36
9. Bibliografía.....	37
10. Anexos	40
ANEXO 1. Uso de la Kinect como Laser Scanner en Windows	40
ANEXO 2. La cámara Kinect 360	43
ANEXO 3. Acceso y grabación de los datos que ofrece Kinect.....	45
ANEXO 4. Nubes de puntos con Kinect y librería PCL.....	49
ANEXO 5. Instalación completa	51
ANEXO 6. Instalación OpenCV en Eclipse Neon	56
ANEXO 7. Conectar la cámara Web y realizar una fotografía	62
ANEXO 8. Reconocimiento facial con imagen estática	65
ANEXO 9. Detección de matrículas.....	67
La inteligencia artificial.....	67

Detección de matrículas	69
ANEXO 10. Operaciones con el valor de los pixeles de una imagen	83
Variación de la gravedad	83
ANEXO 11. Armónicos esféricos para el cálculo del potencial.....	89
ANEXO 12. Gravedad en superficie calculada a partir de parámetros GRACE 2016	91
ANEXO 13. Código para el cálculo de la gravedad en superficie.....	92
ANEXO 14. Correcciones de la gravedad	96
ANEXO 15. Funcionamiento del programa de correcciones gravimétricas	99
ANEXO 16. Valoración económica.....	102

Índice de ilustraciones

Ilustración 1 Estructura caja de arena topográfica	22
Ilustración 2 Tamizado de arena	23
Ilustración 3 Detección de matrículas	24
Ilustración 4 Diferencias obtenidas en España	28
Ilustración 5 Gráficos de dispersión según la latitud y la longitud	29
Ilustración 6 Superficie calculada a partir de parámetros GRACE 2016	29
Ilustración 8 Objeto escaneado con ReconstructMe	41
Ilustración 9 Habitación escaneada con Skanect.....	41
Ilustración 10 Pared escaneada con Skanect.....	42
Ilustración 11 Partes de la cámara Kinect (Magallón, 2013)	43
Ilustración 12 Datos obtenidos a partir de Freenect.....	46
Ilustración 13 Ejecución Fakenect	47
Ilustración 14 cpp_pcview	49
Ilustración 15 Nube de puntos generada	49
Ilustración 16 Logo PCL.....	50
Ilustración 17 Crear nuevo proyecto C++ en Eclipse	58
Ilustración 18 Elección de compilador.....	59
Ilustración 19 Configuración OpenCV (1).....	60
Ilustración 20 Configuración OpenCV (2).....	61
Ilustración 21 Abrir webcam con OpenCV y capturar una imagen	62
Ilustración 22 Imagen almacenada en la carpeta del proyecto	63
Ilustración 23 Haar-like features	65
Ilustración 24 Matrícula española Fuente especificada no válida.	69
Ilustración 25 Object Marker	70
Ilustración 26 Lista de imágenes positivas.....	75
Ilustración 27 Resultado createsamples	76
Ilustración 28 Creación de cascada	78
Ilustración 29 XML generado	79
Ilustración 30 Resultados obtenidos detección de matrículas.....	82
Ilustración 31 Formato de grabación BIL Fuente especificada no válida.	83
Ilustración 32 Corrección aire libre.....	84
Ilustración 33 Imagen clasificada por intervalos.....	85
Ilustración 34 Corrección topográfica.....	97
Ilustración 35 Teoría Airy-Heiskanen corrección isostática.....	98
Ilustración 36 Programa correcciones gravimétricas.....	99
Ilustración 37 Datos de salida. Correcciones en el MDT de Bizkaia.	100
Ilustración 38 Datos de entrada. MDT Bizkaia.	100

Índice de ecuaciones

Ecuación 1 Geopotencial	17
Ecuación 2 Gravedad en superficie	17
Ecuación 3 Gradiente del potencial	25
Ecuación 4 Corrección de la gravedad según altura (Desarrollo)	84
Ecuación 5 Corrección de la gravedad según altura (Final).....	84
Ecuación 6 Geopotencial	89
Ecuación 7 Gravedad en superficie	89
Ecuación 8 Gradiente del potencial	89
Ecuación 9 Potencial centrífugo.....	89
Ecuación 10 Derivadas del geopotencial	90
Ecuación 11 Derivadas del potencial centrífugo	90
Ecuación 12 De coordenadas geodésicas a geocéntricas.....	90
Ecuación 13 De geocéntricas a esféricas	90
Ecuación 14 Gravedad en un punto cercano	96
Ecuación 15 Corrección aire libre	96
Ecuación 16 Corrección Bouguer	96
Ecuación 17 Corrección topográfica	97
Ecuación 18 Corrección isostática	98

Índice de tablas

Tabla 1 Comparación de la gravedad observada por el IGN con la calculada por el programa.....	27
Tabla 2 Correlación de los datos	28
Tabla 3 Gastos fijos	102
Tabla 4 Amortizaciones.....	102
Tabla 5 Gastos de asignación directa caja de arena	103
Tabla 6 Gastos de asignación directa geofísica.....	103
Tabla 7 Gastos de asignación directa reconocimiento de matrículas	103
Tabla 8 Total caja de arena topográfica.....	104
Tabla 9 Total cálculos geofísicos	104
Tabla 10 Total reconocimiento de matrículas	104

Agradecimientos

A mis padres, por quererme mucho, creer en mí y por apoyarme. A mis abuelos, por estar conmigo con tanta ilusión.

A mi pareja y amigos, por alentarme en todo momento.

A mis profesores, en especial a Beñat, por su apoyo, su tiempo, sus consejos y su motivación. Al personal de la universidad por su colaboración y gran ayuda.

Y a toda la gente que he ido conociendo y ha estado a mi lado a lo largo de estos años.

Todo este trabajo ha sido posible gracias a ellos, gracias por ayudarme y por aportar cosas buenas en mi vida.

1. Resumen

Dentro del área de conocimiento de la ingeniería geodésica, topográfica y fotogramétrica se han realizado avances en los últimos años. Cabe destacar la combinación de estos con la informática, que avanza desde finales del siglo XX. En todos estos campos, como son la cartografía, la geodesia, la teledetección, la fotogrametría, etc. se ejecutan procesos que son susceptibles de ser programados.

La alta competitividad actual exige unos productos cada vez de mayor calidad, de las antiguas imágenes estáticas se ha pasado a representaciones mucho más visuales que aplican técnicas de realidad aumentada y en 3D. Esta facilidad de interpretación visual hace de estas soluciones una herramienta didáctica muy potente.

En este trabajo final de grado (TFG) mediante el desarrollo de tres experimentos se ha trabajado sobre la potencialidad de la geoinformática, abordándose desde tres puntos de vista diferentes.

El primero ha sido un enfoque didáctico, combinando la potencialidad de la Kinect de Xbox, con un cajón de arena, un proyector y programación en C++, se consigue un producto final que enseña conceptos hidrológicos, geográficos y geológicos de una manera lúdica.

El segundo se ha desarrollado desde un punto de vista científico, el cálculo del geopotencial de la gravedad a partir de armónicos esféricos. Por otra parte se han calculado las correcciones de la gravedad a partir de imágenes tiff.

Por último, en el campo de la inteligencia artificial mediante un enfoque de análisis masivo de datos y detección de patrones, se ha desarrollado un programa de identificación de matrículas.

Son solo unos ejemplos de lo que se puede hacer hoy en día y un punto de partida hacia retos mayores.

Palabras clave: realidad aumentada, Kinect, clasificadores, inteligencia artificial, geomática, reconocimiento de objetos, programación, gravedad.

Abstract

In recent years substantial progress has been made in Geodesic, Topographic, and Photogrammetric engineering areas. It is noteworthy how this has occurred simultaneously with the development of the computer science, especially in the last decades. In the development of related areas such as cartography, geodesy, remote sensing, or photogrammetry easily programmable processes are carried out.

Nowadays, the increasingly higher level of competitiveness requires even more quality products. Therefore, the use of new techniques like augmented reality or 3D representation has been encouraged. Furthermore, the focus on increasingly visual and intuitively accessible tools makes the aforementioned solutions very powerful didactic means.

In this Bachelor Thesis, experimental work has been conducted within three different fields in which the combination of these techniques is of particular interest.

The first of them is a didactic approach on which combining the potential of the Kinect device for Xbox, a sandbox, a projector, and C++ code. The resulting application could be used to teach hydrological, geographical, and geological concepts in an entertaining and user-friendly way.

The second one is a calculation of the gravity geopotential from spherical harmonics using a scientific approach. Additionally, corrections on the gravity have been estimated from a TIFF image.

Finally, an application for license plate detection has been created from a massive data analysis and pattern detection approach.

These are just few examples of the actually available applications, which constitutes a starting point to greater upcoming challenges in the field.

Keywords: augmented reality, Kinect, classifiers, artificial intelligence, geomatics, object recognition, programming, gravity.

2. Introducción

Este trabajo de fin de grado pretende relacionar la potencialidad de la programación con la geomática. La combinación de ambas es un frente abierto en proceso de desarrollo e investigación.

Es un hecho que la automatización de procesos mejora la productividad. Todas las ingenierías deben innovar y adaptarse conforme avanza la tecnología, incluyendo como no, la geomática y la topografía. Es tanta la relación que presenta con la informática, que se ha acuñado un nuevo término, la geoinformática.

Fundamentado por tanto en la realidad aumentada, el tratamiento masivo de información y la inteligencia artificial, se desarrollan competencias de la geomática desde las perspectivas educativa, científica y tecnológica.

La realización de la caja de arena está más enfocada a fines educativos, ya que es el usuario el que tiene el protagonismo. Puede jugar a crear conjuntos de montañas, caminos, hoyos, etc. Se vuelve más interesante cuando se coloca la mano encima de la arena como si se tratara de una nube y cae agua virtual. Esto permite crear ríos, presas o cascadas. El alcance que tiene trabajar con estas tecnologías es infinito, el poseer una cámara Kinect significa tener una herramienta muy potente que no para de crecer. Con esta cámara se puede tener un láser scanner capaz de detectar objetos de 2,5cm de tamaño en el entorno de una habitación con muy poca luz gracias a su sensor de infrarrojos. Otro punto positivo de esta cámara es la posibilidad de desarrollo que tiene, es decir, todas las aplicaciones que se pueden generar sobre nubes de puntos, tratamiento de imágenes, grabación de videos, etc. Puede ser empleada en control de calidad de piezas, en seguridad para la identificación de objetos, en generar modelos 3D de objetos o espacios, y un montón de posibilidades más aún por explorar.

En cuanto a la perspectiva científica, está el desarrollo del cálculo de la gravedad en superficie para realizar nivelaciones de alta precisión minimizando el uso de un gravímetro, ya que el cálculo a partir de armónicos esféricos es el que hasta ahora ha demostrado que más se ajusta a la realidad. Para la representación de estas superficies de nivel complejas se acude a la resolución de la ecuación de Laplace, pero en este caso la solución viene dada por un polinomio especial. El cual tiene en cuenta que la gravedad presenta variaciones de baja frecuencia también en longitud. Este polinomio es conocido como la expansión de armónicos esféricos del potencial V . Esta superficie que se genera es la representación matemática más aproximada que se puede realizar del geoide (figura de la Tierra). (Devesa, Martín Furones, & García García, 2001). Por otra parte, se calculan a partir de un modelo digital del terreno (MDT) en formato TIFF las correcciones de la gravedad pixel a pixel. Las observaciones de la gravedad se realizan sobre la superficie terrestre que es irregular y sobre la que se tendrá una superficie equipotencial diferente para cada punto observado, es decir, que no se tendrán todas las medidas referidas a la misma superficie de nivel, por lo que se

deberá reducir a una única superficie de nivel (el geode) para que los valores de gravedad observados sean comparables entre si y permitan formarse una idea del relieve gravimétrico.

Finalmente, la herramienta de visión artificial para la detección de matrículas está relacionada con la parte de Big Data. En este experimento se utilizan un conjunto de datos, que en este caso son imágenes, muy extenso y se utilizan procedimientos para encontrar patrones repetitivos dentro de estas. Las aplicaciones de este tipo tienen muchas líneas de desarrollo actualmente en temas de seguridad o en mobile mapping.

Todo esto ha llevado a enfocar este TFG a conocer y profundizar todas las vías de investigación y desarrollo que presentan la combinación de estos conocimientos.

Este trabajo se ha desarrollado del siguiente modo: en primer lugar se definieron los objetivos, posteriormente se desarrolló una importante labor de investigación en especial a través de búsquedas en bases de datos indexadas con el objetivo de conocer los últimos avances en estos campos. Para el correcto desarrollo del trabajo, se establecieron una serie de hitos que permitieran conocer el grado de cumplimiento del proyecto y a su vez ayudaran a mantener una línea bien definida. Mediante el cumplimiento de estos hitos, se desarrolló todo el proceso de aprendizaje. Finalmente, en el apartado de conclusiones se reflexiona sobre el contenido de este trabajo.

Como elementos adicionales, se ha considerado efectuar una valoración económica, unas posibles líneas de desarrollo futuras y se hace entrega de un CD con toda la documentación aquí contenida.

En todo momento se ha intentado generar una documentación lo más completa posible pero tratando de que fuera sencilla de seguir.

3. Objetivos

Es evidente que el objetivo de este trabajo es el de cubrir la exigencia académica para la obtención del título de graduada. Sin embargo, se ha entendido como una oportunidad para investigar y trabajar en el campo de la informática combinado con la formación que le es inherente a la ingeniera en geomática.

Se estableció como objetivo primordial conocer el método científico para investigar las tendencias actuales en estos campos, viendo los proyectos y aplicaciones que se han desarrollado en los últimos tiempos. Se busca aprender a utilizar y descubrir las bases de datos bibliográficas de resúmenes y citas de artículos de revistas científicas que ayuden a profundizar y centrar el tema.

Desde un principio se evidenció la necesidad de aprender a manejar Linux, siendo un sistema operativo muy utilizado en ambientes científicos, por lo que su aprendizaje constituyó uno de los primeros objetivos.

Se deseó profundizar en el funcionamiento y calibración de las cámaras que combinan los datos de color RGB más los D de profundidad, buscando la generación de nubes de puntos a partir de imágenes digitales, por ejemplo, empleando la Kinect de Xbox. Se tratará de buscar puentes para exportar las nubes en formato propietario a estándares abiertos.

Otro de los objetivos a alcanzar es el aplicar y utilizar librerías gráficas en técnicas de cálculos complejos y con su integración simplificar la utilización de los programas de cálculo creados. También explotar las librerías OpenCV en aplicaciones fotográficas y audiovisuales.

Trabajar con las imágenes tiff de los modelos digitales del terreno y operar con ellas es otro objetivo deseado, accediendo al nivel de pixel y gestionarlo y manipularlo conservando los datos geoespaciales. Obtener así la gravedad en el geoide calculando las correcciones a realizar pixel a pixel.

Para lograr el objetivo anterior, será necesario previamente la realización de un cálculo numérico complejo. Es por tanto otro de los objetivos encontrar la manera de enfocarlo para minimizar tiempo de cálculo y simplificar el trabajo. Se compararán los resultados obtenidos con datos de mediciones reales y los residuos servirán por un lado para diagnosticar la bondad del cálculo realizado, y por otro lado, para barajar diferentes hipótesis de mejora.

Encontrar una aplicación lúdica e interactiva que permita al usuario conocer conceptos topográficos e hidrológicos. Para ello reproducir la caja creada por la UC Davis e implantarla en la Escuela de ingeniería de Vitoria-Gasteiz. Explicar paso a paso de manera sencilla toda la instalación de forma que cualquier persona pueda reproducirlo y dar unas pautas y recomendaciones acerca de problemas que se han ido encontrando.

Otro objetivo es trabajar con Big data en tratamiento masivo de información para aplicaciones de inteligencia artificial, sobre todo en reconocimiento de patrones. Gracias a esto ser capaz de manipular grandes cantidades de información ya que es un tema que está de plena actualidad y se trata de un sector que está en plena expansión.

Para completar este proyecto, se definió como objetivo el hacer una estimación del coste que tendría el proyecto en el mundo real. Y estandarizar en la medida de lo posible este proceso que permita realizar presupuestos de una manera autónoma, ágil y objetiva.

Por último, conocer todos los avances que ha habido dentro de este sector y las posibilidades de investigación y desarrollo que hay en temas de realidad aumentada e inteligencia artificial, así como profundizar el nuevo uso de estas técnicas en diversos campos.

4. Estado del arte

Con el fin de aclarar qué aporta este proyecto al campo de la topografía y la programación, se darán a continuación unos conceptos de términos empleados en el trabajo y se repasarán los últimos avances relacionados en este campo.

Actualmente está de moda el uso de nuevas tecnologías que permitan al usuario combinar el mundo real con el virtual. Desde hace tiempo la realidad aumentada está presente en videojuegos, por los cuales a través de una cámara o de unos sensores permitían el control del mismo. Esta moda, sin embargo, va más allá de los videojuegos y se está aplicando en la vida real en aspectos de ocio o marketing como pueden ser los catálogos virtuales, y en aspectos educativos, científico-tecnológicos y robóticos.

La Augmented Reality o Realidad Aumentada (a partir de aquí, AR) consiste en combinar el mundo real con el virtual mediante un proceso informático, enriqueciendo la experiencia visual y mejorando la calidad de comunicación. Se encarga del estudio de técnicas que permiten la integración en tiempo real entre el mundo real (a partir de imágenes capturadas u otra información como la geolocalización) y el mundo virtual (que normalmente añade información digital, cálculos o modelos 3D). (Serrano L. P., 2011)

La cual tiene aplicaciones de todo tipo en casi todos los campos como se ha mencionado previamente. En publicidad, en la prensa escrita introduciendo contenidos adicionales, en videojuegos, en arquitectura, medicina, educación, automóviles, etc.

Otro concepto a introducir es la Inteligencia Artificial, que es la inteligencia exhibida por máquinas. Se entrenan máquinas para aprender y resolver problemas como si se tratase de un ser humano. Más adelante se desarrollará mejor este concepto. Las aplicaciones son numerosas como lingüística computacional o robótica entre otros. La idea que subyace es enseñar a las máquinas al autoaprendizaje.

Enfocando estos avances en temas geomáticos y topográficos, se han hecho diversas investigaciones y aplicaciones.

Como mencionaron en la 7ª Conferencia IEEE de juegos de entretenimiento multimedia, se están empleando actualmente sistemas LIDAR para crear representaciones de un área en 2D y 3D (McCormack, 2015). Presentando así otro ámbito en el que emplear la cartografía o los modelos 3D.

Se desarrolla el uso de las cámaras que ofrecen datos RGB más la profundidad de cada pixel para crear escenas interiores 3D. Así consiguen un modelo de la zona útil para la navegación de robots, generación de cartografía o aplicaciones de la realidad aumentada como se muestra en el anexo 1. (Elchaoui Elghor, 2016)

La Kinect, es un sensor de este tipo, ofrece la información de color en las bandas RGB y por otro lado la profundidad en una banda llamada D, de ahí el término RGB+D que se está empleando últimamente.

Otra aplicación que además últimamente es muy demandada, es la digitalización y visualización de elementos de patrimonio cultural como son los sitios arqueológicos, edificaciones y monumentos.

En resumen, la representación de la realidad en un dispositivo tecnológico está en auge y es de gran utilidad en muchos ámbitos. Como por ejemplo la aplicación de esta para la mejora de la seguridad en un entorno industrial, dirigiendo al trabajador mediante una aplicación implementada en dispositivos móviles que da instrucciones de procedimientos de trabajos y de seguridad que se deben realizar. (Tatić, 2017)

Respecto al ámbito educativo, se está investigando sobre cómo implementar la realidad aumentada al mundo de la traducción, de manera que a través de la cámara de un teléfono se enfoquen textos de carteles, impresos, señales, etc. y se traduzcan automáticamente. (Wołk, 2017)

Son muchos los autores que hablan de cómo mejorar el mundo de la medicina haciendo así las operaciones más precisas y seguras utilizando cámaras RGB-D como es la Kinect. Con la AR se crean simulaciones y así se pueden planificar las trayectorias de la aguja, meditar la estrategia que se va a seguir en una operación, robotizar las operaciones o tener una guía en las mismas. (Girbacia, 2017) (Bernhardt, 2016)

Por tanto, la realidad aumentada va mucho más allá de los videojuegos y queda demostrado que se está implementando cada vez más en todos los aspectos de la vida.

Por otra parte, se ha desarrollado el campo de la visión artificial, que trabaja en la detección de objetos automáticamente. Esto puede tener aplicaciones como el inventario de objetos (señales de tráfico), la conducción autónoma, el difuminado de caras o matrículas para el mobile mapping o en temas de seguridad (como puede ser el reconocimiento de caras o armas).

La idea de construir una máquina que pueda ejecutar tareas percibidas como requerimientos de inteligencia humana es siempre algo interesante. El término Inteligencia Artificial surge en 1950 a raíz de que Donald Hebb desarrolla un algoritmo de aprendizaje para dichas redes neuronales. (Pulido, 2011)

En cuanto al ámbito del cálculo científico de la gravedad, cabe introducir algunos conceptos sobre la importancia de la gravedad en la nivelación. Las redes de nivelación de precisión constituyen el almacén fundamental sobre el que se sustenta la altimetría del territorio. Se entiende por red de nivelación a la formada por una serie de marcas o construcciones efectuadas sobre el terreno con el fin de determinar sobre ellas la diferencia de altitud respecto al nivel medio del mar que cada país adopte como origen, en España se toma en Alicante. (Furones, 2000)

Antiguamente las redes geodésicas dispuestas eran tres, de primero, segundo y tercer orden con precisiones progresivamente decrecientes. Actualmente se dispone de tres redes nacionales. La red ROI (Red de Orden Inferior) con una precisión entre 10 y 30 cm, REDNAP (Red de Nivelación de Alta Precisión) con observación de nivelación geométrica y gravimétrica y REGENTE (Red Geodésica Nacional por Técnicas Espaciales) cuyo fin es unificar los marcos de referencia de la península y los archipiélagos.

A nivel del País Vasco, existen otras redes altimétricas, como son las de cada provincia, gestionadas por las Diputaciones Forales y por otro lado, el intento de Gobierno Vasco por unificarlas en una única red.

Desde hace muchos años se sabe que es necesario introducir observaciones gravimétricas en las nivelaciones de alta precisión, no solo desde el punto de vista teórico si no como se puede también observar en las diferencias obtenidas del análisis estadístico realizado por Adolfo Martín Asín con la Red de Nivelación de Precisión antigua. (Asín, 1959)

Con vistas a una mejora de la REDNAP se ha programado una herramienta que permite calcular la gravedad en la superficie terrestre dados la latitud, la longitud y la altura. Este cálculo se ha realizado utilizando los armónicos esféricos, que es la mejor aproximación del cálculo del potencial, tanto del campo gravitatorio como de la electrostática.

Para realizar correctamente el cálculo de una nivelación, no se pueden realizar mediciones geométricas exclusivamente, el paso de una superficie equipotencial a otra se basa en el incremento de trabajo. La cota geopotencial de un punto A está asociada tanto a la gravedad media existente a lo largo de la línea de la plomada, como la distancia geométrica que le separa de un punto teórico O en el geoide.

Tradicionalmente en Euskadi, los cálculos de la NAP se mandaban al Instituto Geográfico Nacional. Con esta herramienta, aunque no se consiga la misma precisión que con un gravímetro, se consigue un valor aproximado, que en caso de no disponer de ninguno, ofrece resultados mejores que no teniendo en cuenta esta corrección y no presenta limitaciones a la hora de diseñar las líneas de nivelación.

Este geopotencial se mide en ugp y se calculará multiplicando la gravedad en superficie (g) por el desnivel en kilómetros. Es por eso que es necesario calcular la gravedad en superficie si no se dispone de un gravímetro para medirla.

Para obtener la gravedad en superficie se emplearán armónicos esféricos aplicados a la geofísica. Los coeficientes de los armónicos esféricos representan la estructura global y las irregularidades del campo geopotencial de la Tierra. Todo este desarrollo se explica en el libro Physical Geodesy de Hofmann-Wellenhof & Moritz (Bernhard Hofmann-Wellenhof, 1967).

La ecuación que lo representa es la siguiente:

$$W_a(r, \lambda, \varphi) = \frac{GM}{r} \sum_{l=0}^{lmax} \sum_{m=0}^l \frac{R^l}{r} P_{lm}(\sin \varphi) (C_{lm}^W \cos m\lambda + S_{lm}^W \sin m\lambda)$$

Ecuación 1 Geopotencial

Donde:

(r, λ, φ) : Coordenadas esféricas geocéntricas del punto a calcular

R: Radio de referencia

GM: Producto de la constante gravitacional y la masa de la Tierra

l, m: Grado y orden de los armónicos esféricos

P_{lm} : Normalización de las funciones de Legendre

C_{lm}^W, S_{lm}^W : Coeficientes de Stoke

Para calcular la gravedad en superficie se necesita el gradiente del potencial:

$$g = |\Delta W|$$

Ecuación 2 Gravedad en superficie

5. Metodología

5.1. Planificación

Este proyecto nace de la idea de combinar la pasión que siente la autora por la programación con los conocimientos que se han adquirido en el campo de la topografía a lo largo de estos años de estudio.

Dentro del gran abanico de oportunidades que ofrece, se decidió enfocar el proyecto en tres líneas concretas educativa, científica y tratamiento masivo de información.

Para alcanzar los objetivos, se diseñaron tres experimentos, marcados por unas fases o hitos donde poder verificar los avances conseguidos y mantener la línea prevista inicialmente evitando el desvío del objetivo establecido.

5.2. Recorrido

Este trabajo se ha comenzado realizando una investigación sobre los proyectos actuales y se han marcado unos objetivos a alcanzar. Es el punto de partida de cualquier proyecto, antes de desarrollar los experimentos es necesario investigar el estado actual y analizar las soluciones similares que ya están creadas. Una vez definidos los objetivos, se han marcado y logrado una serie de hitos para planificar y organizar el trabajo a realizar y así conseguir los requisitos que ha de satisfacer el proyecto. Estos hitos se enuncian a continuación.

Primeramente, para conseguir realizar la caja de arena topográfica, se ha trabajado con la tecnología RGB-D, en concreto con la cámara Kinect. Como primer paso, se han trabajado las conexiones, en concreto la de la Kinect y el ordenador con el sistema operativo Ubuntu, para el que no existen drivers de conexión como con Windows. Una vez realizada esta conexión a partir de herramientas de código abierto, se han realizado experimentos con esta, como son la calibración o la obtención de imágenes RGB y de profundidad y nubes de puntos. Continuando con las conexiones, se ha enlazado el proyector con el ordenador, este paso costó más de lo normal por temas de compatibilidad en drivers y conversores de VGA a HDMI ya que se contaba con un proyector de salida VGA, por lo que se optó a comprar un proyector con salida HDMI directa. Después de verificar que todas las conexiones eran correctas, se ha construido la estructura de la caja de arena a partir de las recomendaciones dadas por la UC Davis y de otras instituciones que la han montado también. Se ha contado con personal de la universidad para lograr este paso. A continuación, se han ajustado los dispositivos a la misma, de modo que el campo visual de los dispositivos abarcase todo el recinto de la caja. Una vez obtenido el montaje completo, se ha instalado el software de UC Davis y se han hecho las calibraciones necesarias, documentando cada paso dado de una manera comprensible para que cualquier usuario futuro pueda reproducirla.

En referencia al reconocimiento de patrones, como primer paso se han instalado la librería gráfica OpenCV, por su gran capacidad en temas de visión artificial y encima ser de código abierto. Se ha introducido en Eclipse vinculando todas las librerías que OpenCV posee, porque si no los programas generados no reconocen las funciones utilizadas. Como no se conocía el lenguaje C++, se han realizado programas básicos para no abordar directamente los programas de inteligencia artificial, los cuales algunos se adjuntan en los anexos. Posteriormente se han explotado las funciones más conocidas de esta librería gráfica, como por ejemplo la conexión con la WebCam y la exportación de imágenes y vídeos desde esta y su tratamiento. Se ha accedido a bases de datos de imágenes para crear las muestras para el entrenamiento del clasificador, para ello se necesitan imágenes positivas en las que aparezca el objeto a reconocer y negativas. Una vez obtenidas las muestras, se han listado los elementos en un archivo de texto. Después, con toda la información recopilada, se ha entrenado el clasificador, o lo que es lo mismo, se le ha instruido a reconocer el elemento deseado. Tras conseguir esto, se ha utilizado este clasificador en programas, en este caso el desarrollado detecta la matrícula y superpone un cuadrado sobre esta.

Para finalizar, respecto a los cálculos geofísicos, se ha comenzado manipulando imágenes, accediendo a los datos de cada pixel, modificando su valor y aplicando filtros. Esto se ha realizado utilizando lenguaje C++ y Matlab, pero viendo los resultados obtenidos en la ejecución, por su rapidez de procesamiento se ha seguido desarrollando con Matlab. Una vez logrado esto, se han leído imágenes tiff y almacenando los datos de cabecera que poseen, donde vienen los datos espaciales, para posteriormente exportar los resultados y que estos estén georreferenciados. Después de lograr guardar estos datos, se han programado las correcciones de la gravedad pixel a pixel y se han exportado en formato de archivo de texto y en formato tiff.

Posteriormente, se ha investigado sobre el cálculo de la gravedad en superficie, y una vez descubierto que era necesario el gradiente del geopotencial, se han desarrollado las parciales de este y del potencial centrífugo respecto al radio, la longitud y la latitud, las cuales eran necesarias para el cálculo. A continuación, se han buscado los parámetros necesarios para el cálculo de armónicos esféricos. Estos parámetros son los coeficientes normalizados de Stoke y se han utilizado los del satélite GRACE del año 2016. Una vez obtenidos todos los datos para el cálculo se ha procedido a su programación. Tras alcanzar esto, se han normalizado los cálculos, utilizando la misma normalización que los coeficientes de Stoke utilizados, que es la altamente normalizada. Se ha procedido después a la exportación de los resultados y a la comparativa de los medidos con un gravímetro por el IGN a lo largo de toda la península y las islas. Para finalizar, se ha hecho un análisis de las diferencias y se han calculado las correlaciones de los residuos para obtener conclusiones sobre los resultados conseguidos.

5.3. Experimentos desarrollados

En los diferentes experimentos desarrollados, se especifica el material utilizado incluyendo datos de hardware y software cuando se entienda que pueda ser relevante, ya que se ha constatado que se simplifica el camino si se hace la correcta elección en estos apartados.

Diferenciando las técnicas aplicadas en función de su enfoque, se pueden señalar las siguientes aplicaciones:

- Educativa: caja de arena topográfica
- Científico: cálculos geofísicos
- Big Data: reconocimiento de matrículas

5.3.1 Educativo: Caja de arena topográfica

Para la realización de este proyecto se han utilizado principalmente las siguientes herramientas y software:

- Portátil Lenovo Y50 Intel® Core™ i7-4710HQ CPU @ 2.50GHz 8GB de RAM 64bits Nvidia Geforce GTX 860M
- Sistema operativo Ubuntu 16.04
- Kinect Xbox 360 14001
- Proyector Uhappy BL20 2600 Lumenes
- Estructura del cajón con los soportes necesarios para el proyector y la Kinect
- Arena de sílice blanca

La cámara Kinect Xbox, que es la principal protagonista de este proyecto, es un periférico de entrada desarrollado por Microsoft para jugar en la videoconsola Xbox 360 sin necesidad de ningún mando, percibiendo y reconociendo los cuerpos de los jugadores y los movimientos que realizan. Consta de dos cámaras frontales, una convencional de RGB y un sensor de distancia además de una serie de micrófonos. Para conocer más especificaciones técnicas y el funcionamiento de los sensores y las partes de esta cámara, consultar el anexo 2. (Magallón, 2013)

Antes de montar el proyecto, se ha trabajado con la cámara Kinect para familiarizarse con esta tecnología. Las acciones que se han desarrollado son por ejemplo el acceso y grabación de los datos que ofrece la cámara, tanto imágenes RGB como de profundidad como se ve en el anexo 3 o el acceso y descarga en formato genérico de las nubes de puntos obtenidas con la Kinect y la librería PCL, mostrado en el anexo 4.

A continuación se muestran unas recomendaciones para elegir las herramientas basadas en la experiencia adquirida. Respecto al proyector, es aconsejable que sea de tiro corto, porque la altura a la que se tiene que colocar disminuye. La relación aspecto tiene que ser 4:3 para que coincida con el campo de visión de la cámara Kinect, la misma con la que se debe configurar el proyector para que encajen perfectamente y abarquen el recinto de la arena.

La resolución nativa del proyector no es muy importante, porque está limitada por la de la Kinect que es 640x480 pixels. La estructura dependerá del proyector, depende de cómo proyecte se deberá colocar en el centro, o en la base, en este caso su colocación ha sido en la base, esa posición es recomendable para que no interfiera la Kinect en su campo visual. Respecto a los lúmenes, con 2600 es suficiente para que se vea lo proyectado con luz no muy abundante dentro de una habitación. En cuanto a la conexión, se ha tenido que comprar un proyector que tenga salida HDMI, ya que el conversor VGA-HDMI entraba en conflicto con la tarjeta gráfica del ordenador utilizado.

En cuanto a la estructura, se ha montado un cajón con las siguientes dimensiones: 79,5 cm de ancho y 106,2 cm de largo, incluyendo el grosor de la madera que es de 1,6 cm. Tiene una altura de 30cm, suficiente ya que la altura de la arena será de unos 10cm. El formato de está será 4:3 como el campo que abarca la cámara y el proyector, ya mencionado anteriormente.

Para el sustento del proyector, se han colocado dos perfiles de aluminio en ángulo recto de 2m de altura sobre el que irá sujeto, este tendrá cierta inclinación que permitirá abarcar todo el cajón. De esa estructura saldrá hacia el centro del recinto la Kinect, que será sujeta con unas barras de aluminio que permiten el desplazamiento en todos los sentidos para poder centrarla adecuadamente. Se debe comprobar que la Kinect no interrumpa el campo de proyección del proyector.

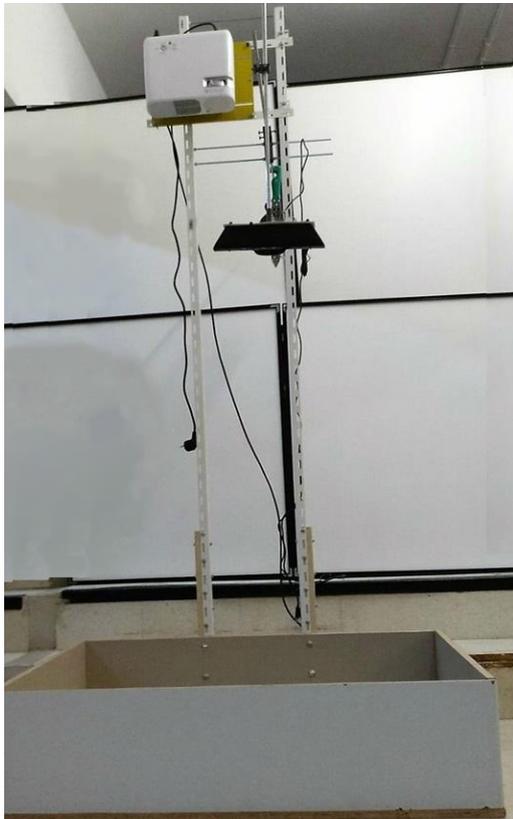


Ilustración 1 Estructura caja de arena topográfica

Por último, se ha empleado arena de sílice con una profundidad aproximada de 10 cm. Por tanto en este caso, se han utilizado 120 kg. Tiene que ser fina y lo más clara posible para que los colores no se distorsionen y para que se proyecten lo más real posible con la luz de la habitación. Para que sea regular y el efecto sea mejor, se ha tamizado después de dejarla secar quitando así todas las piedras y residuos que venían incluidos.

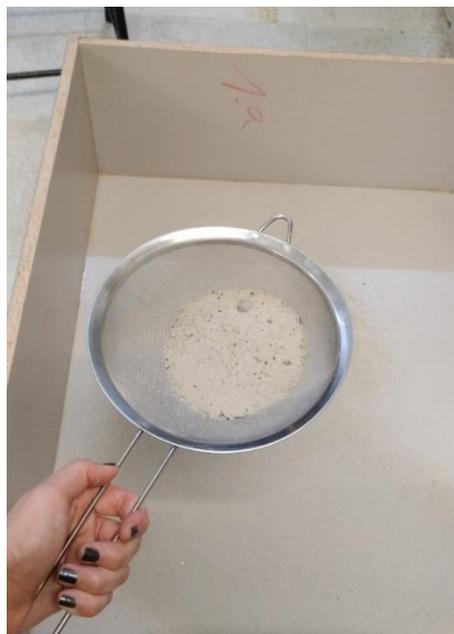


Ilustración 2 Tamizado de arena

Una vez obtenida la estructura completa y viendo que la cámara y el proyector están a la altura ideal, se puede empezar con la instalación y la calibración. Se deben instalar tres grandes software, todos desarrollados por la UC Davis, estos permitirán el control de los dispositivos y aportarán los complementos necesarios para su correcto funcionamiento.

Como resumen de la larga instalación, se debe calibrar primero la cámara individualmente con unos parámetros aconsejados, después, se debe acotar el recinto de la arena y por último hay que calibrar el proyector con la Kinect, creando una especie de puntos de enlace. La instalación completa y detallada se muestra en el anexo 5, una vez realizada, se lanzará el software y estará lista para su utilización.

5.3.2. Big Data: Reconocimiento de objetos

Para esta parte del proyecto se han utilizado las siguientes herramientas y software:

- Portátil Lenovo Y50 Intel® Core™ i7-4710HQ CPU @ 2.50GHz 8GB de RAM 64bits
- Sistema operativo Ubuntu 16.04
- Eclipse Neon
- OpenCV 2.4.13

En este proyecto se desarrolla todo lo necesario para crear un clasificador de objetos y muestra un ejemplo de utilización. El objeto elegido a detectar es la matrícula europea, en particular la correspondiente a España.

Esto se desarrollará utilizando las librerías OpenCV ya mencionadas previamente. En el anexo 6 se explica las diferentes librerías y complementos que se han tenido que instalar para poder operar con estas librerías gráficas. Para trabajar con ellas, se ha tenido que configurar Eclipse, esta configuración de debe realizar cada vez que se quiera desarrollar un programa que emplee OpenCV para que los comandos utilizados sean reconocidos.

Para llegar a saber trabajar con clasificadores, se han llevado a cabo una serie de pasos previos con la librería, como son el realizar y almacenar fotografías y videos a través de la cámara del ordenador como se puede ver en el anexo 7 o profundizando más en la inteligencia artificial, el reconocimiento facial en una imagen estática, desarrollado en el anexo 8.

Con estas herramientas que OpenCV, es posible detectar cualquier objeto en imágenes o videos tanto estáticos como en vivo. Esto es gracias a la posibilidad de trabajar con clasificadores en cascada que se pueden entrenar, obteniendo mejores resultados cuanto mayor sea la muestra de fotos introducida y cuanto más diversas sean las imágenes del objeto a detectar.

Observando las necesidades existentes actualmente respecto a difuminar rostros y matrículas de las imágenes captadas para mobile mapping, se ha pensado en crear una herramienta que detecte las matrículas. El programa creado las detecta en un video tomado en directo y las señala con un rectángulo.

Esto es una muestra de lo que se puede hacer, pero se podrían crear herramientas tan potentes como detección de caras o armas para temas de seguridad.

A continuación se muestra una imagen del resultado del programa. El proceso del entrenamiento del clasificador y el código se adjuntan en el anexo 9.



Ilustración 3 Detección de matrículas

5.3.3. Científico: Cálculo de la gravedad en superficie a partir de armónicos esféricos y corrección de anomalías gravimétricas

Para este cálculo se ha utilizado lo siguiente:

- Portátil Lenovo Y50 Intel® Core™ i7-4710HQ CPU @ 2.50GHz 8GB de RAM 64bits
- Sistema operativo Windows 10
- Matlab

Al comenzar, se optó por realizar estos cálculos en C/C++ en Eclipse con OpenCV, se programó el cálculo de la variación de la gravedad respecto a la altura a partir de un MDT en formato TIFF, donde se explican todos los detalles en el anexo 10. Pero la rapidez al efectuar los cálculos era mínima, es por esto y porque previamente se habían realizado cálculos de armónicos esféricos en Matlab, por lo que se decidió realizar los cálculos en dicho programa.

Los armónicos esféricos son funciones armónicas que representan la variación espacial de un conjunto ortogonal de soluciones de la ecuación de Laplace cuando la solución se expresa en coordenadas esféricas. Por lo tanto, cada potencial armónico que satisface la ecuación de Laplace, se puede expandir en armónicos esféricos. Por esta razón, la parte estacionaria del potencial gravitatorio de la tierra (W_a) en cualquier punto (r, λ, φ) sobre y por encima de la superficie de la Tierra se expresa sumando una expansión armónica esférica dependiendo del orden y el grado.

Los coeficientes de los armónicos esféricos representan la estructura global y las irregularidades del campo geopotencial de la Tierra.

El gradiente del potencial teniendo en cuenta el potencial centrífugo viene dado por la siguiente fórmula:

$$|\Delta W| = \sqrt{[W_{ar} + \Phi_r]^2 + \left[\frac{1}{r \cos \varphi} (W_{a\lambda} + \Phi_\lambda)\right]^2 + \left[\frac{1}{r} (W_{a\varphi} + \Phi_\varphi)\right]^2}$$

Ecuación 3 Gradiente del potencial

El desarrollo completo del cálculo está presente en el anexo 11.

Una vez obtenida la gravedad en superficie se calcula la diferencia de geopotencial entre tramos, el cual se mide en ugp y se calcula multiplicando la gravedad obtenida por el desnivel en kilómetros. Se compensa la nivelación con los geopotenciales en función del desnivel, de las distancias o haciendo una media.

Se ha programado por tanto una función que realiza estos cálculos en Matlab, en la cual se introducen la latitud, la longitud en coordenadas geodésicas y la altitud elipsoidal (WGS84) de un punto y se da la gravedad en superficie en miligales.

Posteriormente se han comparado los resultados obtenidos con los datos que se tenían de las líneas de nivelación del Instituto Geográfico Nacional (IGN). En esta red hay puntos observados y otros calculados. Se han elegido puntos medidos con gravímetro a lo largo de toda la península y de las islas para detectar las posibles causas de las diferencias encontradas.

El IGN no posee mediciones en todos los puntos de las nivelaciones, las mediciones mayormente están en las redes principales. En la siguiente tabla se adjuntan las diferencias entre los datos observados y los cálculos realizados con el programa. Se muestra el número de punto, su posición planimétrica y altimétrica, la medición realizada por el IGN, el valor obtenido con el programa, la precisión con la que se han tomado las coordenadas del punto y la diferencia entre ambos valores.

NUMERO	LONGITUD	LATITUD	ALTITUD	IGN	PROGRAMA	PRECISION	DIF
126	-7.43751667	41.9402167	439	980214.58	980238.7	1	-24.1197
542004	-7.39981889	41.9500325	566.6	980195.52	980200.728	0.1	-5.208
831002	-4.49613889	37.4170361	411.5166	979761.12	979759.819	1	1.301
10404001	-6.33730056	43.3943575	708.2	980350.25	980354.949	0.1	-4.699
62	-4.69976111	41.5370972	759	980072.81	980085.51	1	-12.6999
91	-4.70875556	40.6571864	1119	979935.82	979930.823	1	4.997381
115	-4.36487222	38.0097167	253	979935.43	979943.485	1	-8.054756
528019	-4.3409825	38.1315783	471.6	979892.6	979898.431	0.1	-5.8308899
230	-3.38045833	38.9888361	715	979902.37	979890.632	1	11.7377
822006	-3.32485778	39.0216575	720.62	979902.86	979889.857	0.05	13.0034
10620002	-6.4058175	38.4291319	588.4	979931.76	979917.425	0.1	14.3346
197	-6.3995	38.6844722	402	979994.53	979972.816	10	21.7145
47	-1.63909444	41.3522972	579	980089.55	980116.905	1	-27.3547
214037	-1.44313917	41.1773258	910.5	980008.96	979991.048	0.1	17.9117
631067	-5.29027917	37.7210281	118.8	979948.99	979949.339	0.1	-0.349
20311037	-3.75680278	40.3801694	715	979962.33	979959.888	1	2.4415116
20311097	-3.61108389	40.0694389	544	979989.38	979988.502	0.1	0.877987
20311117	-3.52934167	39.9591583	701	979953.7	979959.591	1	-5.891444
187	-3.77166667	40.3939167	717	979960.37	979961.347	10	-0.9766266
58	-3.92560278	40.5743139	904	979938.13	979942.036	1	-3.9056
161	-6.05687778	37.5431083	78	979973.67	979976.643	1	-2.973
582005	-7.58081111	43.0123917	479	980340.79	980362.906	1	-22.116
175	1.43676667	38.9174528	51	980128.16	980115.776	1	12.38444
87	-3.97107222	39.6798278	804	979936.5	979932.296	1	4.204
22525001	-4.64932778	37.5722639	419	979797.31	979776.77	1	20.54
215	-5.63238833	42.803429	1002.79	980155.71	980168.863	0.05	-13.153
37	-2.05877778	41.4894444	208	980266.08	980249.71	1	16.3696
201056	-2.6333311	43.176575	163	980420.7	980494.3	1	-73.6
10234001	-2.6428889	43.1761111	154	980422.66	980447.015	1	-24.355
555036	-0.23125361	42.6287447	1206.9	980065.1	980098.71	0.1	-33.61
233042	2.65858611	41.7409217	102	980312.59	980319.553	0.1	-6.963
20220179	-0.22047917	40.9892264	463	980115.38	980103.39	0.1	11.99
10225034	0.21807167	40.5099967	390.8	980118.8	980061.9	0.1	56.9
254050	-1.81932778	39.9048928	830.3	979875.49	979842.716	0.1	32.7736
823021	-0.82483333	38.8538611	428.0604	979946	979913.36	10	32.6397
832023	-2.3689444	37.5945278	1137.208	979658.19	979621.683	10	36.507

Tabla 1 Comparación de la gravedad observada por el IGN con la calculada por el programa

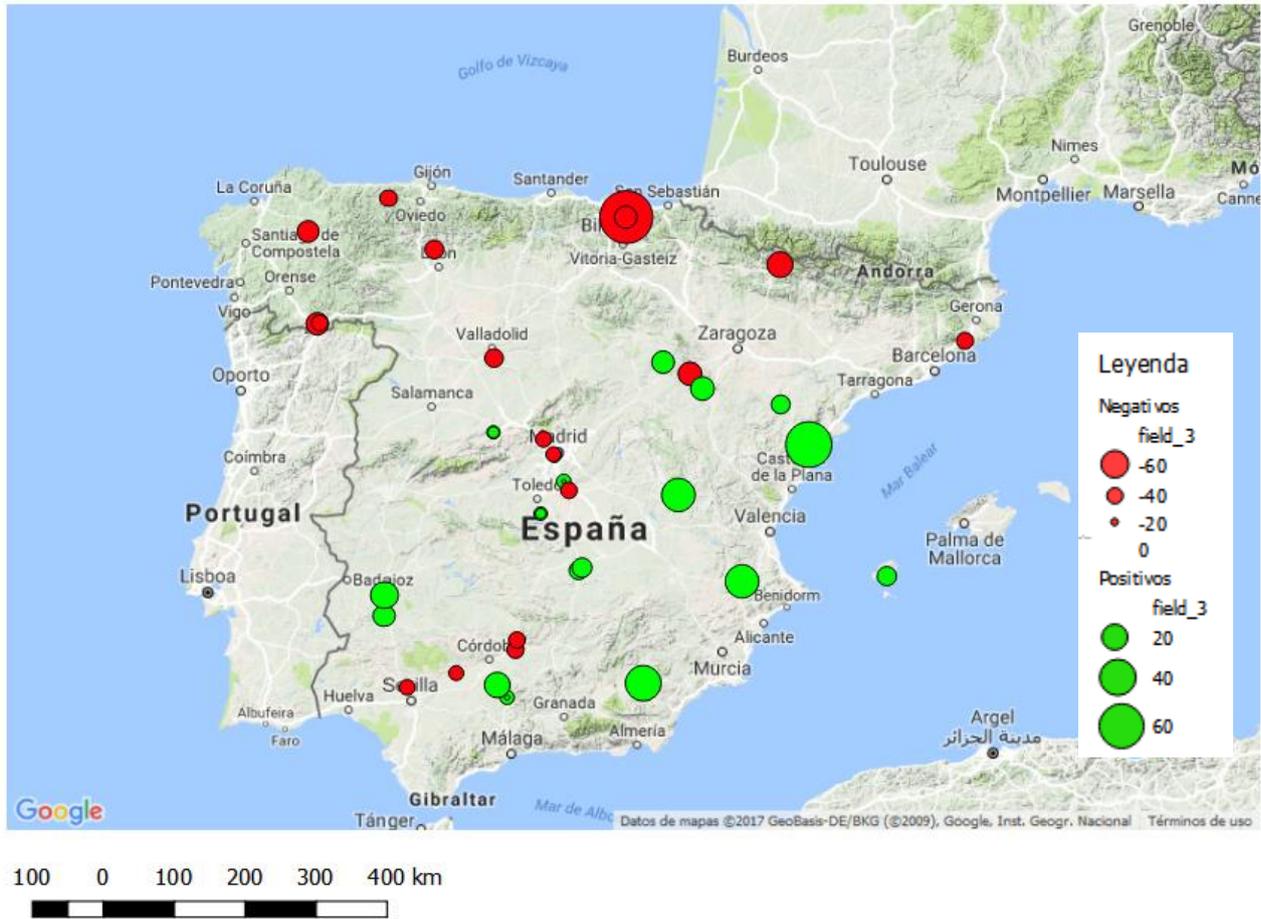


Ilustración 4 Diferencias obtenidas en España

Se calculan las correlaciones entre las diferencias y la longitud, la latitud, la altura y la precisión de la medida de las coordenadas tomadas por el IGN.

CORR. LON	CORR. LAT	CORR. ALT	CORR. PREC
0.3226544	-0.47680967	-0.06824145	0.1952445

Tabla 2 Correlación de los datos

Existe una correlación bastante alta entre la longitud y la latitud, es lógico ya que en el cálculo de los armónicos esféricos hay una componente zonal muy marcada. En la siguiente imagen se muestra el gráfico de dispersión.

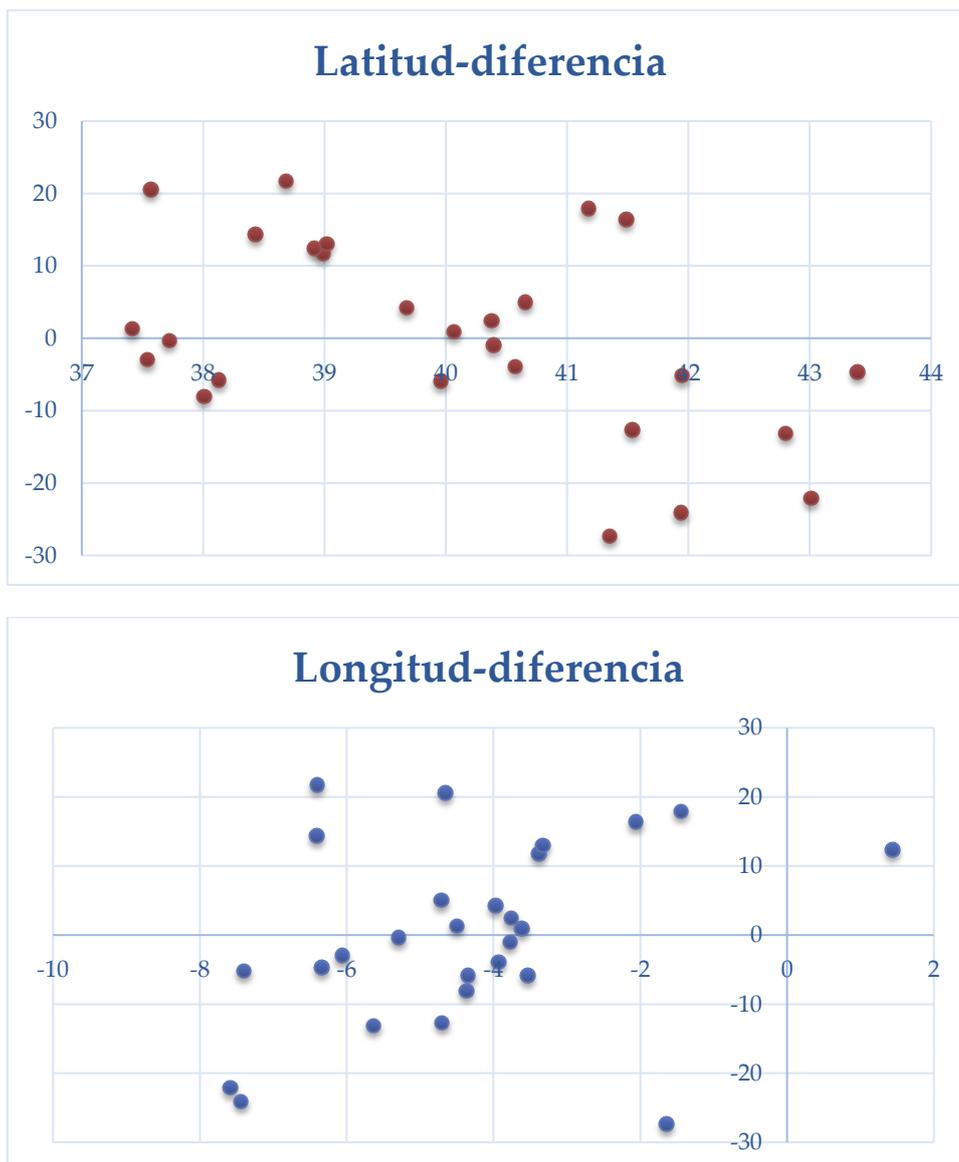


Ilustración 5 Gráficos de dispersión según la latitud y la longitud

Los resultados obtenidos siguen una distribución normal centrada en el cero. Lo cual es muy positivo ya que señala que no existe un error sistemático o salto de escala entre los dos modelos. Por ejemplo, al realizar la media de los residuos entre los dos modelos se obtiene la media de 1 mgal.

En el anexo 12 se muestra el gráfico de la gravedad en el mundo calculada en una malla de 10° que aparece en la ilustración 6.

El código desarrollado se muestra en el anexo 13. Los coeficientes utilizados y todas las funciones utilizadas en el código principal se incluyen en el CD adjunto por su extensión.

GRAVEDAD EN SUPERFICIE CALCULADA A PARTIR DE PARÁMETROS GRACE 2016

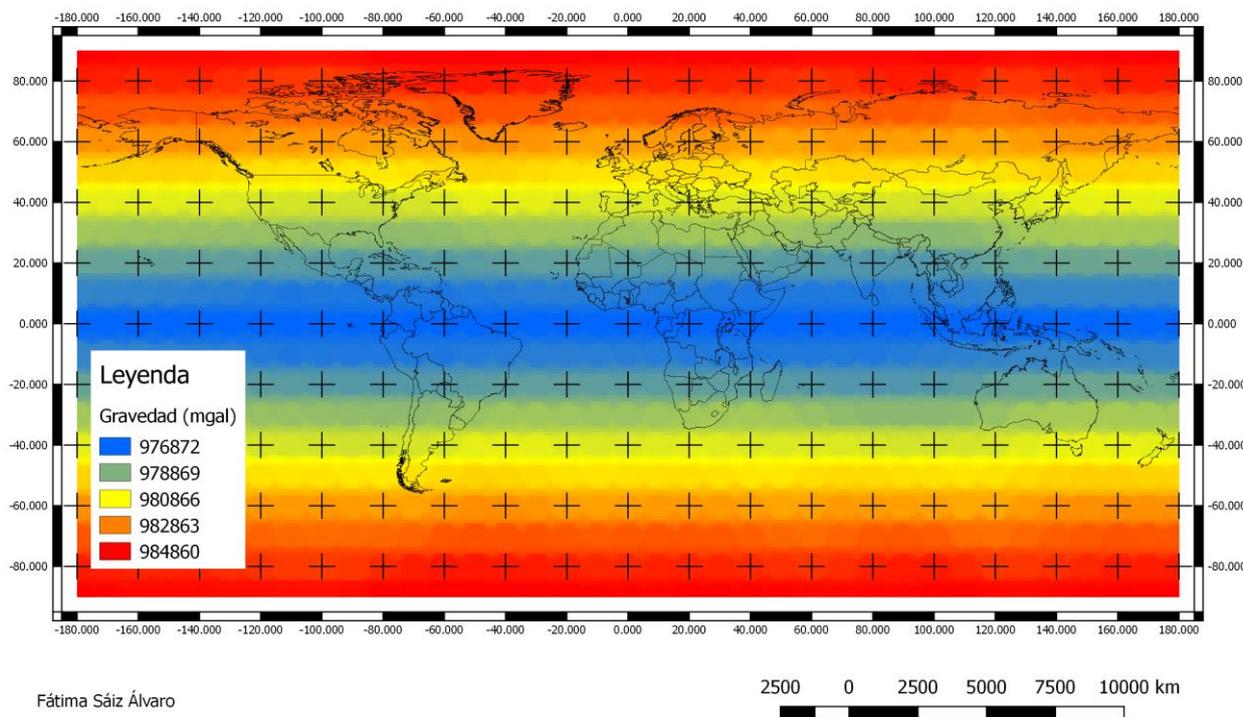


Ilustración 6 Gravedad en superficie calculada a partir de parámetros GRACE 2016

Por otra parte, se han calculado las correcciones de la gravedad. Las correcciones de la gravedad constituyen un dato básico para el análisis y resolución de la forma de la Tierra. Las mediciones de gravedad se realizan en la superficie topográfica y la gravedad normal se determina a nivel de geode, por eso es necesario realizar una serie de correcciones. Para ello se considera por separado cada efecto.

Estas correcciones son: aire libre, Bouguer, corrección topográfica y corrección isostática. Todas ellas vienen explicadas en el anexo 14.

Se ha desarrollado un programa para calcular a partir de un MDT en formato tiff, todas las correcciones inversas para calcular la gravedad en el geode. El funcionamiento del programa se muestra en el anexo 15.

Como resultado, se obtienen las correcciones pixel a pixel en formato Excel y un archivo de texto en el que se incluye la coordenada X, la coordenada Y y la corrección a efectuar. En el cálculo de prueba, se ha introducido un MDT de Bizkaia, con un paso de malla de 200m, el cual se muestra en el anexo 15.

A partir del valor de la altura obtenida a través del MDT, se efectúan las correcciones mencionadas antes. Como se aprecia en las ecuaciones, todas las correcciones dependen de la altura de los puntos. Se crea un mapa de las correcciones según su magnitud.

Las correcciones se corresponden totalmente con las elevaciones del terreno, en especial en las zonas de cambio. Los valores que se muestran en blanco sin datos se corresponden a zonas marítimas. Esto es lógico puesto que la corrección topográfica es mayor cuanto mayor es la diferencia de cota entre un punto y los que tenga más próximos, disminuyendo estos valores con la distancia.

Este trabajo, está especialmente pensado para el empleo de las técnicas gravimétricas en análisis del territorio, es decir, en estudios o prospecciones gravimétricas conducentes al conocimiento de la distribución de masas del subsuelo. Teniendo en cuenta que la corrección topográfica es la más compleja de modelar de todas las que intervienen aunque en valor absoluto no sea ni con mucho la más importante.

6. Valoración económica

Se aborda a continuación un análisis económico, revisando los gastos necesarios para llevar a cabo cada proyecto por separado.

La valoración económica se ha planteado como si de una profesional liberal se tratara, en base a los siguientes argumentos.

Para obtener un presupuesto ajustado a la realidad, se tendrán en cuenta los gastos generales que incluyen los gastos de oficina, y las licencias de software utilizadas habitualmente en una empresa de este tipo. Se incluyen también los asociados a la mano de obra, este se calculará en base a la figura de una autónoma, en este caso, una graduada en ingeniería geomática que de acuerdo a todas las partidas detalladas en el anexo 16, se estima un coste por hora de 20.87€. Este dato y la forma en la que se ha obtenido son de vital importancia en la generación de presupuestos de una manera ágil. Para que la estimación final sea lo más real posible, se tendrán en cuenta los gastos de asignación directa de cada proyecto.

Respecto a los gastos generales, para la correcta realización de los proyectos planteados se ha considerado necesario alquilar una oficina dotándola del equipamiento básico (mobiliario, ordenador, etc.) y estimando los costes habituales de una infraestructura de este tipo (electricidad y agua). Para las labores de contabilidad, impositivas y/o laborales se cuenta con un gasto fijo por parte de una asesoría externa. Se incluye el seguro civil, el teléfono fijo e internet. Todo esto suma por tanto unos gastos fijos de 6.25 €/h en oficina.

Cabe decir que no se valoran costes asociados a desplazamientos o amortización de vehículo ya que para estos proyectos no son necesarios, y en caso de que fuera necesario se alquilará, al igual que cualquier instrumento topográfico. Se asume que esta forma de proceder puede encarecer la hora de campo, sin embargo, para una empresa nueva el invertir en material del que se desconoce la capacidad de amortización real, puede ser causa de fracaso empresarial.

Se ha establecido que el montaje, separado por proyectos, incluyendo la mano de obra, la compra de los equipos necesarios y el IVA al 21% es de:

- Caja de arena topográfica #3406.80€# TRES MIL CUATROCIENTOS SEIS EUROS Y OCHENTA CÉNTIMOS IMPUESTOS INCLUIDOS que incluye el montaje completo, la compra del material necesario para el mismo e instrucciones de uso para su explotación.
- Cálculos geofísicos #8983.31€# OCHO MIL NOVECIENTOS OCHENTA Y TRES EUROS Y TREINTA CÉNTIMOS IMPUESTOS INCLUIDOS que incluye el desarrollo de los programas y la licencia de Matlab, no su explotación.
- Reconocimiento de matrículas #7875.98€# SIETE MIL OCHOCIENTOS SETENTA Y CINCO EUROS Y NOVENTA Y OCHENTA CÉNTIMOS IMPUESTOS INCLUIDOS que incluye el entrenamiento de la cascada y el desarrollo del programa, no su explotación.

En el anexo 16 vienen desglosados todos los precios detalladamente.

7. Conclusiones

Este trabajo ha sido un proyecto complejo con muchos frentes simultáneos que se han ido resolviendo a lo largo del camino. Los hitos marcados se han mostrado útiles para conocer en cada momento el grado de cumplimiento del proyecto.

En cuanto al tema de las tendencias actuales, se ha podido investigar y profundizar en las materias tratadas, y se ha demostrado que estos temas están de actualidad, basta mirar las fechas de las referencias bibliográficas donde doce de las treinta pertenecen al periodo 2015-2017. Para lograrlo se ha hecho una profunda investigación consultando numerosos artículos de carácter científico. La explícita citación de treinta referencias es el mejor argumento para avalar este hecho.

Se ha cumplido con el objetivo de manejar Linux, concretamente Ubuntu, instalando software y haciendo uso de la terminal de comandos porque, más allá de esto, se han creado ejecutables que reducen la introducción de comandos en la terminal agilizando así el trabajo. Se ha mostrado muy estable, es por tanto un sistema operativo muy recomendable por su calidad pero también por el hecho de ser distribuido en código abierto.

La obtención de nubes de puntos con la Kinect se ha logrado apoyándose en la tecnología RGB-D. Se han podido realizar modelos de manera rápida y eficaz a través de distintos software y con la utilización de las librerías PCL, exportándose los resultados a software libre, destacando que por el coste que tienen estos equipos la calidad resulta sorprendentemente alta.

Respecto a las librerías gráficas, destacar OpenCV, con la que se han desarrollado diversas aplicaciones, sobre todo de reconocimiento de objetos en imágenes y vídeos, siendo esta la parte más explotada y más potente de esta librería. Se han utilizado también en Matlab, demostrándose que estas librerías propietarias tienen muy bien controlado el código, es decir, han minimizado los errores del operador.

Se ha logrado cargar, almacenar, manipular y exportar imágenes tiff usando lenguaje C++ y Matlab. Como se ha mencionado, Matlab presenta un código más optimizado y ofrece, en este caso, un procesado más eficaz. Se han conseguido calcular las correcciones de una manera sencilla, rápida e intuitiva gracias a la interfaz gráfica creada. Esta herramienta ofrece correcciones de una sección de terreno muy amplia en un tiempo mínimo por lo que ahorra tiempo de cálculos.

Se ha programado la fórmula de armónicos esféricos aplicados a la teoría del geopotencial. Se ha obtenido con ellos la gravedad en superficie, la cual es necesaria para las nivelaciones de alta precisión. Son cálculos científicos de una gran complejidad por lo que ha sido necesario un profundo análisis del tema. Hay varios tipos de armónicos esféricos y de normalizaciones a utilizar dependiendo de la naturaleza de los cálculos a realizar.

Fue una ardua tarea encontrar la normalización adecuada, finalmente se verificó que la más conveniente era la totalmente normalizada, siendo la que emplea el GOCE y también la más adecuada en los cálculos geofísicos.

En cuanto a los resultados obtenidos con el programa, se hace una valoración positiva basándose en la comparación con los datos obtenidos de las mediciones del Instituto Geográfico Nacional, ya que la diferencia es de una media de 5mgal lo que supone en una nivelación una diferencia de altura entorno a 1cm.

Aun así, los datos obtenidos en toda la península y las islas son positivos, por lo que es una herramienta útil en las nivelaciones. Un error mayor a 1cm en una nivelación de alta precisión es inadmisibles, pero una NAP no se puede tomar como una nivelación geométrica simple, por tanto se necesita una medida gravimétrica para corregir ese error. El error que se posee es de 1cm en valor absoluto, es decir, sin emplear ni una sola medida gravimétrica real. Es por tanto un objetivo del ingeniero correspondiente densificar esa información para bajar de 1 cm a la precisión de una NAP. Pero al utilizar un modelo que se asemeja mucho a la forma de la Tierra, siempre se va a tener el mismo valor independientemente del camino seguido. Con lo cual se va a mejorar si o si el resultado de la nivelación, aunque lo ideal sea medir con un gravímetro.

Como aplicación educativa para temas topográficos se ha implementado la caja de arena topográfica creada por la UC Davis. Se ha documentado en el Anexo 5 el montaje completo paso a paso desde su instalación hasta su calibración para cualquier futura reproducción. Se mencionan a continuación una serie de recomendaciones a tener en cuenta antes de abordar el proyecto.

Cabe destacar la importancia de la elección del portátil a utilizar. Debe cumplir las siguientes características: una buena tarjeta gráfica que sea compatible con Linux y que sea AMD/ATI Radeon o Nvidia GeForce. Tiene que tener una buena CPU pero no es necesaria mucha RAM, con 2GB es suficiente. Respecto al disco duro, son necesarios solo 20GB para la instalación de Linux y del software de la caja de arena topográfica. Es recomendable, que sea Intel Core i5 o Core i7, y que la CPU corra a 3GHz. Para que la simulación de agua no presente problemas, es importante instalar todos los drivers de la tarjeta gráfica que se tenga. No acertar en la elección de un portátil adecuado puede dificultar mucho el proceso.

El portátil que se ha empleado posee una tarjeta gráfica híbrida de nVidia e Intel, que es muy útil y muy potente a la hora de la rapidez ya que elige una u otra dependiendo de la aplicación que se lance. El control de estas en Windows es sencillo, pero en Ubuntu no lo es tanto. Por tanto, se recomienda utilizar un ordenador que solo posea una sola tarjeta gráfica que cumpla con las especificaciones técnicas mencionadas previamente en la memoria, y en el caso de que está presente problemas, actualizar la versión del kernel para solucionarlo.

Así mismo son clave las conexiones. Hay que tener en cuenta que no todos los proyectores ni los ordenadores tienen conexión HDMI, la cual es necesaria para este proyecto. Son detalles que pueden ralentizar el proyecto.

El reconocimiento de patrones necesita de muchos fotogramas para la detección de los objetos, lo que viene a llamarse Big Data por la gran cantidad de imágenes necesarias para el entrenamiento de clasificadores. El reconocimiento en las imágenes que se crean en un vídeo es imposible de realizar manualmente por la gran cantidad de tiempo empleado en hacerlo. Estas técnicas permiten un reconocimiento al instante. Para este entrenamiento se han utilizado las librerías de OpenCV, que una vez más, demuestran todo el potencial que poseen en explotación de imágenes y vídeos.

En este tipo de proyectos muy importante controlar las versiones del OpenCV, ya que existen diferentes versiones en las herramientas y no son compatibles unas con otras debido al lenguaje utilizado. Se han entrenado los clasificadores con una muestra de imágenes pequeña. Se mejoraría introduciéndole más imágenes porque el programa aprende a medida que se van incluyendo nuevas. Para que la detección mejore se propone someter al objeto a diferentes situaciones de iluminación y realizar las fotografías desde diferentes ángulos.

La valoración económica realizada se ha enfocado desde un punto de vista de un trabajador autónomo, en ella se incluyen la mano de obra, los costes, los materiales necesarios y el IVA. Se ha desglosado el presupuesto en tres partes que incluyen el montaje de los proyectos, no su explotación. Se ha fijado también como objetivo simplificar al máximo la obtención de presupuestos, de forma que sea rápido y claro.

Tras analizar los resultados obtenidos de todo el trabajo, se puede llegar a la conclusión de que el uso de la programación se puede aplicar perfectamente al ámbito de la geomática y la topografía y que muestra amplias posibilidades de aplicación que aún no se ha desarrollado. La complejidad del trabajo ha sido mayor a lo que se esperaba en un principio pero satisface y supera totalmente las expectativas personales establecidas en el inicio.

Seguidamente, se resaltan las principales aportaciones realizadas teniendo en cuenta el recorrido llevado a cabo en este trabajo.

En el experimento didáctico, se ha trabajado con la tecnología RGB-D y se han obtenido nubes de puntos de una calidad buena dentro de las limitaciones que la cámara utilizada presenta en cuanto a distancia al objeto. Se ha implantado la caja de arena topográfica desarrollada por la UC Davis para su futuro uso en ferias o clases, y una guía de instalación y calibración paso a paso para su utilización con una serie de recomendaciones y soluciones a problemas frecuentes.

Se han efectuado cálculos complejos con una cantidad de datos masiva y se ha minimizado el tiempo de cálculo en el caso de los experimentos geofísicos. Además, es una herramienta eficaz que modela el geopotencial terrestre.

En el caso de la detección de matrículas, se le ha dado un enfoque de solución a un problema presente en la actualidad y se ha detallado cada paso para el entrenamiento del clasificador.

Se considera este trabajo como una aportación a las diversas líneas de trabajo fuera de lo habitual, y se ha planteado cada paso dando un enfoque diferente a lo convencional.

8. Líneas futuras de trabajo

Son infinitas las líneas de trabajo que combinan técnicas geomáticas con los últimos desarrollos tecnológicos en los campos de la realidad aumentada e inteligencia artificial, que van en la línea de este trabajo de fin de grado. Por citar algunas de ellas:

El uso de la fotografía y de las cámaras RGB-D en la creación de modelos 3D es un camino muy amplio de desarrollo. Sería interesante aplicar el uso de estas en ámbitos como la documentación del patrimonio, el control geométrico, la calidad o el cálculo de volúmenes. Para ello es necesario conocer los parámetros que intervienen en este proceso además de las tecnologías que ofrecen estos sensores, para así elegir el algoritmo de procesado correcto.

Se considera atractivo el desarrollo de la detección de objetos o personas en una nube de puntos 3D y profundizar en las librerías PCL.

También se considera interesante ampliar y depurar las herramientas desarrolladas con el objetivo de explotar estas mismas.

Sobre todo, para finalizar, hay que destacar la potencialidad de unir ambas técnicas. Siguiendo este camino se abren líneas de trabajo infinitas. Campos como el tratamiento de imágenes, las cuales se pueden manipular y analizar en grandes cantidades a partir de las brindadas por los satélites o los drones; o el tratamiento de datos masivos, puesto que cada vez hay más información a explotar. El Big Data es una oportunidad de crecimiento indudable en cualquier tipo de empresa y sector, los datos explotados de forma innovadora y creativa permiten crear cualquier aplicación de inteligencia artificial que recibirá y manejará datos en tiempo real. Este TFG solo es una semilla de todo el desarrollo que sufrirán estas áreas aun por explorar.

9. Bibliografía

- Asín, A. M. (Junio de 1959). Análisis estadístico de las discordancias en una red de nivelación de alta precisión. *Trabajos de estadística*, 10(2), 113-152.
doi:10.1007/BF03003987
- Bernhard Hofmann-Wellenhof, H. M. (1967). *Physical Geodesy*. Springer.
- Bernhardt, S. N. (2016). Automatic localization of endoscope in intraoperative CT image: A simple approach to augmented reality guidance in laparoscopic surgery. *Medical Image Analysis* 30, 130-143. doi:10.1016/j.media.2016.01.008
- Blázquez, J. T. (2012). *Modelado 3D de objetos a partir de la Kinect*. Madrid: Universidad Carlos III de Madrid.
- Carmody, T. (11 de 03 de 2010). *Wired*. Obtenido de How motion detection works in Xbox Kinect: <https://www.wired.com/2010/11/tonights-release-xbox-kinect-how-does-it-work/>
- Devesa, J. P., Martín Furones, Á., & García García, F. (2001). *Gravimetría*. Valencia: Universidad Politécnica de Valencia.
- Elchaoui Elghor, H. D.-H. (2016). D plane-based maps simplification for RGB-D slam systems. *Journal of Theoretical and Applied Information Technology* 93, 402-411.
- Estrada, L. A. (2012). *Prospección Gravimétrica*. Tucuman: Facultad de ciencias exactas y tecnología. Departamento de geodesia y topografía.
- Furones, A. M. (2000). *Análisis y ajuste de modelos de geoide. Observación y cálculo de la red gravimétrica de tercer orden en la Provincia de Valencia*. Valencia: Universidad Politécnica de Valencia.
- Gîrbacia, F. B. (2017). Planning of needle insertion for robotic-assisted prostate biopsy in augmented reality using RGB-D camera. *25th International Conference on Robotics in Alpe-Adria-Danube Region, RAAD 2016* (págs. 515-522). Belgrade, Serbia: Advances in Intelligent Systems and Computing. doi:10.1007/978-3-319-49058-8_56
- Imprimiendo el futuro. (23 de 09 de 2013). *Imprimiendo el futuro*. Obtenido de Blog venezolano sobre impresión 3D: <https://imprimiendoelfuturo.wordpress.com/2013/09/24/convierte-la-kinect-de-tu-xbox-en-un-escaner-3d/>
- Kreylos, O. (11 de 03 de 2015). *Augmented Reality Sandbox*. Obtenido de <https://arsandbox.ucdavis.edu/forums/topic/complete-installation-instructions/>

- Magallón, M. S. (2013). *Sistema interactivo para manejo de electrodomésticos en entornos domésticos*. Zaragoza: Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza.
- ManCTL, O. (2013). *Skanect*. Obtenido de 3D Scanning:
<http://skanect.occipital.com/contact/>
- Marín, C. (2015). *Tutor de programación*. Obtenido de Entrenar OpenCV en Detección de Objetos: <http://acodigo.blogspot.com.es/2015/12/entrenar-opencv-en-deteccion-de-objetos.html>
- McCormack, J. P. (2015). *2D LIDAR as a distributed interaction tool for virtual and augmented reality video games*. Toronto, Canada: 2015 IEEE Games Entertainment Media Conference, GEM 2015. doi:10.1109/GEM.2015.7377221
- OpenKinect. (2012). *OpenKinect*. Retrieved from About:
https://openkinect.org/wiki/Main_Page
- Paukstadt, O. (2016). *ManPag.es - linux manpages*. Obtenido de Fakenect Record:
<http://manpag.es/debian-wheezy/1+fakenect-record>
- PROFACTOR. (2016). *ReconstructMe*. Obtenido de <http://reconstructme.net/>
- Pulido, L. F. (2011). *Reconocimiento de patrones*. Obtenido de Aicitel:
https://aicitel.files.wordpress.com/2011/08/rp_clase1.pdf
- Ramirez, F. (2013). *Tuelectrónica*. Obtenido de Detección de rostro con OpenCV:
<http://www.tuelectronica.es/tutoriales/programacion/deteccion-de-rostro-con-opencv.html>
- Rodríguez, M. D. (2012). *Extracción automática de caras en imágenes captadas con móviles Android*. Barcelona: Universitat Politècnica de Catalunya.
- Serrano, L. P. (2011). *Proyecto Fin de Carrera. Robótica, software y telecomunicaciones*. Obtenido de <https://landerpfc.wordpress.com/2011/02/18/kinect-freenect-openkinect-openn/>
- Serrano, L. P. (2011). *Virtual Blackboard: colour and human gestures motion tracking*. Cáceres: Universidad Politécnica de Cáceres.
- Tatić, D. &. (2017). The application of augmented reality technologies for the improvement of occupational safety in an industrial environment. *Computers in Industry* 85, 1-10. doi:10.1016/j.compind.2016.11.004
- Torra, V. (2011). Instituto de Investigación en Inteligencia Artificial (CSIC). *Lychnos*(7).

Universidad de Las Palmas de Gran Canaria. (10 de 2015). *Universidad de Las Palmas de Gran Canaria*. Obtenido de <http://www2.ulpgc.es/hege/almacen/download/16/16817/tema6.pdf>

Universidad politécnica de Valencia. (10 de 2015). *Geodesia*. Obtenido de <http://www.upv.es/unigeo/index/docencia/etsigct/geodesia/teoria/TEMA4.pdf>

Wilson, A. (s.f.). *Achu´s TechBlog*. Obtenido de <https://achuwilson.wordpress.com/>

Wolk, K. W. (2017). Implementing statistical machine translation into mobile augmented reality systems. *10th International Conference on Multimedia and Network Information Systems, MISSI 2016* (págs. 61-73). Wrocław, Poland: Advances in Intelligent Systems and Computing. doi:10.1007/978-3-319-43982-2_6

10. Anexos

ANEXO 1. Uso de la Kinect como Laser Scanner en Windows

Existe la posibilidad de usar las cámaras RGBD como láser ya que cuentan con la información de profundidad. Previamente se había conseguido con libfreenect pero se había encontrado el problema de no poder exportar las nubes de puntos 3D, por lo que no resultaba del todo útil.

Leyendo blogs sobre el tema como Imprimiendo el futuro, un blog venezolano que se dedica a la explotación de modelos 3D (Imprimiendo el futuro, 2013), se descubrió una serie de herramientas para crear nubes de puntos, y lo que es más importante, descargarlas desde la Kinect.

Para ello se ha cambiado al sistema operativo Windows ya que los programas estaban desarrollados solo para este.

Para trabajar con la Kinect en Windows es necesario descargar los SDK de la página web de Microsoft. Se debe descargar una versión y otra dependiendo del modelo de la cámara que se tenga, en este caso se ha descargado la versión 1.8, ya que la cámara es la Kinect Xbox 360 modelo 1414. Estos drivers son totalmente compatibles con Windows 10.

La instalación de los mismos se debe realizar sin la Kinect conectada al ordenador, si no pueden surgir problemas.

En caso de no poder instalar los drivers o que no funcionen, existe otra alternativa que sería la instalación de OpenNI, que es una plataforma de código abierto mencionada en otros apartados previos.

Una vez instalados los drivers, se debe instalar uno de los siguientes programas:

- ReconstructMe: es un software que ofrece un sistema de escaneo 3D en tiempo real. Su uso es similar al de una cámara de video ordinaria. Se debe mover la cámara alrededor del objeto a ser modelado. Permite exportar a OBJ, PLY, OFF y STL. (PROFACTOR, 2016)
- Skanect: digitaliza en 3D a todo color cualquier objeto, persona o habitación. Permite exportar a OBJ, PLY, STL y VRML. (ManCTL, 2013)

Para el uso de ambos software se recomienda un ordenador con buena RAM para que sea capaz de procesar bien y rápido las nubes de puntos.

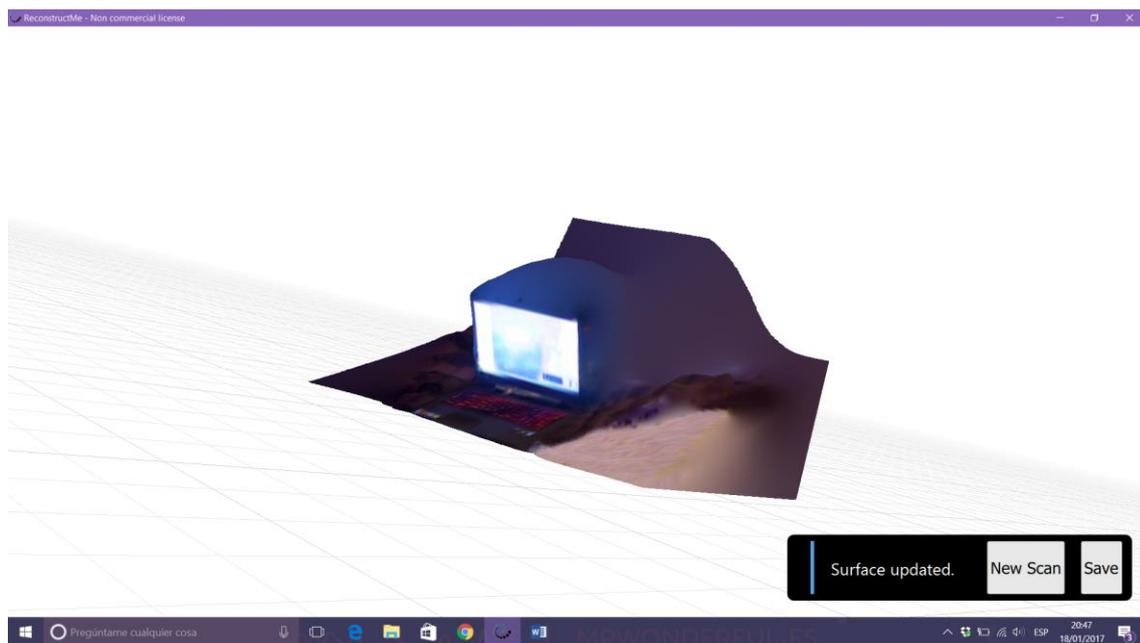


Ilustración 7 Objeto escaneado con ReconstructMe

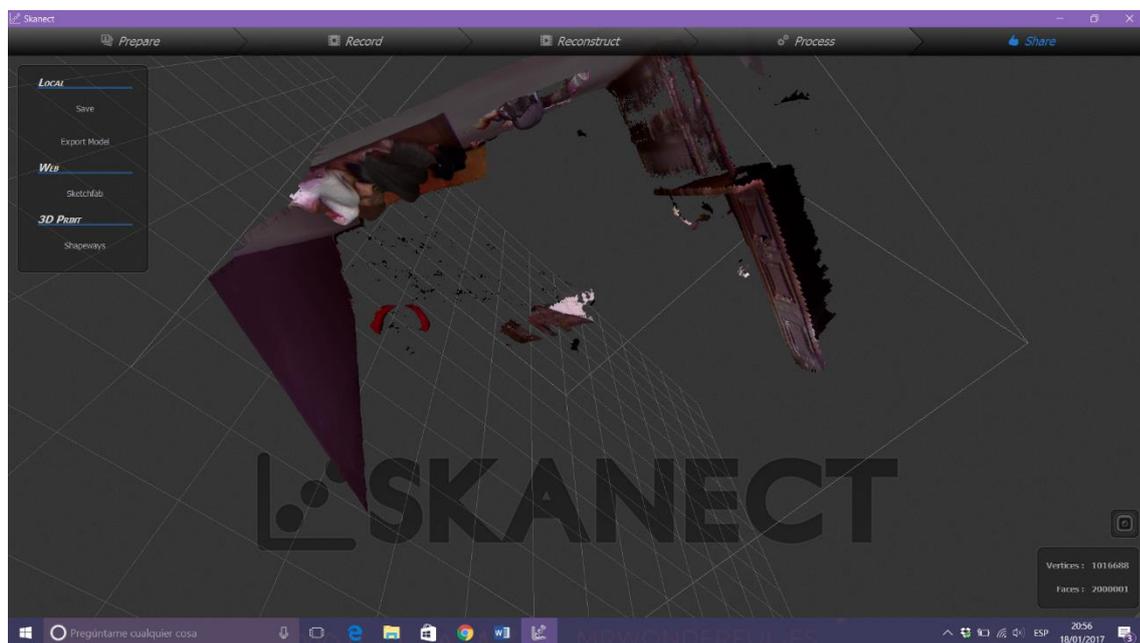


Ilustración 8 Habitación escaneada con Skanect

Skanect es menos sensible a los movimientos y se pierde menos a la hora de tomar el video. Lo malo es que al ser las dos versiones de prueba tienen sus limitaciones, como no poder descargar archivos mayores a un número determinado de caras del objeto.

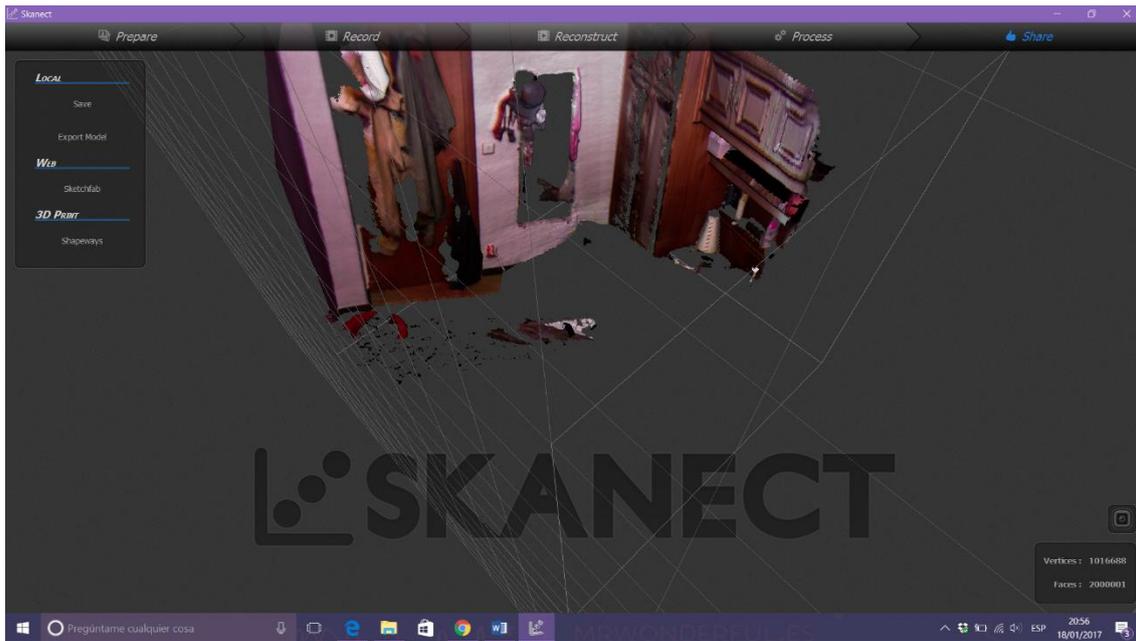


Ilustración 9 Pared escaneada con Skanect

Existen SDK de los mismos fabricantes para desarrollar el programa o utilizar sus módulos de manera libre para crear un software igual en lenguaje C.

ANEXO 2. La cámara Kinect 360

Kinect es un dispositivo de juego (para Xbox 360) mediante el cual puedes jugar con todo el cuerpo, es decir, sin mandos. Kinect también reconoce gestos, comandos de voz, objetos e imágenes.

Hace uso de dos cámaras frontales, una convencional RGB y un sensor de distancia, y una serie de micrófonos. El sistema de percepción de profundidad consta de tres partes básicas: el proyector láser de infrarrojos, el sensor CMOS y el microchip que procesa la información. Fue creado y desarrollado por PrimeSense. Su funcionamiento se basa en la proyección de un patrón de puntos pseudo-aleatorio y su lectura y triangulación se realiza mediante el sensor CMOS. (Magallón, 2013)

Las partes de las que se compone son las siguientes:

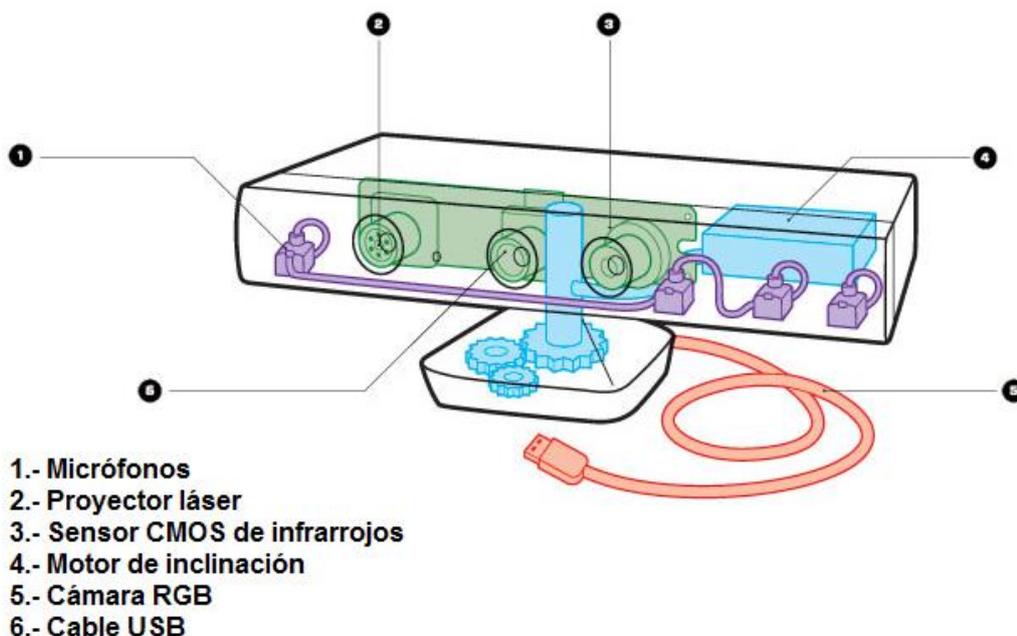


Ilustración 10 Partes de la cámara Kinect (Magallón, 2013)

Está compuesta por cuatro micrófonos que permiten la comunicación con el dispositivo.

Posee un proyector laser cuya longitud de onda es de 830nm y consta de un sistema de difracción que subdivide el rayo en múltiples instancias, proyectando un patrón de puntos pseudo-aleatorio como se ha comentado previamente.

El sensor infrarrojo, pese a que Microsoft no ha desvelado de qué componente específico se trata, han determinado expertos que se trata del sensor CMOS de la marca Micron. Este posee un tamaño de pixel de 5,2 μ m, un formato de video 5:4 con resolución SXGA 1280 x 1024 (1,3 megapíxeles), 30 fps, posee un filtro de paso de infrarrojos y se sitúa en el rango de captación del espectro electromagnético del infrarrojo cercano ya que mide 830nm.

Respecto a su campo de visión, el horizontal de la cámara es 57° y el rango máximo para la captación de profundidad es de 0,8 a 4 metros. En cuanto al vertical es de 43,5°.

La cámara cuenta con un motor de inclinación, que la inclina arriba o abajo 27° para poder abarcar el cuerpo de los usuarios independientemente de la altura a la que se encuentre el aparato.

En cuanto a la cámara RGB, tampoco ha sido revelado el modelo del sensor, pero se ha descubierto que es un Micron MT9M112 que posee las siguientes características: un tamaño de pixel de 2,8µm, 1,3 Megapíxeles, una resolución de 1280 X 1024, filtro de color Bayer y 15 fps a máxima resolución.

La conexión de la Kinect tanto a la consola como al ordenador se realiza a través de USB 2.0.

No hay duda de que una de las prestaciones más interesantes que ofrece esta cámara es la percepción de profundidad. Funciona emitiendo un haz de luz invisible ya que es infrarroja y puede conocer el tiempo que tarda la luz al volver al sensor tras reflejarse en los objetos. El sensor actúa como un sonar, conociendo el tiempo de cada salida y llegada de la luz tras reflejarse en un objeto, sabiendo la velocidad absoluta de la luz, se puede tener la distancia a la cual se encuentra ese objeto. Este método hace que la Kinect sea capaz de distinguir la profundidad de cada objeto con diferencias de un centímetro, y su altura y anchura con diferencias de 3 milímetros. (Carmody, 2010)

ANEXO 3. Acceso y grabación de los datos que ofrece Kinect.

Para acceder a los datos de la Kinect, es necesario comunicarse con un componente llamado KinectComp, el cuál dispone de una interfaz desde la cual se puede conseguir los datos de imagen y profundidad para poder tranajar con ellos.

Este componente hace de intermediario entre la Kinect y los demás componentes que se dispongan, permitiendo así controlar el sensor.

El driver que permite realizar esto se está desarrollando desde OpenKinect, que es una comunidad abierta de personas que aportan código al proyecto de hacer uso del hardware de Xbox Kinect con los ordenadores u otros dispositivos. (OpenKinect, 2012)

El driver se llama Freenect, y hasta el momento es capaz de capturar la imagen de la cámara con resolución VGA (640x480), así como los datos de profundidad, y controlar el motor en su base.

Por otro lado está PrimeSense, los diseñadores del chip PS1080 que lleva incorporado la Kinect. PrimeSense junto con WillowGarage (desarrolladores de OpenCV), están creando unas librerías llamadas OpenNI (Open Natural Interaction), y se encargan de facilitar el tracking de personas, control por gestos, etc. (Serrano L. P., 2011)

Si se quiere utilizar KinectComp se debe tener conectada la Kinect e instalar una versión del driver adecuada. Freenect está en constante cambio, por eso se recomienda instalar la última versión desde un repositorio, en este caso GitHub, para que no genere problemas en el ordenador como puede ser el consumo excesivo de RAM.

```
sudo apt-get install git-core cmake freeglut3-dev pkg-config build-essential libxmu-dev  
libxi-dev libusb-1.0-0-dev  
git clone git://github.com/OpenKinect/libfreenect.git  
cd libfreenect  
mkdir build  
cd build  
cmake ..  
make  
sudo make install  
sudo ldconfig /usr/local/lib64/
```

Una vez instalado el driver, se podrá acceder a ver los datos desde la demo que trae Freenect llamado glview.

```
sudo freenect-glview
```

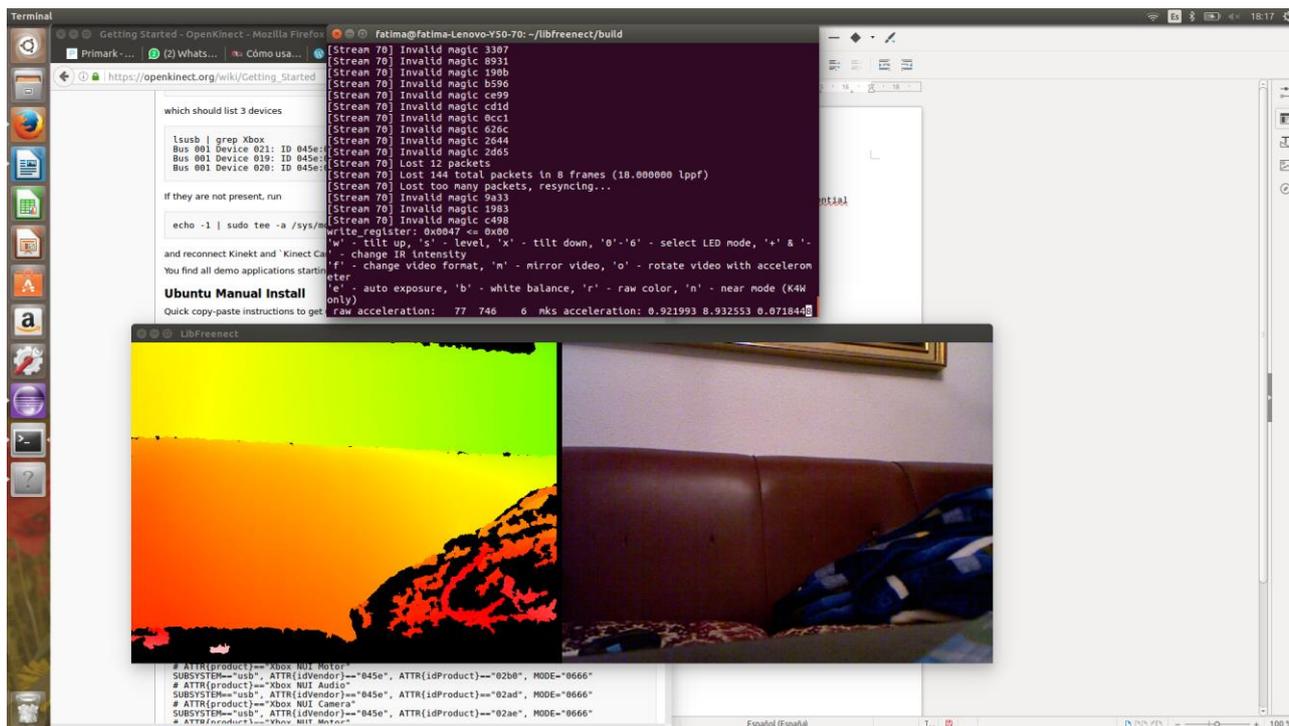


Ilustración 11 Datos obtenidos a partir de Freenect

Una vez que se ha accedido a estos datos, se va a proceder a guardar las imágenes RGB y los datos de profundidad usando la herramienta Fakenect. Como se ha comentado previamente, Freenect consigue capturar el audio de sus micrófonos, los datos de las cámaras y cambiar el estado del LED frontal. Se está trabajando también en un sistema llamado Fakenect, con el cual se puede grabar una sesión con la Kinect.

Este sistema funciona ejecutándolo desde la terminal introduciendo unos parámetros que dependen del tipo de datos que se desee almacenar. El comando sería el siguiente:

```
fakenect-record [-h] [-ffmpeg] [-ffmpeg-opts options] outputdir
```

Este componente graba en el directorio de salida introducido los datos de aceleración, profundidad y los datos RGB en ficheros individuales con un nombre con las estructura “tipo - marca de tiempo – hora actual”, donde:

Tipo: es aceleración (a), profundidad (d) o RGB (r).

Marca de tiempo: es el tiempo asociado a la observación.

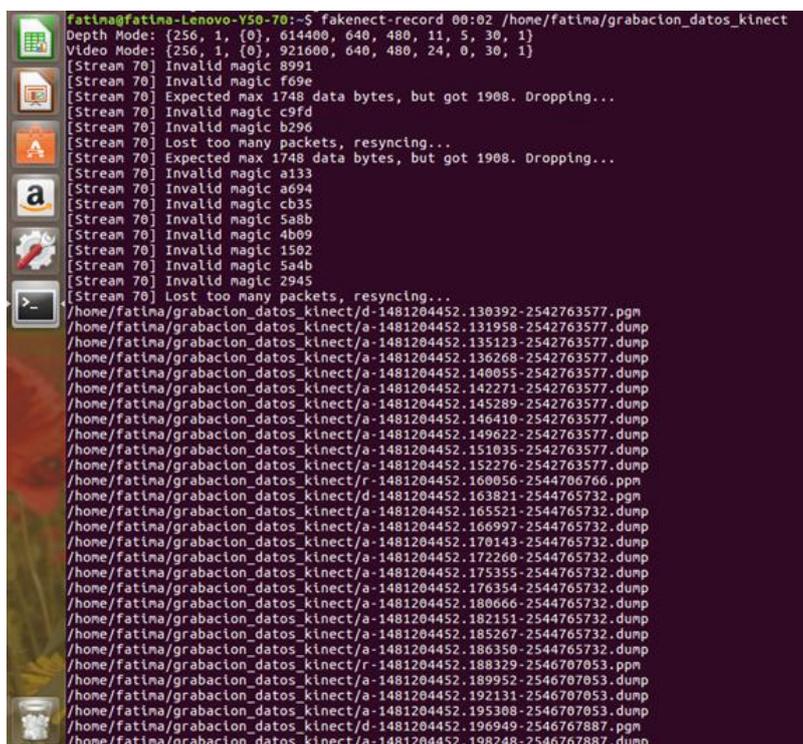
Hora actual: versión del tiempo en segundos.

Se guarda la hora actual para que los retrasos puedan ser recreados exactamente cómo ocurrieron. Los datos de RGB y profundidad se vuelcan en ficheros PPM y PGM respectivamente. Además se incluye un archivo llamado INDEX.txt con todos los nombres de los archivos locales de ese directorio.

Una vez iniciado, salvo que no se indique el tiempo de grabación, el programa continuará adquiriendo datos de la Kinect. Cuando se quiera detener, se deberá presionar Ctrl-C y se parará el bucle. (Paukstadt, 2016)

Las opciones que ofrece el comando son las siguientes:

- ffmpeg: manda el video a ffmpeg.
- ffmpeg-opts: son las opciones de los formatos de salida de los archivos que se pueden consultar en la ayuda.
- h: comando de ayuda



```

fatima@fatima-Lenovo-Y50-70:~$ fakenect-record 00:02 /home/fatima/grabacion_datos_kinect
Depth Mode: {256, 1, {0}, 614400, 640, 480, 11, 5, 30, 1}
Video Mode: {256, 1, {0}, 921600, 640, 480, 24, 0, 30, 1}
[Stream 70] Invalid magic 8991
[Stream 70] Invalid magic f69e
[Stream 70] Expected max 1748 data bytes, but got 1908. Dropping...
[Stream 70] Invalid magic c9fd
[Stream 70] Invalid magic b296
[Stream 70] Lost too many packets, resyncing...
[Stream 70] Expected max 1748 data bytes, but got 1908. Dropping...
[Stream 70] Invalid magic a133
[Stream 70] Invalid magic a694
[Stream 70] Invalid magic cb35
[Stream 70] Invalid magic 5a8b
[Stream 70] Invalid magic 4b09
[Stream 70] Invalid magic 1502
[Stream 70] Invalid magic 5a4b
[Stream 70] Invalid magic 2945
[Stream 70] Lost too many packets, resyncing...
/home/fatima/grabacion_datos_kinect/d-1481204452.130392-2542763577.pgm
/home/fatima/grabacion_datos_kinect/a-1481204452.131958-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.135123-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.136268-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.140055-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.142271-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.145289-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.146410-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.149622-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.151035-2542763577.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.152276-2542763577.dump
/home/fatima/grabacion_datos_kinect/r-1481204452.160056-2544706766.ppm
/home/fatima/grabacion_datos_kinect/d-1481204452.163821-2544765732.pgm
/home/fatima/grabacion_datos_kinect/a-1481204452.165521-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.166997-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.170143-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.172260-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.175355-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.176354-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.180666-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.182151-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.185267-2544765732.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.186350-2544765732.dump
/home/fatima/grabacion_datos_kinect/r-1481204452.188329-2546707053.ppm
/home/fatima/grabacion_datos_kinect/a-1481204452.189952-2546707053.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.192131-2546707053.dump
/home/fatima/grabacion_datos_kinect/a-1481204452.195308-2546707053.dump
/home/fatima/grabacion_datos_kinect/d-1481204452.196949-2546767887.pgm
/home/fatima/grabacion_datos_kinect/a-1481204452.198248-2546767887.dump
  
```

Ilustración 12 Ejecución Fakenect

Si la conexión a la Kinect da error, se debe realizar lo siguiente. Se ejecuta el siguiente comando para usar la Kinect sin ser root:

```
sudo adduser $USER video
```

Y se crean una serie de normas para Linux:

```
sudo nano /etc/udev/rules.d/51-kinect.rules
```

```
# ATTR{product}=="Xbox NUI Motor"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02b0", MODE="0666"  
# ATTR{product}=="Xbox NUI Audio"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ad", MODE="0666"  
# ATTR{product}=="Xbox NUI Camera"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ae", MODE="0666"  
# ATTR{product}=="Xbox NUI Motor"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02c2", MODE="0666"  
# ATTR{product}=="Xbox NUI Motor"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02be", MODE="0666"  
# ATTR{product}=="Xbox NUI Motor"  
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02bf", MODE="0666"
```

De esta manera se autorizan las conexiones al sensor. (OpenKinect, 2012)

ANEXO 4. Nubes de puntos con Kinect y librería PCL

Una de las opciones que ofrece la Kinect es crear nubes de puntos 3D gracias a la cámara de profundidad y la cámara RGB. Se pueden acceder a estas con la librería libfreenect ya mencionada anteriormente, la herramienta se llama `cpp_pcvview`.

El resultado es una nube de puntos no muy densa, pero es interesante el abanico de posibilidades que esta ofrece.

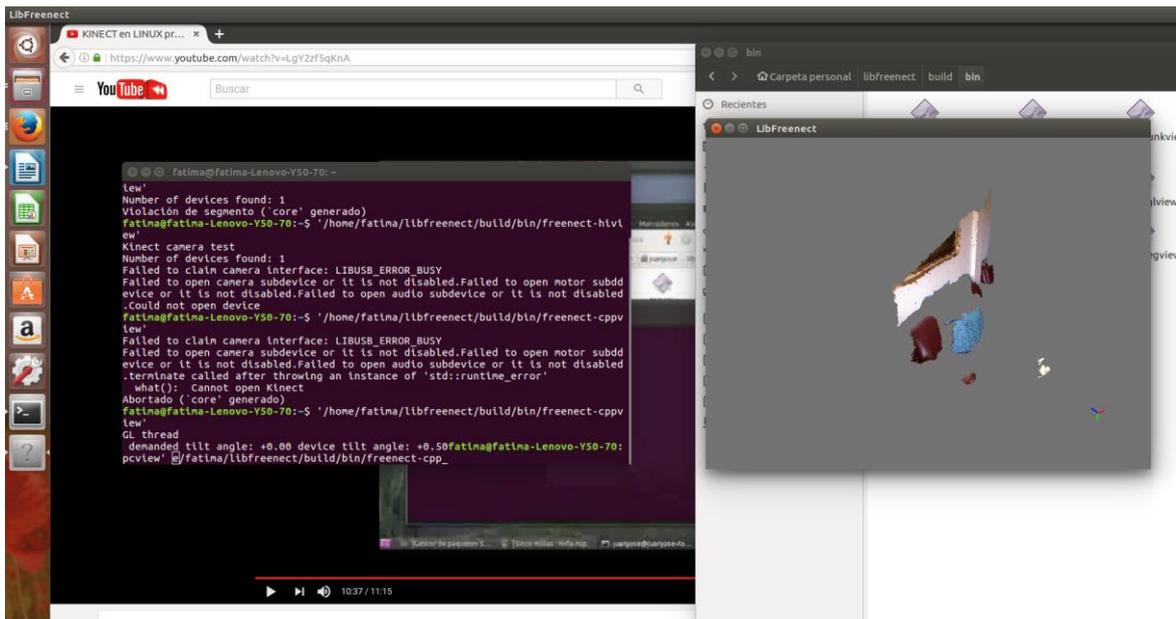


Ilustración 13 `cpp_pcvview`

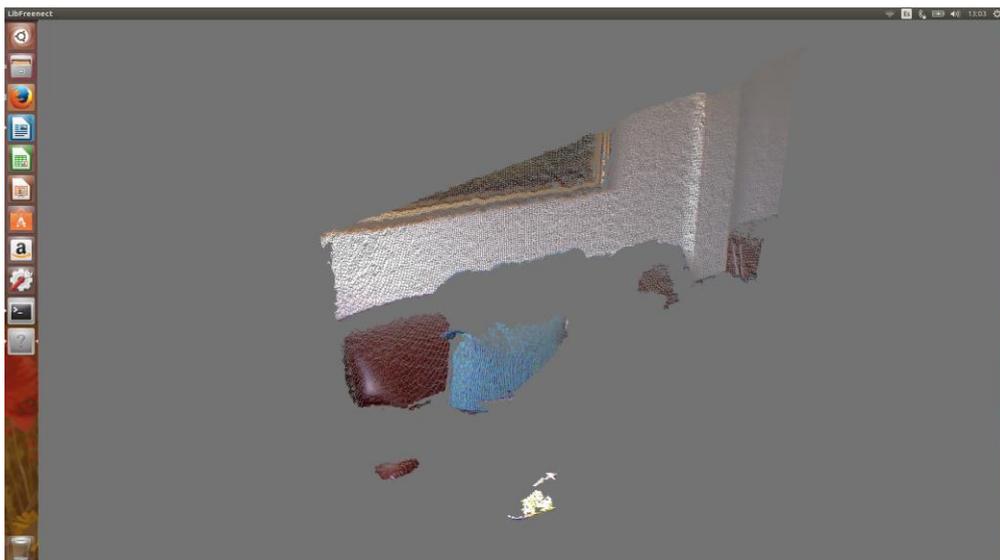


Ilustración 14 Nube de puntos generada

Para tratar las nubes de puntos existen unas librerías llamadas PCL (Point Cloud Library), descubiertas gracias al trabajo de fin de grado de Juan Toribios Blázquez, el cual desarrolla el modelado 3D de objetos a partir de la cámara Kinect.

Para entender un poco mejor que son estas librerías y como se vinculan a la Kinect se dará una breve explicación.

PCL es un proyecto que inició Willow Garage en marzo de 2011 para el procesamiento de nubes de puntos en 3D. El proyecto ha continuado en todo el mundo con la colaboración de científicos, empresas y universidades gracias a que el software es un código abierto y gratuito. PCL es multiplataforma, y ha sido compilado con éxito y desplegado en Linux, MacOS, Windows y Android / iOS. (Blázquez, 2012)



Ilustración 15 Logo PCL

Las librerías que componen PCL están preparadas para trabajar con nubes de puntos. Una nube de puntos es una estructura de datos en 4D en la que encontramos cuatro datos, las coordenadas X Y Z de cada punto y el color en formato RGB, esta nube de puntos se puede adquirir a partir de sensores como cámaras 3D o láser, en el mercado encontramos hardware como PrimeSensor 3D, XTionPRO Asus y el Kinect de Microsoft que ocupa lugar en este proyecto.

Los archivos que contendrán las nubes de puntos son del formato .PCD, un formato original de PCL basado en formatos como .PLY, .STL, .OBJ o .X3D. Cada archivo .PCD incluye una cabecera con información:

- Versión
- Nombre de cada dimensión (x, y, z, rgb, normal_x, normal_y, etc.)
- Tamaño (4bytes)
- Tipo (F, float)
- Número de elementos que ocupa cada dimensión (normalmente 1)
- Ancho
- Alto (si es 1 los puntos están desordenados y el ancho es el total de la nube)
- La orientación en la que se han adquirido los datos
- Número total de puntos.
- Tipo de código usado en el archivo (ASCII o binario).

ANEXO 5. Instalación completa

Para la instalación del software necesario para este proyecto, es necesario seguir los siguientes pasos. Se deben instalar tres software: Vrui, Kinect y Sandbox, en este orden en el que aparecen. (Kreylos, 2015)

Se debe partir de un ordenador que contenga Linux instalado, debe de ser un ordenador "real", en una máquina virtual no funciona. En este caso se parte de Ubuntu 16.04.1 LTS. Se instalarán todos los drivers de la tarjeta gráfica del ordenador, se instalarán los de Nvidia porque como previamente se ha mencionado el ordenador posee una tarjeta gráfica NVIDIA® GTX-860M.

Una vez que se tenga el ordenador preparado se empezará con la instalación de los tres software necesarios. El primero será Vrui, para instalarlo se introducirá en consola el siguiente comando:

```
cd ~  
wget http://idav.ucdavis.edu/~okreylos/ResDev/Vrui/Build-Ubuntu.sh  
bash Build-Ubuntu.sh  
build
```

El siguiente paso será instalar el paquete de video Kinect 3D. En la terminal introducir los siguientes comandos:

```
cd ~/src  
wget http://idav.ucdavis.edu/~okreylos/ResDev/Kinect/Kinect-3.2.tar.gz  
tar xzf Kinect-3.2.tar.gz  
cd Kinect-3.2  
make  
sudo make install  
sudo make installudevrules  
ls /usr/local/bin
```

Se debe comprobar si existen los ficheros KinectUtil y RawKinectViewer tras el último paso.

Por último, se instalará AR Sandbox.

```
cd ~/src  
wget http://idav.ucdavis.edu/~okreylos/ResDev/SARndbox/SARndbox-2.3.tar.gz  
tar xzf SARndbox-2.3.tar.gz  
cd SARndbox-2.3  
make  
ls ./bin
```

Se comprobará si se crean CalibrateProjector y SARndbox.

Una vez que se tienen todos los programas necesarios instalados, se deberán realizar las calibraciones de la Kinect y el proyector.

Se conecta la cámara Kinect (de primera generación) y se descargarán los parámetros intrínsecos de calibración.

```
sudo /usr/local/bin/KinectUtil getCalib 0
```

Utilizando la herramienta RawKinectViewer, se debe de alinear la Kinect con la caja de arena, de modo de que esta ocupe todo el campo de visión, dejando también los marcos de la misma. Para iniciar la herramienta desde la terminal ejecutar el siguiente comando:

```
cd ~/src/SARndbox-2.3  
RawKinectViewer -compress 0
```

A continuación, se debe calcular el plano base de la caja de arena que se posea. Es importante porque todos los colores y las elevaciones parten de esto. A parte, se tendrá que medir la posición de las esquinas de las cajas para la simulación de agua.

Para ello se utilizara también la herramienta RawKinectViewer, donde aparece en la izquierda la imagen de profundidad y en la derecha la imagen RGB. Como efectuar mediciones encima de la arena es complicado, se pondrá un cartón encima de la caja de arena para que el plano sea uniforme.

El programa RawKinectViewer funciona asignado funciones a las teclas. Es decir, al seleccionar por ejemplo el menú utilizando la tecla "1" la selección para ejecutarla se guardará en esa tecla.

Como se desea sacar las ecuaciones del plano, se seleccionará la opción "Extract Planes". Después de poder crear estas ecuaciones se necesitara crear un marco de profundidad promedio, ya que en las esquinas sobre todo se genera mucho ruido. Para eso se recogerán muchas imágenes de profundidad y se calculará la profundidad media.

Se seleccionará en el menú pulsando el botón derecho del ratón "Average Frames", tardará un par de segundos en crear la media y se notará una disminución notable del ruido en los bordes de la imagen de profundidad. Se quedará una imagen estática en la que dibujaremos un rectángulo para calcular la ecuación del plano.

Para ello se presiona la tecla en la que se ha guardado la opción de calcular las ecuaciones del plano y se realizara un rectángulo desde la esquina superior izquierda a la inferior derecha sin contar los bordes con ruido. Después de realizarlo el programa no da ningún feedback,

pero si se vuelve atrás y se observa la terminal desde la que se está ejecutando el programa, se verá la ecuación del plano en dos formatos.

En la primera línea, se muestra la ecuación del plano en el espacio de profundidad, aunque esta no se necesita. En cambio, en la segunda línea se muestra la ecuación en el espacio de la cámara RGB, que es la necesaria.

Ahora se procederá a realizar el segundo paso, que es medir las esquinas interiores de la caja para establecer los límites de la simulación de agua.

Primero, pulsando el botón derecho se desactivará “Average Frames” para tener imagen en vivo otra vez. Para este paso se podrá retirar el cartón y se deberá eliminar la acción grabada en tecla que se haya seleccionado. Se activará la opción “Measure 3D positions” y, como se ha realizado en el paso anterior, se capturará la imagen de profundidad activando “Average Frames”.

La introducción de las esquinas tiene un orden determinado; primero, pulsando la tecla, se seleccionara la esquina inferior izquierda. Como ha pasado antes, el programa muestra los resultados en la terminal, que será la posición en X, Y y Z de las esquina. La magnitud de las coordenadas es en centímetros.

A continuación se medirá la posición de la esquina inferior derecha, luego la esquina superior izquierda y por último la esquina superior derecha.

Seguidamente se podrá cerrar el programa presionando escape, y se procederá a introducir los números de las mediciones realizadas en el archivo BoxLayout.txt. Se abrirá este fichero con el editor de textos o se creará en caso de que no exista.

Lo primero que se deberá introducir es la ecuación del plano en el espacio de la cámara RGB, se copiarán solo los números obtenidos.

Se copiará solo el vector de tres valores que proporciona y se sustituirá el signo igual “=” por una coma. Teniendo así tres números, que definen la dirección de la normal del plano de la cámara, y el último término, que se refiere al desplazamiento.

Ahora se copiarán las posiciones de las cuatro esquinas en el mismo orden en el que han sido medidas. No hace falta modificar nada, se copiará lo que haya salido directamente por la terminal.

Antes de cerrar el nuevo txt generado, falta una cosa importante por hacer. La ecuación del plano define la superficie de elevación cero de la caja de arena, pero esta elevación se ha calculado respecto al plano creado por la caja de cartón que se ha puesto encima de la caja. Por tanto el nivel cero está más elevado que en la realidad. La solución a este problema es sencilla.

Como se ha dicho antes, el último término describe el desplazamiento del plano de cota cero, por tanto, lo único que se debe hacer es observar la Z de los corners y ver el grosor de la

capa de arena que hay en los mismos. Es decir, si el grosor de la arena son 5cm y la Z media de las esquinas es -99,7, bastaría con situar la cota cero 15cm por debajo de lo calculado. El nuevo valor de desplazamiento sería -105,975.

Con esta última modificación ya se tendría el .txt generado y se procedería a guardarlo en el directorio etc/SARndbox-<version>. Para ajustar los niveles de profundidad luego se puede modificar para que se acople mejor a la situación que se posea.

```
cd ~/src/SARndbox-2.3  
pluma etc/SARndbox-2.3/BoxLayout.txt &
```

El último paso es la calibración del proyector con lo capturado por la Kinect, para ello se lanzará la herramienta CalibrateProjector y se superpondrá el cartón por encima de la caja otra vez. Lo que se hará es medir puntos de enlace entre el proyector y la Kinect.

```
cd ~/src/SARndbox-2.3  
./bin/CalibrateProjector -s <width> <height>
```

Donde <ancho> <altura> son el ancho y la altura de la imagen del proyector en píxeles (p.e. 1024 768). Es muy importante pasar a pantalla completa con F11 antes de comenzar la calibración.

Se guardará la herramienta en dos teclas del teclado. Se presiona cualquier tecla para acceder al menú de selección, en él se seleccionara "Capture", el programa ordenará que se presione otro botón que permitirá capturar la imagen del fondo para la calibración. Se retirará el cartón de la superficie y se utilizará la tarjeta de calibración. Se ha hecho utilizando un CD viejo, superponiendo un papel encima con la misma forma y dibujando dos líneas perpendiculares que intersectan exactamente en el centro. Al CD se le ha pegado un alambre para poder manejarlo sin tener que tocarlo, ya que si no, el software no detecta el círculo bien porque aparece deformado por los dedos.

La idea es que las aspas proyectadas por el proyector intersecten con las dibujadas en el CD en forma de asterisco y la Kinect pueda captar el CD, de modo que se cree un punto de enlace entre ambos. Hay que capturar varios a distintas alturas, incluso haciendo un agujero en la arena, para que quede bien atado y pueda generar una calibración de calidad.

Una vez posicionado, se presionará el botón asignado anteriormente, se tirará un par de segundos haciendo la medición y cambiará de posición para realizar una nueva medición. Es importante que la luz proyectada encima del disco para hacer la medición sea verde, no amarilla. Si la luz es amarilla se debe a que el disco está fuera del rango de detección de la Kinect, por tanto el punto de enlace será invalido. El disco se debe mantener lo más ortogonal posible, los ángulos de inclinación empeoran la calibración.

Si al realizar el agujero en la arena no detecta el círculo del CD, se debe de capturar otra vez para que cuente con el cambio realizado en la superficie, pulsando el botón que se ha asignado primero.

La primera ronda de calibración consta de unos 9 puntos de enlace, después de esta ronda debe aparecer una cruz roja que intersecte con el CD en cualquier posición, ya que es capaz de detectarlo. Se moverá alrededor de la caja viendo que coincida en toda su extensión, y así se podrá verificar la calidad de la calibración realizada.

Si la calibración no tiene la precisión deseada, se deberá realizar una segunda ronda de captura de puntos de enlace siguiendo el procedimiento anterior.

En el caso de que la calibración sea correcta, se procederá a salir del programa presionando escape. A continuación se deberá de ejecutar SARndbox, que proyectará los colores de las elevaciones utilizando la calibración que se ha efectuado.

```
cd ~/src/SARndbox-2.3  
./bin/SARndbox -uhm -fpv
```

Para que los valores de calibración se apliquen hay que lanzar la aplicación en pantalla completa con F11.

ANEXO 6. Instalación OpenCV en Eclipse Neon

Al estudiar el trabajo realizado por UC Davis, se observa que la base de programación del software desarrollado por ellos es C/C++ en Linux. Es por eso que se ha tomado la decisión de que el sistema operativo sobre el que se trabajará es Ubuntu 16.04.1 LTS.

Ubuntu es un sistema operativo basado en GNU/Linux con licencia de uso gratuita que se centra en la facilidad de uso y en su rapidez.

Como software para programación en C/C++ se ha elegido Eclipse en su versión: Neon.1a Release (4.6.1). Eclipse es una plataforma de software que está diseñado para la programación y es de código abierto. Normalmente es usada para la programación en Java, pero existen plugins que permiten desarrollar código en otros lenguajes como Android, JavaScript o C/C++ entre otros. El interesante en este caso es el plugin Eclipse IDE for C/C++ Developers.

Para la primera toma de contacto de la programación con la realidad aumentada, se ha elegido la biblioteca OpenCV, que está enfocada a la visión artificial, desarrollada por Intel. OpenCV es multiplataforma, así que se trabajara con ella en Ubuntu para desarrollar pequeñas aplicaciones iniciales sobre reconocimiento de objetos, toma de imágenes o procesado de imágenes.

Para integrar OpenCV en Eclipse, se deben realizar una serie de procesos. Lo primero es descargarse el plugin de Eclipse para C/C++ ya que es un requisito que debe cumplirse en su instalación.

El segundo paso es la instalación de OpenCV, y para ello son necesarios diversos paquetes:

- GCC 4.4.x: es una serie de compiladores de software libre para diversos lenguajes de programación.
- Cmake 2.6: herramienta que permite automatizar el código en cualquier plataforma.
- Git: software de control de versiones de aplicaciones.
- GTK+2.x: conjunto de bibliotecas que sirven para desarrollar interfaces gráficas.
- pkg-config: software que provee una interfaz para llamar bibliotecas cuando se está compilando un programa.
- Python 2.6: lenguaje de programación interpretado.
- Numpy 1.5: paquete fundamental para operar con vectores y matrices multidimensionales con una biblioteca amplia de funciones matemáticas para Python.

- ffmpeg o libav: ffmpeg es una plataforma para grabar, convertir y hacer streaming de audio y video. Libav un software libre para procesar video y audio. Con poseer uno de los dos es suficiente.

- De manera opcional se recomienda instalar los paquetes de Ubuntu que se muestran a continuación: libtbb2, libtbb-dev, libdc1394 2.x, libjpeg-dev, libpng-dev, libtiff-dev, libjasper-dev, libdc1394-22-dev.

El siguiente paso es compilarlo e instalarlo desde la terminal. Para ello se emplearán los siguientes comandos:

```
sudo apt-get install build-essential
sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev
libswscale-dev
sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev
libtiff-dev libjasper-dev libdc1394-22-dev
```

En el caso de no querer instalar los paquetes opcionales de Ubuntu, la última línea de comandos no será necesaria.

A continuación se necesitará descargar el OpenCV utilizando para ello su repositorio. Se accederá desde la terminal utilizando el siguiente comando:

```
cd ~/</usr/local>
git clone https://github.com/opencv/opencv.git
```

Donde /usr/local será el directorio de trabajo en el cual se desee realizar la descarga.

Se procederá a compilarlo, para ello se utilizará el comando de Linux make. Se deberá crear un directorio temporal donde se quiera que almacene los makefiles generados, los binarios y los archivos de proyectos. Esto será a elección de usuario.

Se ejecuta el comando siguiente

```
cd ~/opencv
mkdir release
cd release
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local
```

Se accede al directorio creado y se ejecuta lo siguiente:

```
make
sudo make install
```

Y la instalación de OpenCV estaría completa y solo quedaría integrarlo en Eclipse.

Para ello se debe crear un nuevo proyecto C/C++ Project.

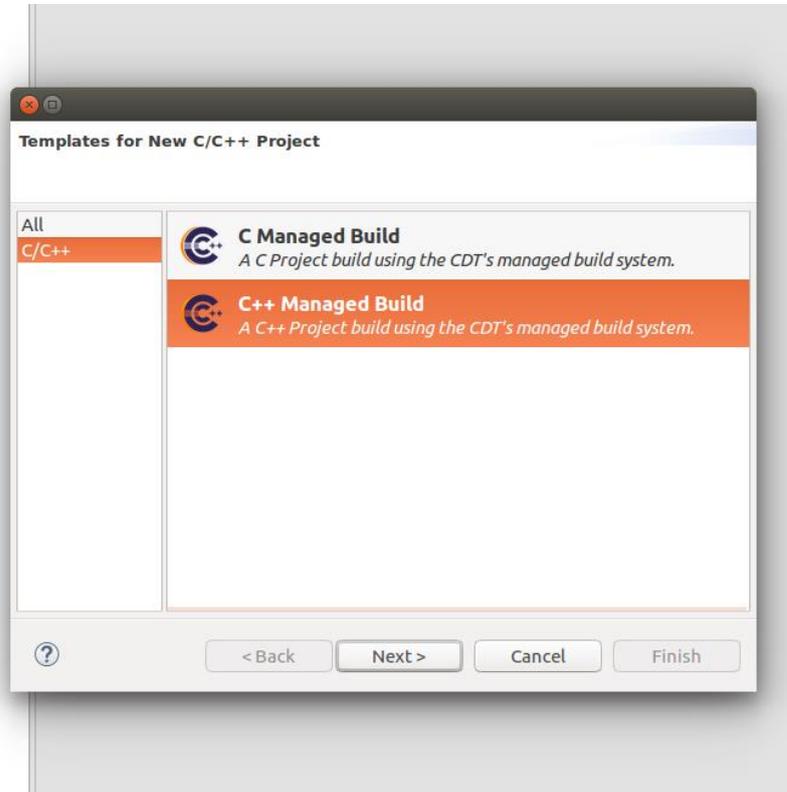


Ilustración 16 Crear nuevo proyecto C++ en Eclipse

Se selecciona un proyecto vacío y como compilador Linux GCC.

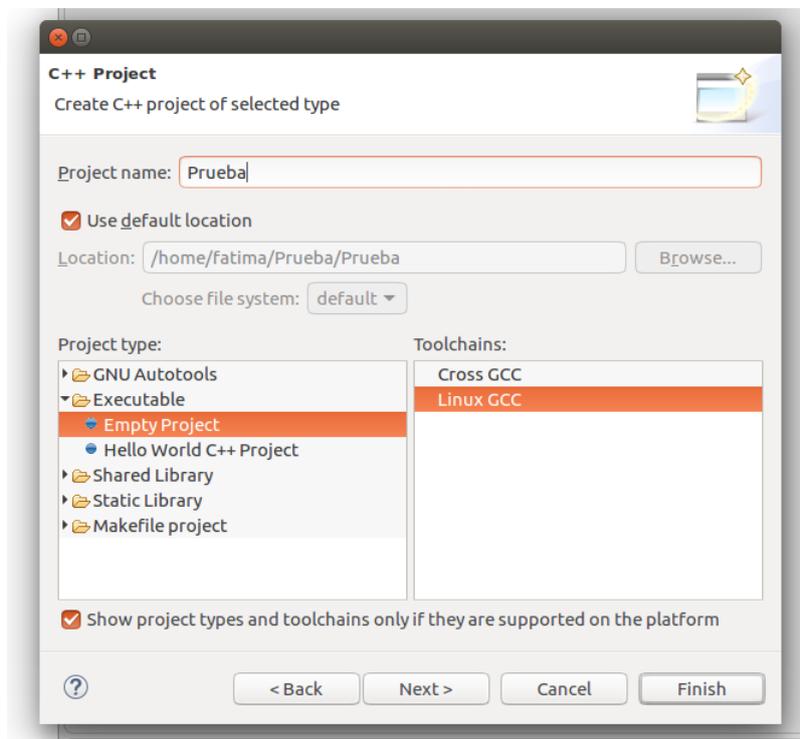


Ilustración 17 Elección de compilador

Ahora se debe configurar el proyecto antes de crear algún fichero de cabecera y programar ya que no reconocerá las librerías.

Los path metidos en la configuración variarán dependiendo de donde se haya realizado la instalación de OpenCV.

Se deben realizar cambios dentro de Propiedades del proyecto → C/C++ Build → Settings. Aquí se deberán cambiar dos cosas, los Includes de GCC C++ Compiler y las librerías de GCC C++ Linker.

En el apartado de Includes se deberá añadir el Path en el cual se encuentre la carpeta opencv. En este caso será /usr/local/include/opencv

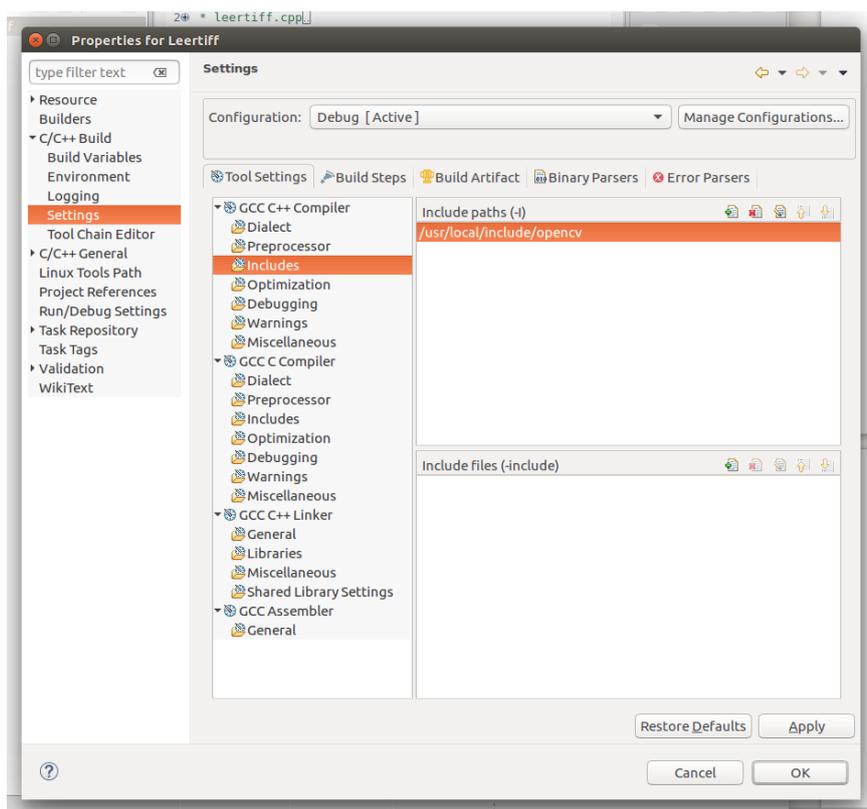


Ilustración 18 Configuración OpenCV (1)

Respecto a las bibliotecas, el path a introducir será la ubicación de la carpeta lib, en este caso /usr/local/lib. En Include files, se deberán introducir una a una las librerías que se van a utilizar, en este caso las introducidas son las siguientes (depende de la versión de OpenCV):

-lopencv_calib3d , -lopencv_core, -lopencv_features2d, -lopencv_flann, -lopencv_highgui, -lopencv_imgcodecs, -lopencv_imgproc, -lopencv_ml, -lopencv_objdetect, -lopencv_photo, -lopencv_shape, -lopencv_stitching, -lopencv_superres, -lopencv_videoio, -lopencv_videostab.

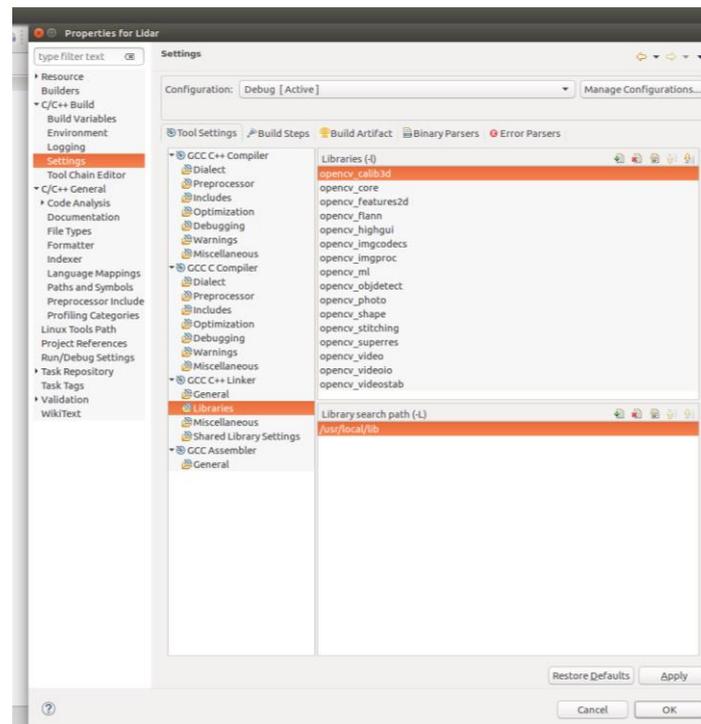


Ilustración 19 Configuración OpenCV (2)

Si no se sabe que ficheros son o se quiere comprobar de cuales dispone la versión que se posee, se puede averiguar escribiendo en la consola el siguiente comando:

```
pkg-config --libs opencv
```

Tras realizar esta configuración, se debe crear un archivo fuente y realizar el siguiente include para que se reconozcan los comandos de OpenCV.

```
#include <opencv2/opencv.hpp>
```

A partir de aquí se debe programar como cualquier otro programa de C/C++.

ANEXO 7. Conectar la cámara Web y realizar una fotografía

Con OpenCV se puede acceder a la cámara que posea el ordenador y grabar imágenes y vídeos.

Para acceder a la webcam se tiene que indicar el número de dispositivo que se desea usar a partir de la clase VideoCapture. Como el ordenador solo posee la webcam, se deberá poner cero.

Los resultados obtenidos con este programa son los siguientes:

- Muestra en una ventana la webCam, mientras que en la consola muestra las intrucciones de uso.

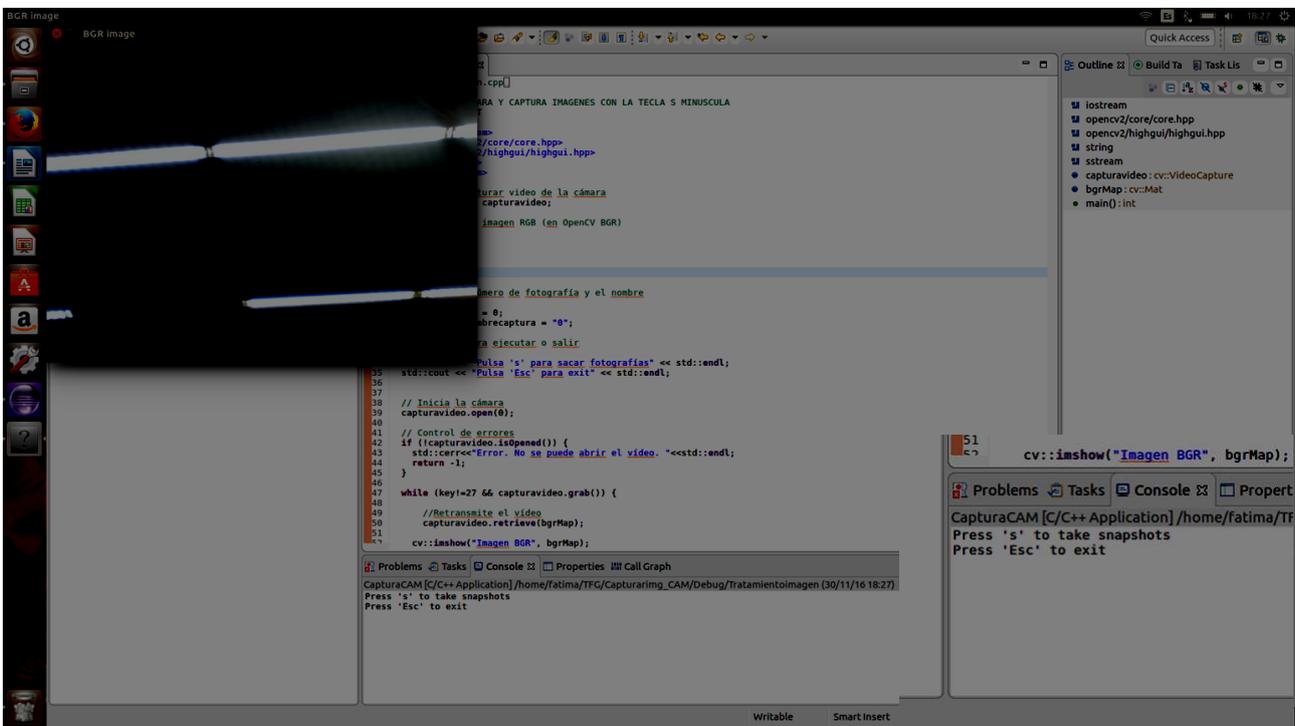


Ilustración 20 Abrir webcam con OpenCV y capturar una imagen

- Tras pulsar la tecla "s" almacena en la carpeta del proyecto la imagen en formato .png

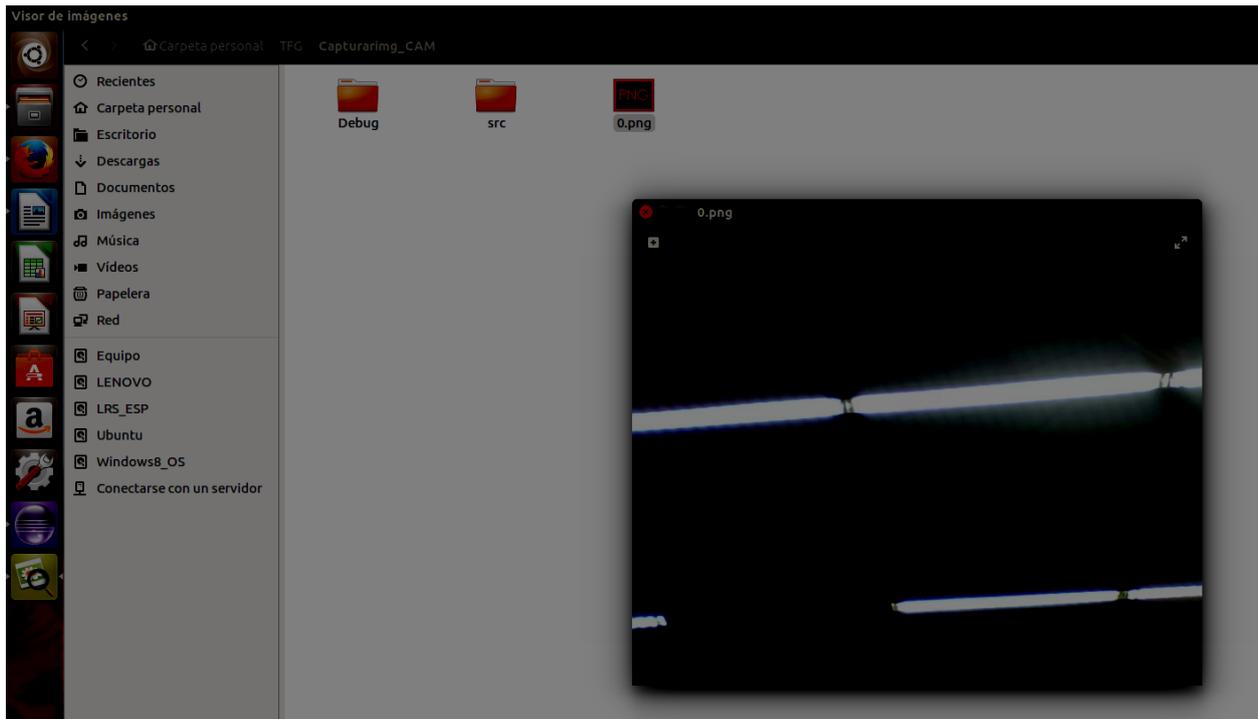


Ilustración 21 Imagen almacenada en la carpeta del proyecto

La manera que tiene OpenCV de interpretar los colores RGB es inversa, los valores que graba o muestra en pantalla en cualquier comando es BGR. Es un factor a tener en cuenta en futuros programas a desarrollar en los que el valor del píxel sea necesario.

A continuación se muestra el código empleado para la captura de imágenes:

```
/*
 * capturarimagen.cpp
 *
 * Created on: 2 nov. 2016
 * Author: fatima
 */

//CONECTA LA CÁMARA Y CAPTURA IMÁGENES CON LA TECLA S MINÚSCULA
//PARA SALIR EXIT

#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <string>
#include <sstream>

//Clase para capturar video de la cámara
cv::VideoCapture capturavideo;
```

```
//Matriz para la imagen RGB (en OpenCV BGR)

cv::Mat bgrMap;

int main() {
    char key = 0;

    // Define el número de fotografía y el nombre

    int numcaptura = 0;
    std::string nombrecaptura = "0";

    // Comandos para ejecutar o salir

    std::cout << "Pulsa 's' para sacar fotografías" << std::endl;
    std::cout << "Pulsa 'Esc' para exit" << std::endl;

    // Iniciala cámara
    capturavideo.open(0);

    // Control de errores
    if (!capturavideo.isOpened()) {
        std::cerr<<"Error. No se puede abrir el vídeo. "<<std::endl;
        return -1;
    }

    while (key!=27 && capturavideo.grab()) {

        //Retransmite el vídeo
        capturavideo.retrieve(bgrMap);

        cv::imshow("Imagen BGR", bgrMap);

        if (key == 115) {
            cv::imwrite(nombrecaptura + ".png", bgrMap);
            numcaptura++;
            nombrecaptura = static_cast<std::ostringstream*>(&(std::ostringstream()
numcaptura))->str();
        }

        key=cv::waitKey(20);

    }
    return(0);
}
```

ANEXO 8. Reconocimiento facial con imagen estática

El reconocimiento facial en una imagen es la utilidad más básica y más conocida de las librerías OpenCV. Una vez conseguido el visualizar una imagen, se procederá a tratarla de manera de que se reconozca en ella cualquier rostro que aparezca.

OpenCV realiza esto con el uso de conjuntos de “bloques” para reconocer formas llamados Haar-like features. Estos bloques son pequeños patrones como los que se muestran en la siguiente imagen:



Ilustración 22 Haar-like features

El algoritmo busca en la imagen combinaciones de estos patrones. Por ejemplo, si se quiere detectar un rostro, como es el caso, el algoritmo buscará en la imagen la combinación de estos bloques que, si se juntan, se aproximan a un rostro. (Ramirez, 2013)

Estos algoritmos se denominan clasificadores en cascada, y se pueden programar como se verá en el anexo 9 para detectar cualquier objeto tanto en imagen estática como en un video grabado o en directo.

Para usar este clasificador se creará un código donde primero se cargará el archivo .xml, que contiene la información del propio clasificador. Luego se cargará la imagen que se quiera analizar y se pasará a escala de grises. Esto facilita la detección y permite trabajar con menos información. Se buscará en esta última imagen los rostros y se marcará con un rectángulo.

El código empleado para obtenerlo se muestra a continuación:

```
#include "opencv2/core/core.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/objdetect/objdetect.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "iostream"
using namespace cv;
using namespace std;
CascadeClassifier face_cascade;
int main() {
vector<Rect> faces; // Vector para las caras detectadas
// Carga el archivo xml para detectar caras:
if
(!face_cascade.load("C:\\opencv\\data\\haarcascades\\haarcascade_frontalface_alt.xml")){
cout << "Cannot load face xml!" << endl;
return -1;
}
Mat src = imread("image.jpg"); // Almacena en la matriz src la imagen
if (src.empty()){
cout << "Cannot load image!" << endl;
return -1;
}
// Pasamos a escala de grises la imagen
Mat gray;
cvtColor(src, gray, CV_BGR2GRAY);
equalizeHist(gray, gray);
//imshow("Gray", gray);
face_cascade.detectMultiScale(gray, faces, 1.2, 3); // Detectamos las caras presentes en
la imagen

// Caras
for (int i = 0; i < int(faces.size()); i++){ // Repetimos tantas veces como caras
detectadas:
Rect r = faces[i];
rectangle(src, Point(r.x, r.y), Point(r.x + r.width, r.y + r.height), CV_RGB(0,255,0));
// Marcamos la cara
}
imshow("Detection", src); // Mostramos la imagen con las caras marcadas
waitKey(0);
return 0;
}
```

ANEXO 9. Detección de matrículas

La inteligencia artificial

La inteligencia artificial (IA) es una de las ramas de la Informática, con fuertes raíces en otras áreas como la lógica y las ciencias cognitivas. Existen muchas definiciones de lo que es la inteligencia artificial aunque todas ellas coinciden en la necesidad de validar el trabajo mediante programas.

La inteligencia artificial nace en una reunión celebrada en el verano de 1956 en Dartmouth (Estados Unidos) en la que participaron los que más tarde han sido los investigadores principales del área, los cuales redactaron la propuesta en la que aparece por primera vez el término inteligencia artificial.

El documento define el problema de la inteligencia artificial como aquel de construir una máquina que se comporte de manera que si el mismo comportamiento lo realizara un ser humano, este sería llamado inteligente.

Hay otra clasificación de la inteligencia artificial según cuáles son los objetivos finales de la investigación en este campo. Son la inteligencia artificial fuerte y la débil.

Inteligencia artificial débil

Se considera que los ordenadores únicamente pueden simular que razonan, y únicamente pueden actuar de forma inteligente. Los partidarios de la inteligencia artificial débil consideran que no será nunca posible construir ordenadores conscientes, y que un programa es una simulación de un proceso cognitivo pero no un proceso cognitivo en sí mismo.

Inteligencia artificial fuerte

En este caso se considera que un ordenador puede tener una mente y unos estados mentales, y que, por tanto, un día será posible construir uno con todas las capacidades de la mente humana. Este ordenador será capaz de razonar, imaginar, etc.

Aunque existen puntos de vista diferentes sobre qué es la inteligencia artificial, hay un acuerdo importante sobre cuáles son los resultados atribuibles a esta rama de la Informática, así como a la clasificación de los métodos y técnicas desarrollados. Repasamos a continuación los cuatro grandes temas de la inteligencia artificial.

1. *Resolución de problemas y búsqueda.* La inteligencia artificial tiene como objetivo resolver problemas de índole muy diferente. Para poder cumplir este objetivo, dado un problema es necesario formalizarlo para poderlo resolver. Este tema se centra en cómo formalizarlo y las formas de resolución.

2. *Representación del conocimiento y sistemas basados en el conocimiento.* Es frecuente que los programas en inteligencia artificial necesiten incorporar conocimiento del dominio de

aplicación (por ejemplo, en medicina) para poder resolver los problemas. Este tema se centra en estos aspectos.

3. *Aprendizaje automático*. El rendimiento de un programa puede incrementarse si el programa aprende de la actividad realizada y de sus propios errores. Se han desarrollado métodos con este objetivo. Existen también herramientas que permiten extraer conocimiento a partir de bases de datos.

4. *Inteligencia artificial distribuida*. Durante sus primeros años la inteligencia artificial era monolítica. Ahora, con los ordenadores multiprocesador e Internet, hay interés en soluciones distribuidas. Estas van desde versiones paralelas de métodos ya existentes a nuevos problemas relacionados con los agentes autónomos (programas software con autonomía para tomar decisiones e interactuar con otros).

Además de los cuatro temas mencionados más arriba, existen otros que están fuertemente relacionados con la inteligencia artificial como son la visión artificial, la robótica, el reconocimiento, la conducción autónoma, etc. (Torra, 2011)

Detección de matrículas

OpenCV cuenta con varios clasificadores en cascada entrenados que podemos encontrar en el directorio `opencv\build\etc\haarcascades` ó en `opencv\build\etc\lbpascades` estos clasificadores sirven para la detección de rostros, elementos concretos del rostro, o gestos entre otras cosas. En el caso de que se desee detectar otro tipo de objeto se puede entrenar clasificadores propios para que detecte por ejemplo, un animal determinado, un automóvil, un teléfono móvil, una matrícula o lo que se desee según lo requiera el proyecto.

Con el uso del mobile mapping, es necesario borrar matrículas y rostros para la publicación de imágenes en la web. Para eso se pueden desarrollar algoritmos que los detecten y los difuminen automáticamente.

A continuación se detallará el proceso seguido para entrenar un clasificador que detecte matrículas de coche. El modelo estándar elegido para la detección será el de España actualmente, que se muestra en la siguiente imagen.



Ilustración 23 Matrícula española Fuente especificada no válida.

En la carpeta `opencv\build\x86\vc11\bin` se pueden encontrar los ejecutables `opencv_createsamples.exe` y `opencv_traincascade.exe` que ayudarán a realizar el trabajo de entrenamiento de manera sencilla, el primero prepara el conjunto de imágenes positivas y el otro genera el clasificador en cascada.

La primera tarea que se debe realizar es recolectar el conjunto de imágenes, se necesitan un conjunto positivo y uno negativo. El conjunto de imágenes positivas es aquel que contiene varias imágenes con el objeto que se desea detectar con OpenCV y el negativo son todas aquellas imágenes que no contengan el objeto. (Marín, 2015)

Una vez recolectadas, se deberán crear dos txt listando todos los ficheros que hay de imágenes positivas y negativas.

Para listar las imágenes positivas se utiliza el programa Object Marker, que se tendrá que compilar dando como argumentos el nombre del archivo de salida y el path donde se encuentren las imágenes. Para definir la zona en la que aparece el objeto se seleccionará el recuadro con el ratón, se dará a la barra espaciadora para grabar y a B para pasar a la siguiente.

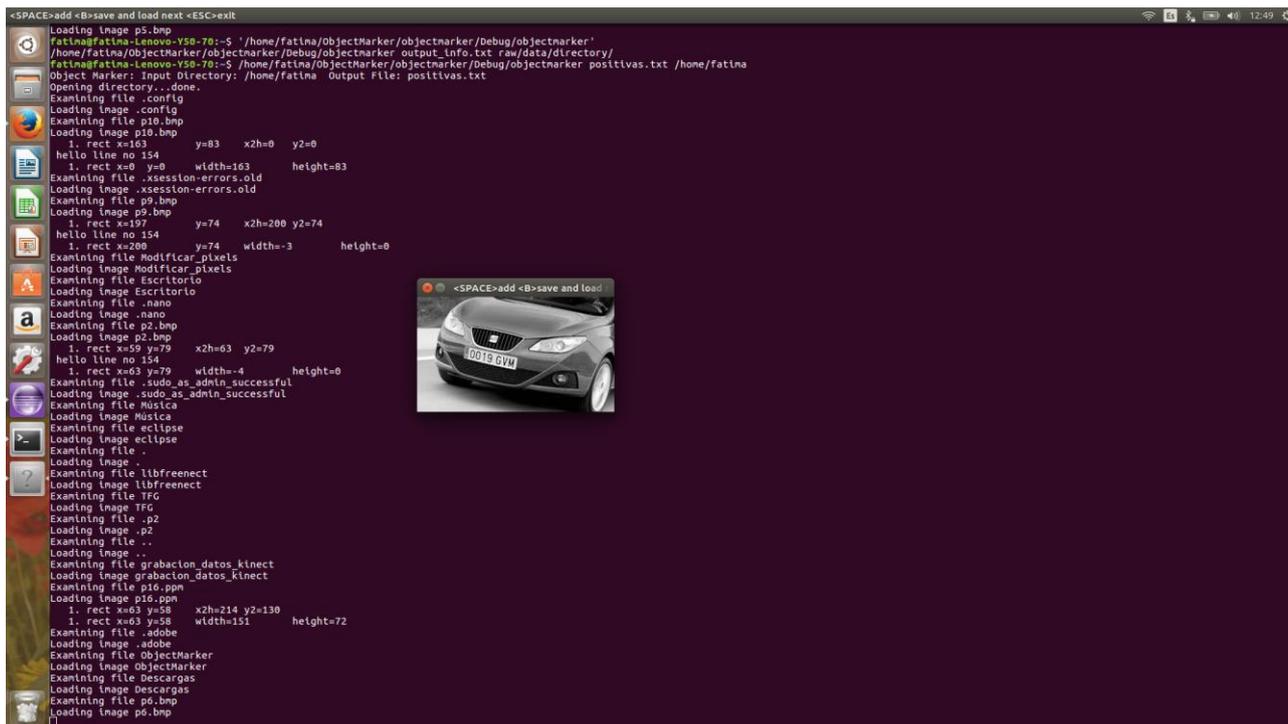


Ilustración 24 Object Marker

El programa ObjectMarker se tendrá que copiar en un archivo fuente y compilarlo desde un programa como puede ser Eclipse. El código se muestra a continuación:

```

/*
 * objectmarker.cpp
 *
 * Created on: 13 dic. 2016
 * Author: fatima
 */

/*****objectmarker.cpp*****/

```

Objectmarker for marking the objects to be detected from positive samples and then creating the description file for positive images.

compile this code and run with two arguments, first one the name of the descriptor file and the second one the address of the directory in which the positive images are located

while running this code, each image in the given directory will open up. Now mark the edges of the object using the mouse buttons

then press then press "SPACE" to save the selected region, or any other key to discard it. Then use "B" to move to next image. the program automatically quits at the end. press ESC at anytime to quit.

*the key B was chosen to move to the next image because it is closer to SPACE key and nothing else.....

author: achu_wilson@rediffmail.com

*/

```
#include <opencv/cv.h>
#include <opencv/cvaux.h>
#include <opencv/highgui.h>
```

```
// for filelisting
#include <stdio.h>
#include <sys/io.h>
// for fileoutput
#include <string>
#include <fstream>
#include <sstream>
#include <dirent.h>
#include <sys/types.h>
```

```
using namespace std;
```

```
IplImage* image=0;
IplImage* image2=0;
//int start_roi=0;
int roi_x0=0;
int roi_y0=0;
int roi_x1=0;
int roi_y1=0;
int numOfRec=0;
int startDraw = 0;
char* window_name="<SPACE>add <B>save and load next <ESC>exit";
```

```
string IntToString(int num)
{
    ostringstream myStream; //creates an ostringstream object
    myStream << num << flush;
    /*
    * outputs the number into the string stream and then flushes
    * the buffer (makes sure the output is put into the stream)
    */
    return(myStream.str()); //returns the string form of the stringstream object
};
```

```
void on_mouse(int event,int x,int y,int flag, void *param)
{
    if(event==CV_EVENT_LBUTTONDOWN)
    {
        if(!startDraw)
        {
```

```
        roi_x0=x;
        roi_y0=y;
        startDraw = 1;
    } else {
        roi_x1=x;
        roi_y1=y;
        startDraw = 0;
    }
}
if(event==CV_EVENT_MOUSEMOVE && startDraw)
{

    //redraw ROI selection
    image2=cvCloneImage(image);
    cvRectangle(image2,cvPoint(roi_x0,roi_y0),cvPoint(x,y),CV_RGB(255,0,255),1);
    cvShowImage(window_name,image2);
    cvReleaseImage(&image2);
}

}

int main(int argc, char** argv)
{
    char iKey=0;
    string strPrefix;
    string strPostfix;
    string input_directory;
    string output_file;

    if(argc != 3) {
        fprintf(stderr, "%s output_info.txt raw/data/directory/\n", argv[0]);
        return -1;
    }

    input_directory = argv[2];
    output_file = argv[1];

    /* Get a file listing of all files with in the input directory */
    DIR *dir_p = opendir (input_directory.c_str());
    struct dirent *dir_entry_p;

    if(dir_p == NULL) {
        fprintf(stderr, "Failed to open directory %s\n", input_directory.c_str());
        return -1;
    }

    fprintf(stderr, "Object Marker: Input Directory: %s Output File: %s\n",
input_directory.c_str(), output_file.c_str());

    //    init highgui
```

```
cvAddSearchPath(input_directory);
cvNamedWindow(window_name,1);
cvSetMouseCallback(window_name,on_mouse, NULL);

fprintf(stderr, "Opening directory...");
//  init output of rectangles to the info file
ofstream output(output_file.c_str());
fprintf(stderr, "done.\n");

while((dir_entry_p = readdir(dir_p)) != NULL)
{
    numOfRec=0;

    if(strcmp(dir_entry_p->d_name, ""))
        fprintf(stderr, "Examining file %s\n", dir_entry_p->d_name);

    /* TODO: Assign postfix/prefix info */
    strPostfix="";
    //strPrefix=input_directory;
    strPrefix=dir_entry_p->d_name;
    //strPrefix+=bmp_file.name;
    fprintf(stderr, "Loading image %s\n", strPrefix.c_str());

    if((image=cvLoadImage(strPrefix.c_str(),1)) != 0)
    {
        //  work on current image
        do

    {
        cvShowImage(window_name,image);

        // used cvWaitKey returns:
        //  <B>=66      save added rectangles and show next image
        //  <ESC>=27    exit program
        //  <Space>=32  add rectangle to current image
        //  any other key clears rectangle drawing only
        iKey=cvWaitKey(0);
        switch(iKey)
        {

        case 27:

            cvReleaseImage(&image);
            cvDestroyWindow(window_name);
            return 0;

        case 32:

            numOfRec++;
```

```

        printf("                                %d.                rect
x=%d\ty=%d\tx2h=%d\ty2=%d\n", numOfRec, roi_x0, roi_y0, roi_x1, roi_y1);
        //printf("                                %d.                rect
x=%d\ty=%d\twidth=%d\theight=%d\n", numOfRec, roi_x1, roi_y1, roi_x0-roi_x1, roi_y0-roi_y1);
        // currently two draw directions possible:
        //          from top left to bottom right or vice versa
        if(roi_x0<roi_x1 && roi_y0<roi_y1)
        {
                printf("                                %d.                rect
x=%d\ty=%d\twidth=%d\theight=%d\n", numOfRec, roi_x0, roi_y0, roi_x1-roi_x0, roi_y1-roi_y0);
                // append rectangle coord to previous line content
                strPostfix+=" "+IntToString(roi_x0)+" "+IntToString(roi_y0)+"
"+IntToString(roi_x1-roi_x0)+" "+IntToString(roi_y1-roi_y0);

        }
        else
                //(roi_x0>roi_x1 && roi_y0>roi_y1)
        {
                printf(" hello line no 154\n");
                printf("                                %d.                rect
x=%d\ty=%d\twidth=%d\theight=%d\n", numOfRec, roi_x1, roi_y1, roi_x0-roi_x1, roi_y0-roi_y1);
                // append rectangle coord to previous line content
                strPostfix+=" "+IntToString(roi_x1)+" "+IntToString(roi_y1)+"
"+IntToString(roi_x0-roi_x1)+" "+IntToString(roi_y0-roi_y1);
        }

        break;
    }
}
while(iKey!=66);

{
// save to info file as later used for HaarTraining:
//      <rel_path>\bmp_file.name numOfRec x0 y0 width0 height0 x1 y1 width1
height1...
if(numOfRec>0 && iKey==66)
{
//append line
/* TODO: Store output information. */
output << strPrefix << " " << numOfRec << strPostfix << "\n";

cvReleaseImage(&image);
}

else
{
fprintf(stderr, "Failed to load image, %s\n", strPrefix.c_str());
}
}
}

```

```
}}  
  
output.close();  
cvDestroyWindow(window_name);  
closedir(dir_p);  
  
return 0;  
}
```

Este programa crea el txt en el que aparecerá la siguiente información:

- Nombre de la imagen
- Número de veces que aparece el elemento a identificar
- Coordenadas x y de la posición del objeto en la imagen

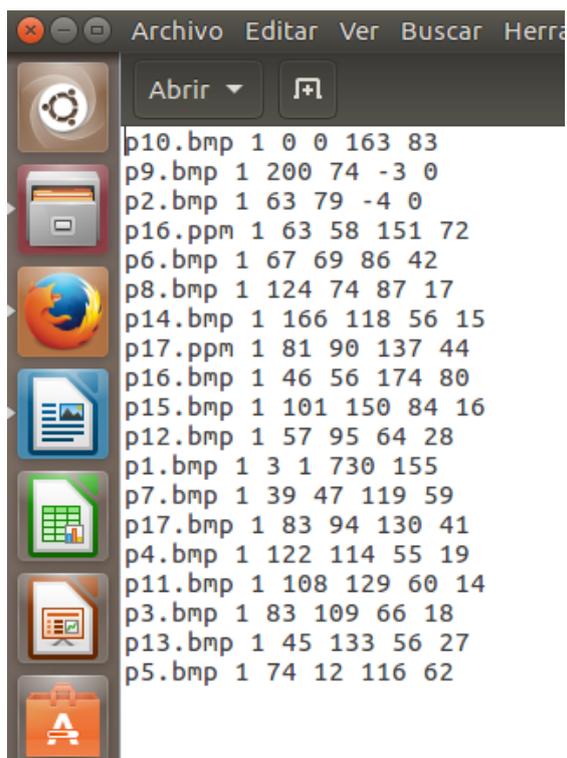


Ilustración 25 Lista de imágenes positivas

Para generar el txt de las imágenes negativas, el comando lo ofrece Linux y bastará con introducir lo siguiente en consola:

```
ls Escritorio/Matriculas/Negativas/* Escritorio/Matriculas/Negativas/Negativas.txt
```

Obteniendo así una lista de todas las que se encuentren en la carpeta.

A continuación se utilizara la herramienta createsamples ejecutando desde la línea de comando lo siguiente:

```
opencv_createsamples -info /home/fatima/Escritorio/MATRICULAS/positivas/positivas.txt -vec /home/fatima/Escritorio/MATRICULAS/positivas/muestra.vec -w 20 -h 20
```

Donde el primer path indica donde se encuentra la información de las imágenes y la segunda dirección indica donde se guardará el archivo vectorial creado.

Ejecutando esto se obtiene lo siguiente:



```
fatima@fatima-Lenovo-Y50-70:~$ opencv_createsamples -info /home/fatima/Escritorio/MATRICULAS/positivas/positivas.txt -vec /home/fatima/Escritorio/MATRICULAS/positivas/muestra.vec -w 20 -h 20
Info file name: /home/fatima/Escritorio/MATRICULAS/positivas/positivas.txt
Img file name: (NULL)
Vec file name: /home/fatima/Escritorio/MATRICULAS/positivas/muestra.vec
BG file name: (NULL)
Num: 1000
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Width: 20
Height: 20
Max Scale: 1
Create training samples from images collection...
/home/fatima/Escritorio/MATRICULAS/positivas/positivas.txt(20) : parse errorDone. Created 19 samples
fatima@fatima-Lenovo-Y50-70:~$
```

Ilustración 26 Resultado createsamples

Donde da información de los datos generados y los samples que ha creado, 19 en este caso. Crea un sample por cada foto positiva que se posea.

Para obtener tantas positivas se puede grabar un video del objeto en el que se vayan viendo distintas perspectivas y condiciones de luz, después con un programa en el que se permitan extraer fotogramas del video se generan estas fotos positivas.

A partir de aquí comienza el entrenamiento. Se realizará con la herramienta haartraining, utilizando para ello el archivo índice generado con las imágenes negativas y el archivo de muestras generado en el paso anterior.

Antes de nada, hay que instalar la herramienta haartraining de la siguiente manera:

```
sudo apt install libopencv-dev
```

Ahora si se podrá comenzar el entrenamiento escribiendo lo siguiente en la consola:

```
opencv_haartraining -data /home/fatima/Escritorio/MATRICULAS/training -vec  
/home/fatima/Escritorio/MATRICULAS/positivas/muestra.vec -bg  
/home/fatima/Escritorio/MATRICULAS/negativas/negativa.txt -nstages 20 -nsplits 2 -  
minhitrate 0.999 -maxfalsealarm 0.5 -npos 17 -nneg 317 -nonsym -mem 2000 -mode ALL -w 20  
-h 20
```

En este comando se definen varias variables que luego el programa interpretará:

```
//Dirección en la que se guarde  
Data dir name: /home/fatima/Escritorio/MATRICULAS/training  
//Nombre del vector  
Vec file name: /home/fatima/Escritorio/MATRICULAS/positivas/muestra.vec  
//Nombre del background (Separar con un salto de linea cada elemento de la lista del .txt)  
BG file name: /home/fatima/Escritorio/MATRICULAS/negativas/negativa.txt, is a vecfile: no  
//Numero de fotos positivas  
Num pos: 17  
//Número de fotos negativas  
Num neg: 317  
//Etapas de entrenamiento  
Num stages: 20  
//Divisiones  
Num splits: 2 (tree as weak classifier)  
//Memoria utilizada  
Mem: 2000 MB  
//Si el objeto es simétrico o no  
Symmetric: FALSE  
Min hit rate: 0.999000  
Max false alarm rate: 0.500000  
Weight trimming: 0.950000  
Equal weights: FALSE  
Mode: ALL  
Width: 20  
Height: 20  
Applied boosting algorithm: GAB  
Error (valid only for Discrete and Real AdaBoost): misclass  
Max number of splits in tree cascade: 0  
Min number of positive samples per cluster: 500  
Required leaf false alarm rate: 9.53674e-07
```

Al darle a ejecutar saldrá lo siguiente en pantalla, en la que se va viendo los cluster que va creando.

```
fatima@fatima-Lenovo-Y50-70: ~
*** 1 cluster ***
POS: 17 17 1.000000
NEG: 317 0.000180606
BACKGROUND PROCESSING TIME: 7.00
Precalculation time: 0.00
+-----+
| N |%SMP|F| ST.THR | HR | FA | EXP. ERR|
+-----+
| 1|100%|-| 0.864707| 1.000000| 0.072555| 0.068862|
+-----+
Stage training time: 1.00
Number of used features: 2
Parent node: 5
Chosen number of splits: 0
Total number of splits: 0
Tree Classifier
Stage
+-----+
| 0| 1| 2| 3| 4| 5| 6|
+-----+
0---1---2---3---4---5---6
Parent node: 6
*** 1 cluster ***
POS: 17 17 1.000000
NEG: 317 1.27434e-05
BACKGROUND PROCESSING TIME: 95.00
Precalculation time: 0.00
+-----+
| N |%SMP|F| ST.THR | HR | FA | EXP. ERR|
+-----+
| 1|100%|-| 1.000000| 1.000000| 0.000000| 0.000000|
+-----+
Stage training time: 1.00
Number of used features: 2
```

Ilustración 27 Creación de cascada

Ahora se generara el XML a partir de la cascada generada. Primero se compilará el programa convert_cascades.c situado en el código de OpenCV en la carpeta samples → c.

Tras esto se ejecutará la siguiente orden en consola:

```
/convert_cascade --size=30x30 /home/fatima/Escritorio/MATRICULAS/cascade matriculas.xml
```

Que indica el tamaño, el path de salida de datos y el nombre del fichero xml a crear.


```
* detectar.cpp
*
* Created on: 13 dic. 2016
* Author: fatima
*/

/*****detect.c*****/
/*
*/

#include <stdio.h>
#include "cv.h"
#include "highgui.h"

CvHaarClassifierCascade *cascade;
CvMemStorage *almacenamiento;

void detect( IplImage *img );

int main( int argc, char** argv )
{
    CvCapture *capture;
    IplImage *frame;
    int key;
    char *filename = "/home/fatima/Escritorio/MATRICULAS/cascade/matriculas.xml";

    cascade = ( CvHaarClassifierCascade* )cvLoad( filename, 0, 0, 0 );
    storage = cvCreateMemStorage(0);
    capture = cvCaptureFromCAM(0);

    assert( cascade && storage && capture );

    cvNamedWindow("video", 1);

    while(1) {
        frame = cvQueryFrame( capture );

        detect(frame);

        key = cvWaitKey(50);
    }

    cvReleaseImage(&frame);
    cvReleaseCapture(&capture);
    cvDestroyWindow("video");
    cvReleaseHaarClassifierCascade(&cascade);
    cvReleaseMemStorage(&storage);

    return 0;
}
```

```
}  
  
void detect(IplImage *img)  
{  
    int i;  
  
    CvSeq *object = cvHaarDetectObjects(  
        img,  
        cascade,  
        storage,  
        1.5, //-----SCALE FACTOR  
        2, //-----MIN NEIGHBOURS  
        1, //-----  
        // CV_HAAR_DO_CANNY_PRUNING,  
        cvSize( 30,30), // -----MINSIZE  
        cvSize(640,480) );//-----MAXSIZE  
  
    for( i = 0 ; i < ( object ? object->total : 0 ) ; i++ )  
    {  
        CvRect *r = ( CvRect* )cvGetSeqElem( object, i );  
        cvRectangle( img,  
            cvPoint( r->x, r->y ),  
            cvPoint( r->x + r->width, r->y + r->height ),  
            CV_RGB( 255, 0, 0 ), 2, 8, 0 );  
  
        //printf("%d,%d\nnumber =%d\n",r->x,r->y,object->total);  
  
    }  
  
    cvShowImage( "video", img );  
}
```

Una vez compilado el programa, el resultado es el que se ve a continuación. Se puede apreciar que se sitúa un cuadrado encima de la matrícula. En este caso es un video en directo, pero se podría enfocar también a fotos.

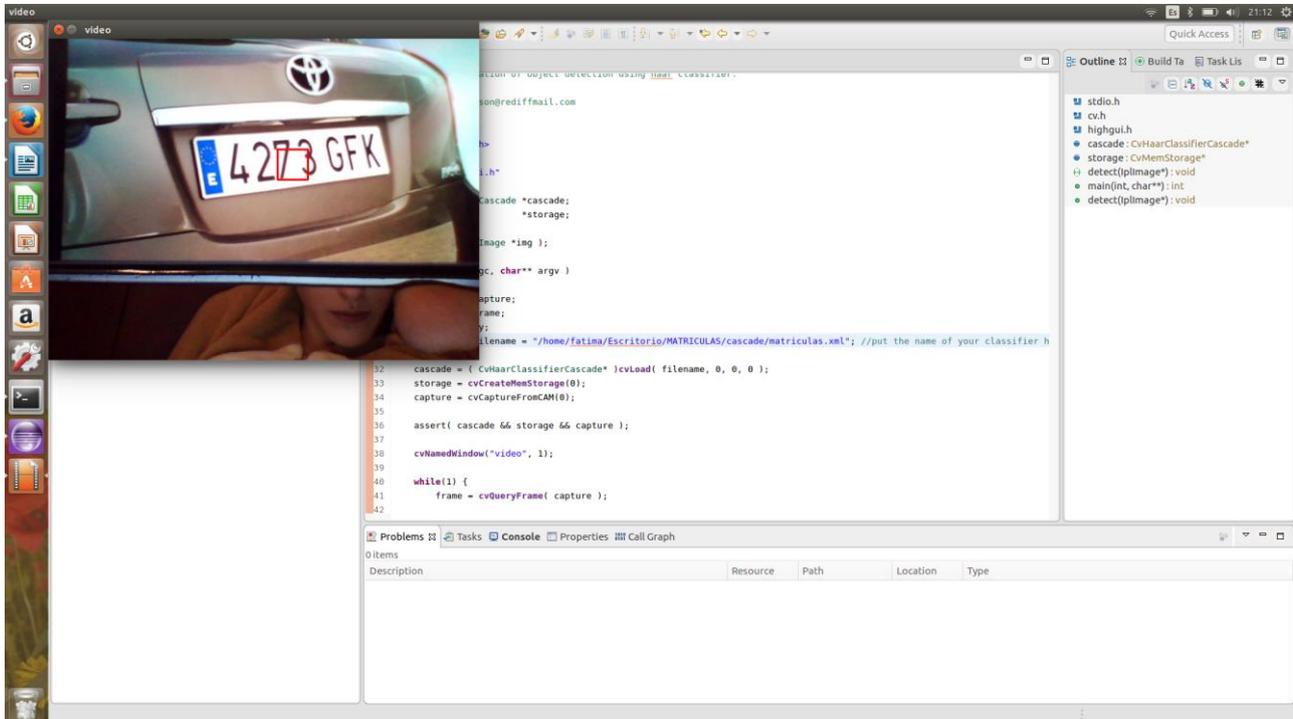


Ilustración 29 Resultados obtenidos detección de matrículas

Para concluir, se ha de decir que para acortar el tiempo de reconocimiento y ampliar el porcentaje de aciertos en la detección, es totalmente aconsejable introducir más imágenes de muestra y lo fundamental es que sea en todas las circunstancias posibles de ángulo e iluminación como ya han mencionado en varios trabajos de este tipo, como M. Delgado Rodríguez en su trabajo de Extracción automática de caras en imágenes captadas con móviles Android. (Rodríguez, 2012)

ANEXO 10. Operaciones con el valor de los pixeles de una imagen

Variación de la gravedad

La ventaja de poder trabajar con cada pixel de la imagen es la precisión con la que se puede trabajar con ellas, por ejemplo en cálculos de geofísica.

Como se ha comentado antes, OpenCV ofrece la posibilidad de cargar imágenes TIFF, pero al ser una imagen piramidal da bastantes problemas. Tras probar otras librerías mencionadas en otro anexo, se ha decidido convertir la imagen a .JPG con Qgis para que cree un fichero, JPGW y no perder la georreferenciación.

La imagen cargada, al venir de un mdt, debería tener una banda, pero se debe especificar al definir la matriz en la que se volcará está en el programa, definiéndola de la siguiente manera:

```
Mat nuevaimagen(img.rows, img.cols, CV_8UC1)
```

Esto quiere decir, que la matriz que se crea tiene las mismas dimensiones que img, que es la imagen original, y que los datos que poseerá serán Integer de 8 bits y trabajará con 1 canal.

Con la imagen se va a calcular la variación de la gravedad respecto a la altura y se creará una imagen final por colores que dependerá de los rangos de dicha variación.

Como la imagen trabaja con valores de pixel de 0 a 255, se deberá calcular un factor para calcular la altura real del terreno, esto se podrá evitar consultando el archivo .JGW, pero no se ha realizado porque no es el objeto de este trabajo. Por tanto, se introducirá la altura máxima por consola.

Para controlar que los cálculos se efectúen en toda la imagen y sabiendo que la matriz RGB es formada mediante el formato de grabación BIL, la imagen se convertirá a escala de grises. BIL es el formato de bandas intercaladas por líneas en la que los Nd se ordenan por línea, es decir se graba la primera línea de todas las bandas, luego la segunda y así sucesivamente.

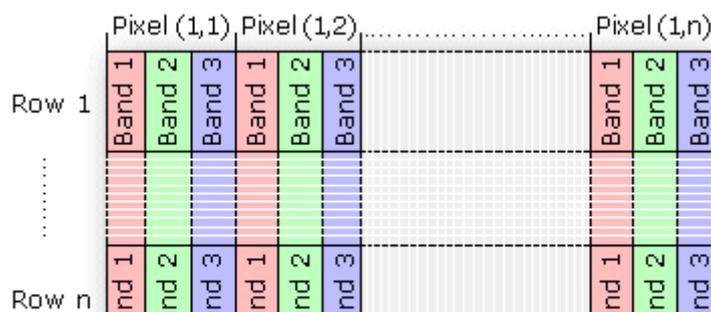


Ilustración 30 Formato de grabación BIL Fuente especificada no válida.

Una vez obtenida la altura de cada valor del pixel y aplicando el factor, se calculará la corrección de la gravedad del aire libre. La gravedad decrece con la distancia al centro de la Tierra en $1/r^2$.

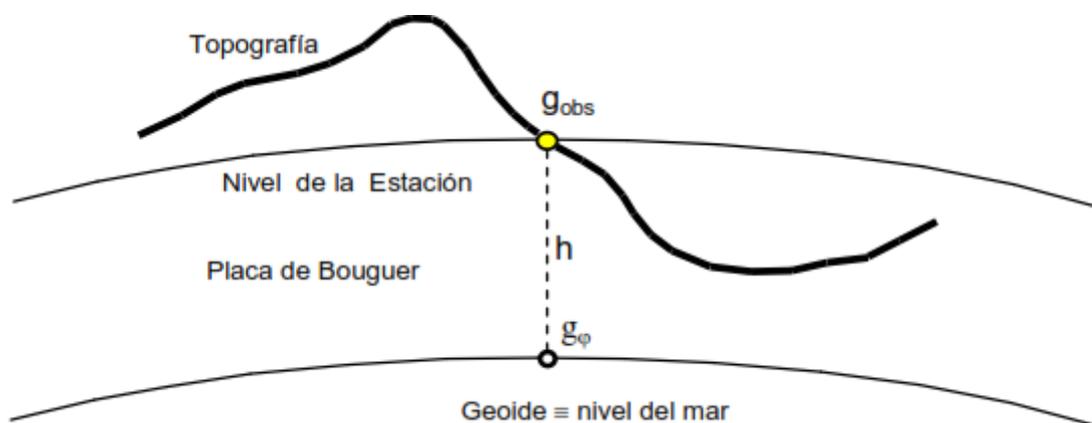


Ilustración 31 Corrección aire libre

Para este análisis basta suponer la Tierra como esférica y no rotacional, por lo tanto $g = GM/R^2$. Si la altura sobre el nivel del mar cambia (por la topografía), la gravedad será distinta porque cambia la distancia al centro de la Tierra por $(R+h)$.

Por tanto:

$$g = G \frac{M}{r^2}$$

$$\rightarrow \frac{dg}{dr} = -2G \frac{M}{r^3}$$

$$\frac{dg}{dr} = -2 \frac{g}{r}$$

$$dg = -\frac{2gh}{r}$$

Ecuación 4 Corrección de la gravedad según altura (Desarrollo)

Para un valor medio de g y r resulta que la corrección será:

$$dg = -0.3086 \text{ mgal/m}$$

Ecuación 5 Corrección de la gravedad según altura (Final)

El signo menos proviene del hecho que al aumentar R disminuye g , entonces la corrección será aditiva. (Estrada, 2012)

Modificando los pixeles, se calcularía el nuevo valor y se grabaría en la nueva imagen creada previamente.

Para darle valor según intervalos, se ha creado una matriz RGB y estudiando el valor de la corrección, se han realizado grupos y se le ha dado un color a cada grupo. Como la variación es respecto a la altura, la imagen generada será parecida a la original pero coloreada.

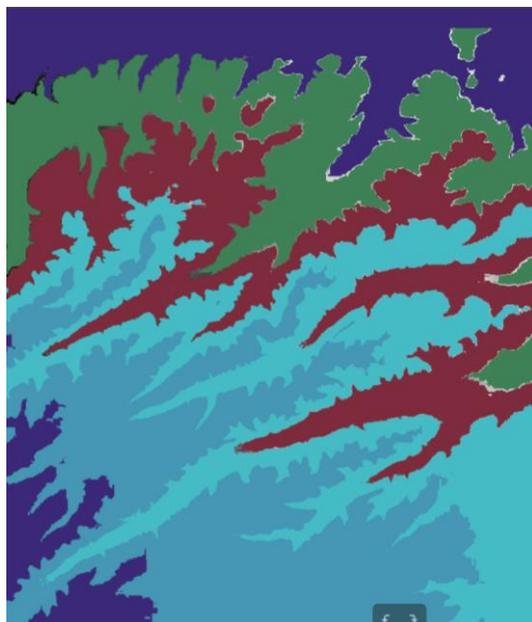


Ilustración 32 Imagen clasificada por intervalos

Esto nos da unos resultados muy precisos ya que el cálculo va pixel a pixel, y a la vez es muy visual. El resultado obtenido no dista mucho de clasificar los valores en un programa GIS, pero es interesante el saber cómo se realiza y el poder realizarlo manual en caso de querer desarrollar algo que en estos programas no exista.

El código es el que se muestra a continuación. Cabe decir que no está optimizado y que el tiempo de procesamiento es lento, y que en un futuro en caso de seguir esta línea será mejorado.

```
* lidar.cpp
*
* Created on: 17 nov. 2016
* Author: fatima
*/

#include <iostream>
#include <opencv2/core.hpp>
#include <opencv2/highgui.hpp>
#include <math.h>
```

```
#include <cvwimage.h>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/opencv.hpp>
#include <opencv2/cvconfig.h>
#include <stdio.h>

using namespace cv;

char name[]="mdt2.jpg";

int main(){

    long max;
    std::cout<<"Introduce la altura máxima del MDT ";
    std::cin>>max;
    int min;
    std::cout<<"Introduce la altura mínima del MDT ";
    std::cin>>min;

    long factor;
    factor= max/255;
    int contador=0;

    Mat img= imread("mdt2.jpg");
    Mat imagenpixel = img.clone();
    Mat imggris;
    // Creo la nueva matriz con dimension igual a img y con datos integer 8bits y 1
canal
    Mat nuevaimagen(img.rows,img.cols,CV_8UC1);
//Convierto la imagen jpg RGB a escala de grises

cvtColor(img,imggris,CV_RGB2GRAY);
imwrite("imagen_gris.jpg", imggris);

//Parámetros de compresión de la imagen de salida
    std::vector<int> compression_params;
    compression_params.push_back(CV_IMWRITE_PNG_COMPRESSION);
    compression_params.push_back(9);

//Filas y columnas de la imagen inicial
    int filas=img.rows;
    int columnas=img.cols;

    printf("Filas %i ", filas);
    printf("Columnas %i ", columnas);

//Leer imagen
```

```
for(long int i=0; i<filas; i++)
{
    for( long int j=0; j<columnas; j++)
    {

contador=contador+1;

int pixel=imggris.at<char>(i,j);
std::cout<< "Pixel: "<<pixel << "\n";
int altura=pixel*factor;
imggris.at<char>(i,j)=altura;
    }
}

imwrite("fotocorrec.jpg", imggris, compression_params);
printf("contador %i ", contador);
imwrite("fotomodificada.png", imagenpixel, compression_params);

imshow("Imagen Modificada", imagenpixel);

Mat imagengris=imread("mdt2.jpg",CV_LOAD_IMAGE_GRAYSCALE);
Mat imgcol=imread("mdt2.jpg",CV_LOAD_IMAGE_COLOR);
int canales=imgcol.channels();
printf("CANALES DE LA IMAGEN %i",canales);

for(long int i=0; i<filas; i++)
{
    for( long int j=0; j<columnas; j++)
    {
int pixel=imagengris.at<char>(i,j);

//Colorear imagen

Vec3b color = imgcol.at<Vec3b>(Point(i,j));

if (pixel>-128 && pixel<-75) {

int r = imgcol.at<cv::Vec3b>(i,j)[0];
int g = imgcol.at<cv::Vec3b>(i,j)[1];
int b = imgcol.at<cv::Vec3b>(i,j)[2];

color.val[0]= 200;
color.val[1]= 188;
color.val[2]= 68;
imgcol.at<Vec3b>(Point(i,j)) = color;

printf("RED ""%i""\n",r);
```

```
printf("GREEN \"%i\"\n",g);
printf("BLUE \"%i\"\n",b);

} else if (pixel>-76 && pixel<-25) {

    color.val[0]= 184;
    color.val[1]= 155;
    color.val[2]= 61;
    imgcol.at<Vec3b>(Point(i,j)) = color;

} else if (pixel>-26 && pixel<25) {

    color.val[0]= 122;
    color.val[1]= 58;
    color.val[2]= 46;
    imgcol.at<Vec3b>(Point(i,j)) = color;

} else if (pixel>26 && pixel<75) {

    color.val[0]= 85;
    color.val[1]= 123;
    color.val[2]= 67;
    imgcol.at<Vec3b>(Point(i,j)) = color;

} else if (pixel>76 && pixel<128) {

    color.val[0]= 59;
    color.val[1]= 48;
    color.val[2]= 126;
    imgcol.at<Vec3b>(Point(i,j)) = color;
}

imwrite("fotocolores.jpg", imgcol, compression_params);
}

waitKey(5000);
```

ANEXO 11. Armónicos esféricos para el cálculo del potencial

La ecuación que representa el geopotencial es la siguiente:

$$W_a(r, \lambda, \varphi) = \frac{GM}{r} \sum_{l=0}^{lmax} \sum_{m=0}^l \frac{R^l}{r} P_{lm}(\sin \varphi) (C_{lm}^W \cos m\lambda + S_{lm}^W \sin m\lambda)$$

Ecuación 6 Geopotencial

Donde:

(r, λ, φ) : Coordenadas esféricas geocéntricas del punto a calcular

R: Radio de referencia

GM: Producto de la constante gravitacional y la masa de la Tierra

l, m: Grado y orden de los armónicos esféricos

P_{lm} : Normalización de las funciones de Legendre

C_{lm}^W, S_{lm}^W : Coeficientes de Stoke

Para calcular la gravedad en superficie se necesita el gradiente del potencial:

$$g = |\Delta W|$$

Ecuación 7 Gravedad en superficie

Que viene dado por la siguiente fórmula:

$$|\Delta W| = \sqrt{[W_{ar} + \Phi_r]^2 + \left[\frac{1}{r \cos \varphi} (W_{a\lambda} + \Phi_\lambda)\right]^2 + \left[\frac{1}{r} (W_{a\varphi} + \Phi_\varphi)\right]^2}$$

Ecuación 8 Gradiente del potencial

Donde Φ es el potencial centrífugo en coordenadas esféricas, expresado de la siguiente manera:

$$\Phi = \frac{1}{2} \omega^2 r^2 (\cos \varphi)^2$$

Ecuación 9 Potencial centrífugo

El potencial centrífugo se debe derivar respecto a (r, λ, φ) al igual que el geopotencial, obteniendo las siguientes ecuaciones cuyo resultado se deberá sustituir en la expresión 3.

$$W_{ar} = -\frac{GM}{r^2} \sum_{l=0}^{lmax} \frac{R^l}{r} (l+1) \sum_{m=0}^l P_{lm}(\sin \varphi) (C_{lm}^W \cos m\lambda + S_{lm}^W \sin m\lambda)$$

$$W_{a\lambda} = \frac{GM}{r} \sum_{l=0}^{lmax} \frac{R^l}{r} \sum_{m=0}^l m P_{lm}(\sin \varphi) (S_{lm}^W \cos m\lambda - C_{lm}^W \sin m\lambda)$$

$$W_{a\varphi} = \frac{GM}{r} \sum_{l=0}^{lmax} \frac{R^l}{r} \sum_{m=0}^l \frac{\delta P_{lm}(\sin \varphi)}{\delta \varphi} (C_{lm}^W \cos m\lambda + S_{lm}^W \sin m\lambda)$$

Ecuación 10 Derivadas del geopotencial

$$\Phi_r = \omega^2 r (\cos \varphi)^2, \quad \Phi_\lambda = 0, \quad \Phi_\varphi = -\omega^2 r^2 \cos \varphi \sin \varphi$$

Ecuación 11 Derivadas del potencial centrífugo

Para que el cálculo sea más preciso se hace una transformación de coordenadas. Primero se convierten de geodésicas a cartesianas geocéntricas, y estas a esféricas utilizando las siguientes fórmulas de conversión:

$$X = (N + h) * \cos \varphi * \cos \lambda$$

$$Y = (N + h) * \cos \varphi * \sin \lambda$$

$$Z = \frac{b^2}{a^2} N + h * \sin \varphi$$

Ecuación 12 De coordenadas geodésicas a geocéntricas

Donde X,Y,Z son coordenadas geocéntricas, φ, λ y h son coordenadas geodésicas, a y b son parámetros del elipsoide de referencia y N la normal principal.

$$r = \sqrt{X^2 + Y^2 + Z^2}$$

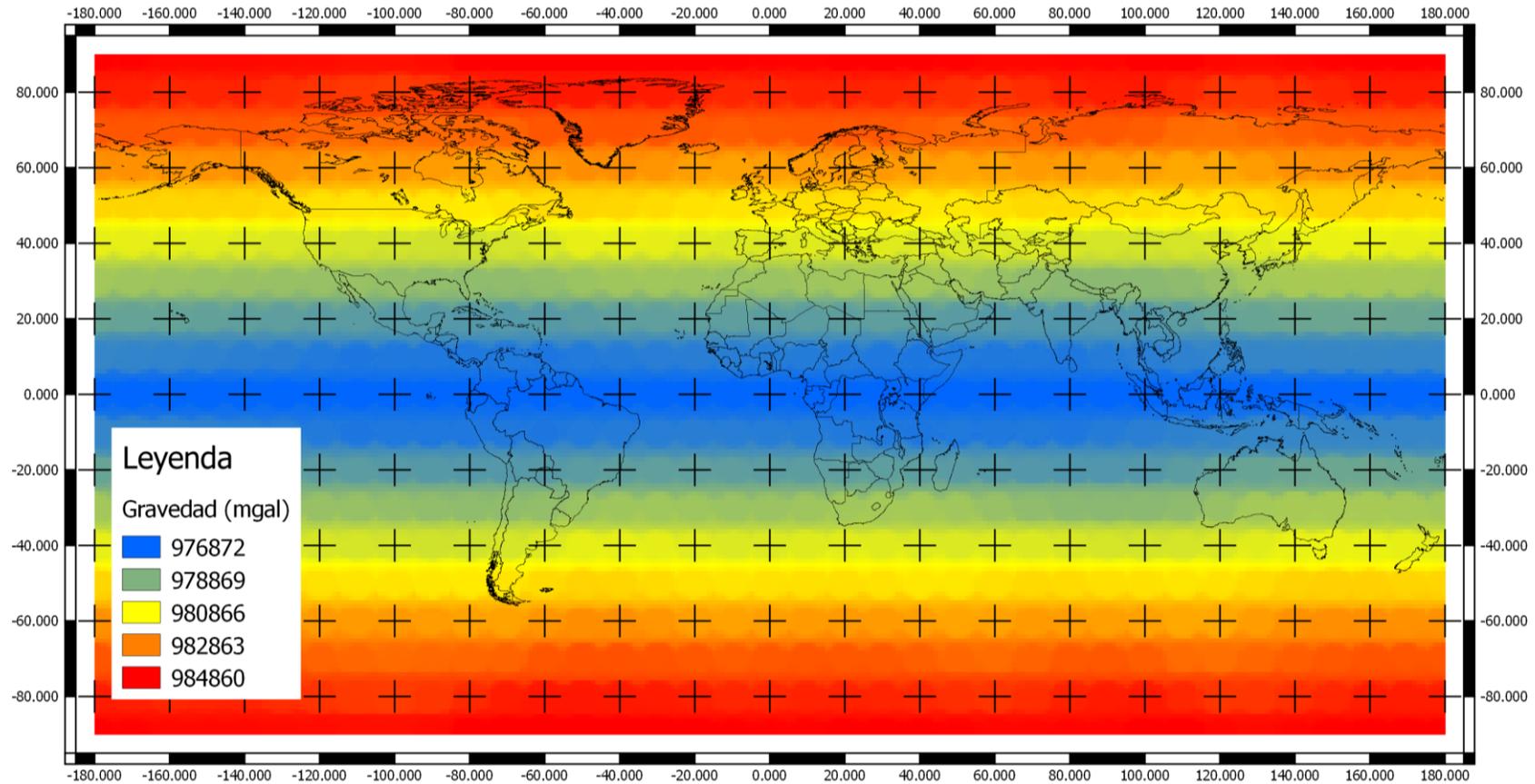
$$\theta = \arccos \frac{Z}{r}$$

$$\lambda = \arctg \frac{Y}{X}$$

Ecuación 13 De geocéntricas a esféricas

ANEXO 12. Gravedad en superficie calculada a partir de parámetros GRACE 2016

GRAVEDAD EN SUPERFICIE CALCULADA A PARTIR DE PARÁMETROS GRACE 2016



Fátima Sáiz Álvaro

2500 0 2500 5000 7500 10000 km

ANEXO 13. Código para el cálculo de la gravedad en superficie

```

%%ICGEM
%%FATIMA SAIZ ALVARO
%% CALCULO DEL GEOPOTENCIAL POR ARMONICOS ESFÉRICOS CON DATOS DEL SATELITE GRACE2016
function [gradw] = ICGEM(latitud, longitud, altitud)
%% LATITUD, LONGITUD, ALTITUD: coordenadas geocentricas esfericas del punto
%% R: radio de referencia
%% GM:cte gravitacional y masa de la Tierra
%% l,m: grado y orden de los armonicos esféricos
%% P: polinomio de Legendre
%% C,S: coeficientes normalizados de Stoke (Sat GRACE 2016)

format long;

%% Abrir los excel de los coeficientes C y S
%fprintf('Calculo del geopotencial gravitatorio a partir de coeficientes ICGEM (Stoke,
armónicos esféricos)\n');
c=xlsread('c_todo.xlsx');
s=xlsread('s_todo.xlsx');

%%Conversion de coordenadas elipsoidales a esfericas
%De elipsoidal a geocentrica

lat=latitud*pi/180;
long=longitud*pi/180;

a=6378137.0 ;
alfa=1/298.257223563;
b=a*(1-alfa);
e2=(2*alfa)-(alfa^2);

u=sin(lat);
w=(1-e2*u^2)^(0.5);
N=a/w;
M=(a*(1-e2))/(1-e2*u^2)^(3/2);

X=(N+altitud)*cos(lat)*cos(long);
Y=(N+altitud)*cos(lat)*sin(long);
Z=((b^2/a^2*N)+altitud)*sin(lat);

%De geocentrica a esferica
r=((X^2)+(Y^2)+(Z^2))^(0.5);
acimut=acosd(Z/r);
  
```

```
landa=atan2d(Y,X);

l=acimut;
lon= landa+360;
alt=r;

%Control de errores
if (l<-90||l>90);
fprintf('Latitud incorrecta, introduce un valor entre 0 y 90')
end;

if (lon<0||lon>360);
fprintf('Longitud incorrecta, introduce un valor entre 0 y 360')
end;

%% Constantes
GM= 398600441800000;
%% m3/s2
R= 6378137.0;
%% m

%% Paso a radianes
lr=l*((2*pi())/360);
lonr=lon*((2*pi())/360);

% Se calcula la variable X como seno de la latitud
X=sin(lr);

n=160;
m=160;

%Se crea una matriz de ceros donde se almacenarán los resultados de los
%polinomios asociados de Legendre
P=zeros(n+2,m+1);
PN=zeros(n+2,m+1);
DP=zeros(n+2,m+1);
DPN=zeros(n+2,m+1);

%Cálculo de los polinomios de Legendre totalmente normalizados
k=1;
for k=1:n+1;
    for n=1:161;

        for m=1:n+1;

                %% m=0 en la realidad, en el programa es m=1
```

```

    P(1:(k+1),k)=(legendre(k,X));

    DP(1:(k+1),k)=(legendre_derivative(k,X));

%Mediante la funcion legendre_derivative se calculan las derivadas de
%los polinomios asiados de legendre y se almacenan en DP

    end;
  end;
end;

%%Normalización de los polinomios: totalmente normalizados (Geodesia)
  for n=1:161;
    for m=1:n+1;
      if m==1;
        PN(m,n)=((1*(2*n+1)*(factorial(n-(m-1))/factorial(n+(m-1))))^(1/2))*P(m,n);
        DPN(m,n)=((1*(2*n+1)*(factorial(n-(m-1))/factorial(n+(m-1))))^(1/2))*DP(m,n);
      else;
        PN(m,n)=((2*(2*n+1)*(factorial(n-(m-1))/factorial(n+(m-1))))^(1/2))*P(m,n);
        DPN(m,n)=((2*(2*n+1)*(factorial(n-(m-1))/factorial(n+(m-1))))^(1/2))*DP(m,n);
      end;
    end;
  end;

%Se calculan las traspuestas para que n sean las filas y m las columnas.
PN=PN';
DPN=DPN';

mwr=zeros(10000,1);
mwlon=zeros(10000,1);
mwlat=zeros(10000,1);
cont=1;

%%Cálculo de los armónicos esféricos
for n=0:160;
  for m=0:n,1;
    if n==0 && m==0;

      wr=((R/r)^n)*(n+1)*(((c(cont,1)*cos(m*lonr))+s(cont,1)*sin(m*lonr)));
      wlon=((R/r)^n)*(((s(cont,1)*cos(m*lonr))-(c(cont,1)*sin(m*lonr)))*m);
      wlat=((R/r)^n)*(((c(cont,1)*cos(m*lonr))+s(cont,1)*sin(m*lonr)));
      mwr(cont,1)=wr;
      mwlon(cont,1)=wlon;
    end;
  end;
end;

```

```

    mwlát(cont,1)=wlat;

else;

    cont=cont+1;

wr=((R/r)^n)*(n+1)*(((c(cont,1)*cos(m*lonr))+s(cont,1)*sin(m*lonr)))*(PN(n,m+1)));
wlon=((R/r)^n)*(((s(cont,1)*cos(m*lonr))-c(cont,1)*sin(m*lonr)))*(PN(n,m+1))*m);
wlat=((R/r)^n)*(((c(cont,1)*cos(m*lonr))+s(cont,1)*sin(m*lonr)))*(DPN(n,m+1)));
mwr(cont,1)=wr;
mwlon(cont,1)=wlon;
mwlat(cont,1)=wlat;

    end;
end;

sumawr=abs(sum(mwr));
sumawlon=abs(sum(mwlon));
sumawlat=abs(sum(mwlat));

wrtotal=(-GM/(r^2))*sumawr;
wlontotal=(GM/r)*sumawlon;
wlattotal=(GM/r)*sumawlat;

omega=7292115*10^-11;

%%METER R COMO ALTITUD PORQUE ES DEL PUNTO

or=(omega^2)*r*((cos(lr))^2);
olon=0;
olat=(-omega^2)*(r^2)*(cos(lr))*(sin(lr));

a1=((wrtotal + or)^2);
a2=((1/(r*cos(lr)))*(wlontotal + olon))^2;
a3= ((1/r)*(wlattotal + olat))^2;
gradw=sqrt( a1 + a2 + a3);

fprintf('G= %.4f (mgal)', gradw)

```

ANEXO 14. Correcciones de la gravedad

La primera corrección es la del **aire libre**, que tiene en cuenta el efecto de la altura h . El hecho de que se observe la gravedad alejada de la superficie del geode repercute en un valor de la gravedad más bajo que el que se observaría sobre el geode. Este problema se puede abordar mediante el teorema de Taylor, donde se calcula el valor de la función g_0 conociendo el valor de la gravedad en un punto cercano $P(g)$, quedando finalmente la ecuación:

$$g_0 = g_p + \frac{2GM}{l^3}h$$

Ecuación 14 Gravedad en un punto cercano

Siendo g_0 el valor calculado de la gravedad sobre el geode, g_p el valor observado de la gravedad y h la altitud de P . Sustituyendo valores se obtiene que:

$$C_{AL} = -0.3086 \text{ mgal/m}$$

Ecuación 15 Corrección aire libre

El signo menos proviene del hecho que al aumentar R disminuye g , entonces la corrección será aditiva.

La segunda corrección es la corrección **Bouguer**. Una vez efectuada la corrección aire libre se obtiene el valor de la gravedad sobre el geode, pero interfiere la masa de la capa que se sitúa sobre el mismo. En este caso la corrección tiene signo negativo, ya que la masa que se halla por debajo del punto ejerce una fuerza extra, que se suma a la ejercida por la masa del geode. (Universidad politécnica de Valencia, 2015)

La corrección se obtiene considerando la influencia generada por un cilindro con radio infinito que hace la función de la lámina plana:

$$C_B = 2\pi G\delta h = 0.04193 \delta \text{ mgal/m}$$

Ecuación 16 Corrección Bouguer

Donde δ es la densidad de la placa Bouguer, que normalmente se considera 2.67 Tn/m^3 .

La siguiente es la **corrección topográfica**. Al igual que las masas existentes entre el punto y el geode tienen influencia sobre la gravedad medida en P , el resto de masas que se hallan alrededor del punto sobre el cual se mide la gravedad ejercerán también influencia sobre esta medida. Estas disminuyen el valor medido, por tanto siempre será una corrección positiva.

El cálculo se desarrollará digitalizando el relieve en forma de una rejilla ortogonal (vea Figura abajo), se calcula la gravedad de cada uno de las prismas, y se suman todos los efectos de los prismas. Usando este método, no se tiene que determinar las alturas medias de nuevo para cada estación. Para determinar el efecto gravimétrico de un prisma se utilizará la siguiente ecuación:

$$C_T = 2 G \delta b^2 \frac{h^2}{r^3} \left(1 + \frac{3b^2}{2r^2}\right)$$

Ecuación 17 Corrección topográfica

Donde G es la constante gravitacional en el sistema métrico internacional, h la diferencia de altura entre el prisma y la altura de la estación, δ la densidad de las masas topográficas y b y r se muestran en la imagen a continuación.

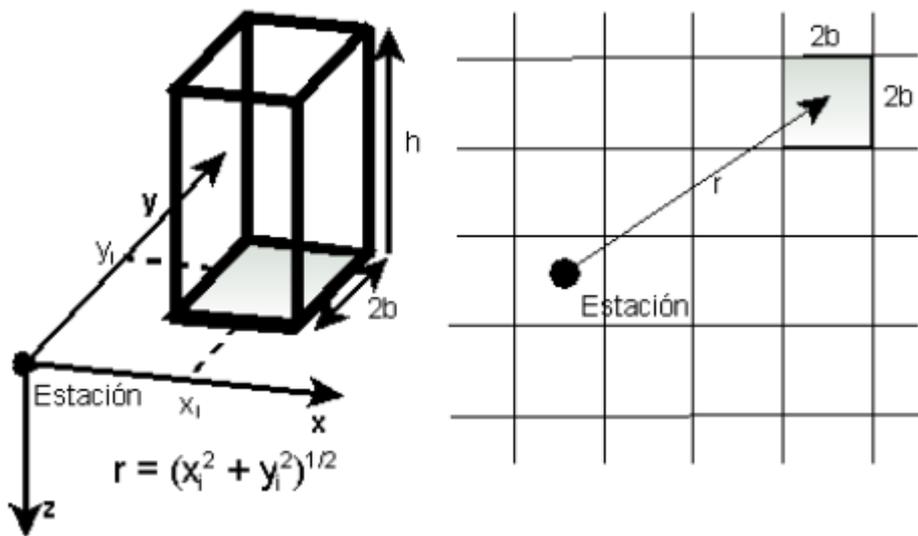


Ilustración 33 Corrección topográfica

Por último, se calcula la corrección **isostática**. Puede ser corregido por dos teorías, una según Pratt-Hayford y otra por Airy-Heiskanen. Se ha programado la corrección de Airy-Heiskanen en este caso.

Se considera una corteza con una densidad homogénea de 2.67 g/cm³. La compensación isostática para áreas montañosas se consigue por la formación de raíces de la corteza en el manto superior de densidad 3.27 g/cm³. De esta forma, en estas raíces existirá una deficiencia de masa que viene caracterizada por una diferencia de densidades: $\Delta\rho = \rho_M - \rho_c = 3.27 - 2.67 = 0.6 \text{ g/cm}^3$.

Se considera un punto P situado a una altura h del Geoide sobre una elevación de forma cilíndrica cuya sección es S. En la teoría de Airy-Heiskanen se considera una corteza de grosor T sobre un manto superior homogéneo de densidad $\rho_m = 3.27 \text{ g/cm}^3$. La corteza irá cambiando de grosor dependiendo de las áreas montañosas y de las zonas oceánicas. (Universidad de Las Palmas de Gran Canaria, 2015)

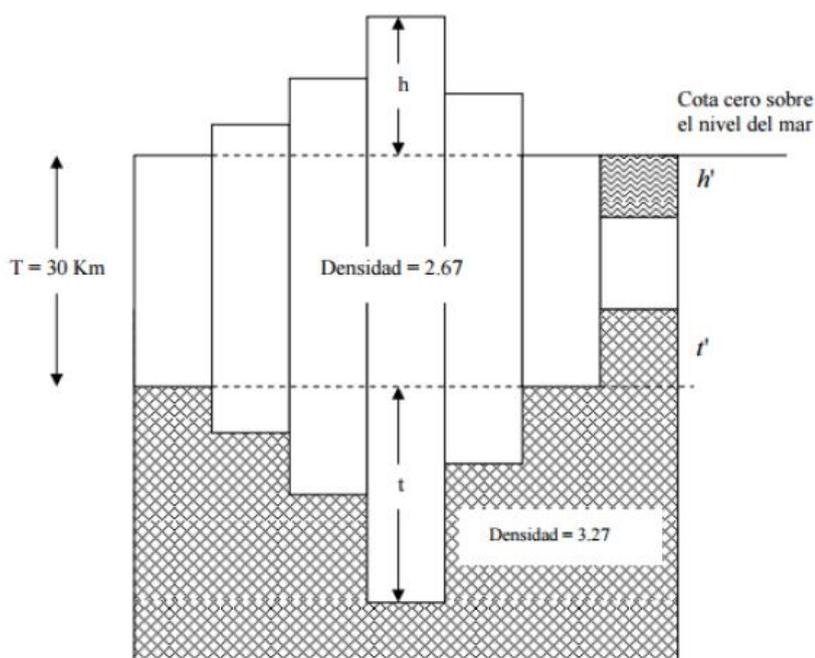


Ilustración 34 Teoría Airy-Heiskanen corrección isostática

Si se designa con h la altitud de la topografía y con t el espesor correspondiente a la raíz, la condición de equilibrio flotante que la hidrodinámica nos proporciona como el efecto del empuje de Arquímedes sobre un medio más denso se transforma en:

$$t\Delta\rho = h\rho_0$$

Ecuación 18 Corrección isostática

El espesor normal de la corteza terrestre se designa por T y se suele expresar como 30 Km . (aproximadamente la profundidad de la discontinuidad de Mohorovicic).

ANEXO 15. Funcionamiento del programa de correcciones gravimétricas

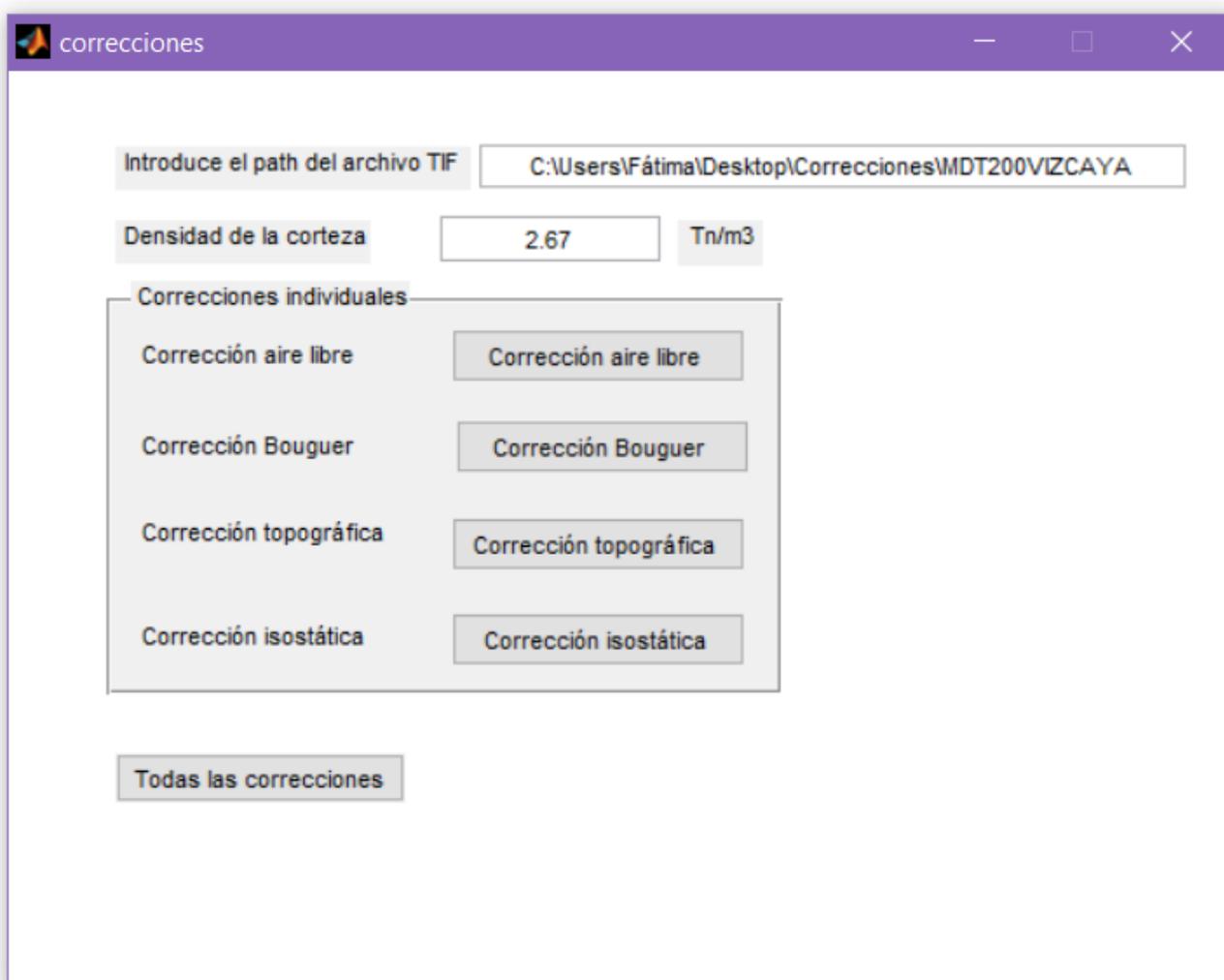


Ilustración 35 Programa correcciones gravimétricas

El funcionamiento de este programa es el siguiente:

Se introduce el path en el que se encuentra el MDT del que se quieren realizar los cálculos. El programa introduce la extensión .tif al fichero cargado.

Se introduce la densidad de la corteza que se desea tomar. Se pueden calcular las correcciones individualmente o efectuar todas las correcciones a la vez.

Los ficheros de salida son un fichero Excel con las correcciones en una matriz que posee el mismo orden que los pixels del TIF original y un fichero de texto en el que se incluyen las coordenadas X e Y y su corrección, que es apto para exportar en cualquier programa GIS.

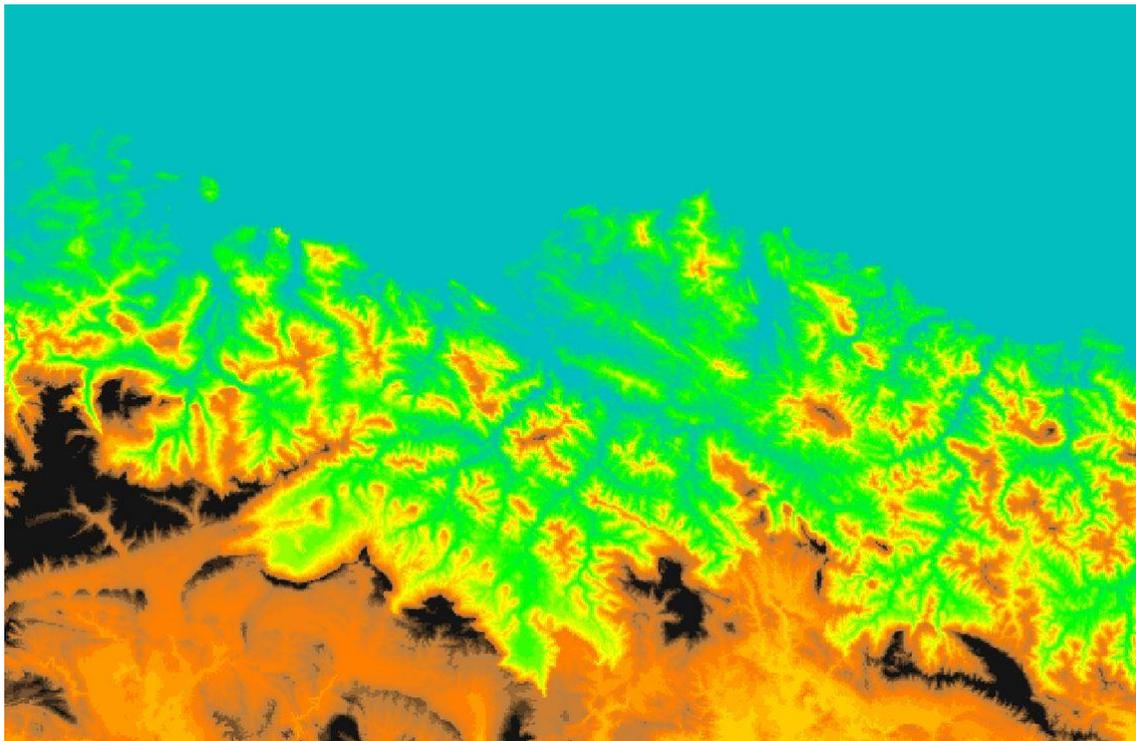


Ilustración 37 Datos de entrada. MDT Bizkaia.

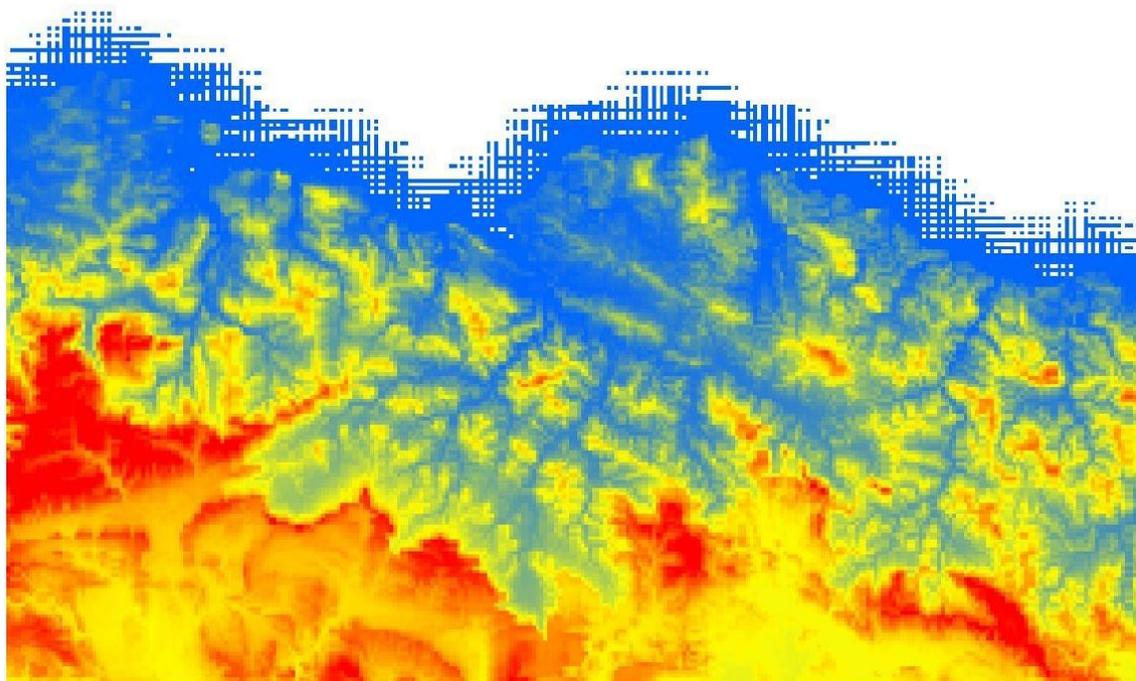


Ilustración 36 Datos de salida. Correcciones en el MDT de Bizkaia.

En el cálculo de prueba, se ha introducido un MDT de Bizkaia, con un paso de malla de 200m.

A partir del valor de la altura obtenida a través del MDT, se efectúan las correcciones mencionadas antes. Como se aprecia en las ecuaciones, todas las correcciones dependen de la altura de los puntos. Se crea un mapa de las correcciones según su magnitud.

Como se puede apreciar, las correcciones se corresponden totalmente con las elevaciones del terreno, en especial en las zonas de cambio. Los valores que se muestran en blanco sin datos se corresponden a zonas marítimas.

ANEXO 16. Valoración económica

Meter mano obra, explicar las horas productivas por cursos y gestión

Gastos Generales

GASTOS FIJOS	Mes	Año
<i>Alquiler</i>	400	4800
<i>Gastos alquiler</i>	40	480
<i>Asesoría</i>	30	360
<i>Seguro oficina</i>	30	360
<i>Seguro civil</i>	20	240
<i>Internet</i>	35	420
<i>Ordenador</i>	11.21	134.48
<i>Impresora</i>	3.34	40.14
<i>Cámara fotográfica</i>	2.0625	24.75
<i>Mobiliario</i>	12.50	150.00
<i>Consumibles y material oficina</i>	50.00	600
<i>Formación</i>	41.67	500
<i>Microsoft Office</i>	19.58	235
<i>Matlab</i>	55.01	660.07
<i>Qgis</i>	0	0
HORAS PRODUCTIVAS		1440
TOTAL COSTE AÑO		9004.43
TOTAL COSTE HORA		6.25

Tabla 3 Gastos fijos

Cálculo amortizaciones

CALCULO AMORTIZACIONES			
	Precio	Años uso	Amortización anual
<i>Ordenador</i>	537.90	4.00	134.48
<i>Impresora</i>	267.73	6.67	40.14
<i>Mobiliario</i>	1500.00	10.00	150.00
<i>Matlab</i>	2000.00	3.03	660.06601
<i>Microsoft Office</i>	235	1	235
<i>Cámara</i>	75.00	3.03	24.75

Tabla 4 Amortizaciones

Gastos de asignación directa

Gastos de asignación directa caja de arena

Nombre	Precio (€)	Detalles
<i>Kinect</i>	37.95	Kinect Xbox 360 Modelo: 14001
<i>Ordenador</i>	537.9	Acer Aspire E15 - Portátil de 15,6 " HD (Intel Core i5-7200U, 8 GB RAM, 1 TB HDD, Nvidia GTX 950M de 2 GB, Linux); Negro - Teclado QWERTY español
<i>Arena</i>	30	Arena fina de sílice blanca
<i>Estructura</i>	40	Madera y perfiles metálicos
TOTAL	645.85	

Tabla 5 Gastos de asignación directa caja de arena

Gastos de asignación directa cálculos geofísicos

Nombre	Precio (€)	Detalles
<i>Matlab</i>	2000.00	Licencia para el uso del cliente
TOTAL	2000.00	

Tabla 6 Gastos de asignación directa geofísica

Gastos de asignación directa reconocimiento de matrículas

Nombre	Precio (€)	Amortización horas trabajo	Precio/h (€)
<i>Eclipse</i>	0	1620	0
<i>OpenCV</i>	0	1620	0
TOTAL			0.00

Tabla 7 Gastos de asignación directa reconocimiento de matrículas

Total por proyecto

CAJA DE ARENA			€/hora	nº horas	Total
<i>Mano de obra</i>	mo	Tabla	20.87	80	1669.44
<i>Gastos Generales</i>	gg	Tabla	6.25	80	500.25
<i>Gastos de asignación directa</i>	gv	Tabla			645.85
SUBTOTAL	ST	mo + gg + gv			2815.54
IVA	IVA	21 % ST			591.26
TOTAL CON IVA	TI	ST + IVA			3406.80

Tabla 8 Total caja de arena topográfica

GEOFISICA			€/hora	nº horas	Total
<i>Mano de obra</i>	mo	Tabla	20.87	200	4173.61
<i>Gastos Generales</i>	gg	Tabla	6.25	200	1250.62
<i>Gastos de asignación directa</i>	gv	Tabla	2000.00		2000.00
SUBTOTAL	ST	mo + gg + gv			7424.23
IVA	IVA	21 % ST			1559.09
TOTAL CON IVA	TI	ST + IVA			8983.31

Tabla 9 Total cálculos geofísicos

MATRICULAS			€/hora	nº horas	Total
<i>Mano de obra</i>	mo	Tabla	20.87	240	5008.33
<i>Gastos Generales</i>	gg	Tabla	6.25	240	1500.74
<i>Gastos de asignación directa</i>	gv	Tabla	0.00	240	0.00
SUBTOTAL	ST	mo + gg + gv			6509.07
IVA	IVA	21 % ST			1366.91
TOTAL CON IVA	TI	ST + IVA			7875.98

Tabla 10 Total reconocimiento de matrículas