



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN

“REGISTRADO Y FUSIÓN DE INFORMACIÓN”

TESINA DE SEMINARIO

Previo a la obtención del título de:

INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

PRESENTADA POR:

Mildred Lizbeth Cruz Tomalá

GUAYAQUIL – ECUADOR

2015

AGRADECIMIENTO

A Dios, por mantener firme mi fe, por darme
paciencia, salud y perseverancia para poder
culminar exitosamente este proyecto.

A mi familia, que es el pilar fundamental de mi
vida.

Un agradecimiento especial a mis sobrinos, por
siempre contagiarme con su alegría y su amor
inmedible.

Al PHD Ángel D. Sappa, Director del presente
proyecto, por su permanente predisposición de
colaboración prestada en mi trabajo.

DEDICATORIA

A mis padres, por el sacrificio que han hecho para que pueda culminar mi carrera profesional.

Por su constante comprensión durante todo este trayecto de mi vida.

A mis amigos, por su incondicional apoyo, motivación y por todas las maravillosas experiencias que hemos compartido juntos.

A todas las personas que creyeron en mí y que aportaron de una u otra manera con sus consejos y vivencias para mi formación humana.

TRIBUNAL DE SUSTENTACIÓN



Ph.D. Angel Sappa
PROFESOR DEL SEMINARIO DE GRADUACIÓN

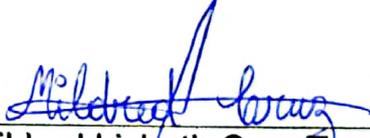


Ph.D. Boris Vintimilla
PROFESOR DELEGADO POR EL DECANO

DECLARACIÓN EXPRESA

"La responsabilidad por los hechos ideas y doctrinas expuestas en esta tesina me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Reglamento de exámenes y títulos profesionales de la ESPOL)


Mildred Lizbeth Cruz Tomalá

RESUMEN

El presente trabajo estará enfocado al registrado y fusión de información multispectral, ya que no solo se trabajará con imágenes visibles sino también con imágenes infrarrojas.

Para realizar el registrado de información, existen varios algoritmos detectores y descriptores de puntos característicos que son utilizados en distintos dominios (en general en imágenes visibles del espacio 2D y 3D) para llevar a cabo dicha función. Sin embargo, al ejecutar dichos algoritmos en imágenes infrarrojas, los puntos obtenidos no son los mismos que fueron hallados en las imágenes visibles, por ende al realizar la correspondencia entre los puntos de cada imagen, éstos no concuerdan totalmente.

Por este motivo, no resulta muy favorable aplicar estos algoritmos, ya que se presentarían muchos inconvenientes al momento de fusionar las imágenes, por lo que se ha optado por un proceso híbrido en el cual la selección de los puntos en cada imagen se la hará de manera manual, mientras que la puesta en correspondencia de la información multispectral se realizará en forma automática utilizando el algoritmo ICP.

El uso del algoritmo ICP básicamente consiste en encontrar la matriz de rotación y el vector de traslación necesarias para hacer la correcta correspondencia entre los puntos de las imágenes visibles e infrarrojas.

Finalmente, fusionando la información de las imágenes multiespectrales, se intentará encontrar índices que permitan distinguir características u objetos para así emular un resultado obtenido por otros índices ya conocidos como el NDVI, RVI, entre otros.

Para el desarrollo de la aplicación, se ha procedido a utilizar la librería OpenCV y el IDE QtCreator utilizando el lenguaje C++.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XIV
ABREVIATURAS	XV
INTRODUCCIÓN.....	XVI
CAPÍTULO 1	1
ANÁLISIS DEL CASO.....	1
1.1. Análisis del problema	1
1.2. Diseño de la solución	4
1.3. Objetivos del proyecto	4
1.3.1. Objetivo general.....	4
1.3.2. Objetivos específicos	5
1.4. Justificación	5
CAPÍTULO 2.....	6
REVISIÓN BIBLIOGRÁFICA	6

2.1.	Registrado de información multiespectral.....	6
2.2.	Iterative Closest Point (ICP)	7
2.2.1.	Inicialización o selección de las nubes de puntos.....	10
2.2.2.	Correspondencia de un punto.....	11
2.2.3.	Ponderación a los pares correspondientes.....	13
2.2.4.	Rechazo de pares para eliminar valores atípicos	13
2.2.5.	Asignación y minimización de la métrica de error	15
2.3.	Fusión de información multiespectral	16
2.3.1.	NDVI	17
2.3.2.	RVI.....	19
2.3.3.	ENDVI.....	20
CAPÍTULO 3.....		23
INSTALACIÓN Y CONFIGURACIÓN DE LA APLICACIÓN		23
3.1.	Herramientas de desarrollo	23
3.2.	Instalación y configuración de la aplicación.....	27
CAPÍTULO 4.....		35
DISEÑO Y DESARROLLO DEL SISTEMA EXPERIMENTAL		35
4.1.	Diseño del sistema (GUI).....	35
4.1.1.	Módulo de archivo.....	38
4.1.2.	Módulo alineación de imágenes.....	44
4.1.3.	Módulo fusión de información multiespectral	46
4.1.4.	Ayuda.....	48

4.2. Desarrollo del sistema (ICP).....	48
CAPÍTULO 5.....	54
RESULTADOS EXPERIMENTALES	54
5.1. Resumen	54
5.2. Metodología de las pruebas	55
5.3. Resultados.....	56
5.3.1. Algoritmo ICP.....	56
5.3.2. Índices de detección de vegetación	59
5.3.3. Índices de detección de cielo.....	71
5.3.4. Índices de detección de calle.....	75
CONCLUSIONES	78
RECOMENDACIONES.....	80
ANEXOS.....	81
Anexo 1.....	82
Glosario	84
Bibliografía.....	88

ÍNDICE DE FIGURAS

Figura 1.1: Aplicación del algoritmo de Harris en imágenes LWIR vs. VS.....	2
Figura 1.2: Alineación de imágenes con diferentes perspectivas.	3
Figura 1.3: Imagen LWIR vs VS.....	3
Figura 2.1: Modelo ideal del ICP.....	8
Figura 2.2: Etapas del ICP.....	9
Figura 2.3: (a) Muestreo Uniforme. (b) Muestreo en el espacio de las normales.	10
Figura 2.4: (a) Punto más cercano. (b) “Normal shooting”. (c) Proyección de un punto.....	11
Figura 2.5: (a) Distancia comparada con el umbral. (b) Puntos sobre vértices finales. (c) Pares inconsistentes.	14
Figura 2.6: Métricas de error.....	16
Figura 2.7: Aplicación de Índices de Vegetación. (a) Imagen VS. (b) Imagen NIR. (c) NDVI. (d) RVI. (e) ENDVI.	21
Figura 2.8: Paleta de colores utilizada para el mapeo de los índices de vegetación.	22
Figura 3.1: Configuración de la variable de entorno.	28
Figura 3.2: Instalación de Qwt(1).	30
Figura 3.3: Instalación de Qwt(2).	30
Figura 3.4: Instalación de los ejemplos de Qwt.	31
Figura 3.5: Project Manager de la aplicación.....	32

Figura 3.6: Configuraciones de OpenCV y Boost.	33
Figura 3.7: Configuración de la librería QCustomPlot.....	34
Figura 4.1: Distribución del código del proyecto.	36
Figura 4.2: Interfaz principal del sistema.	36
Figura 4.3: Menú del sistema.....	38
Figura 4.4: Opción Abrir Imagen. (a) LWIR. (b) VS.....	39
Figura 4.5: Cuadro de diálogo con las imágenes existentes en el PC.	39
Figura 4.6: Carga de imágenes LWIR y VS.	40
Figura 4.7: Error #1 en opción Guardar los puntos.....	41
Figura 4.8: Error #2 en opción Guardar los puntos.....	42
Figura 4.9: Error #3 en opción Guardar los puntos.....	42
Figura 4.10: Selección de puntos de interés.....	43
Figura 4.11: Gráfico posicional de puntos de interés.....	43
Figura 4.12: Iniciar ICP iterativo.....	44
Figura 4.13: Proceso iterativo del ICP.	45
Figura 4.14: Fin del algoritmo ICP.	46
Figura 4.15: Interfaz de la Fusión Multiespectral.	47
Figura 4.16: Interfaz de la Ayuda del Sistema.	48
Figura 4.17: Código del constructor del IcpPointToPoint.....	50
Figura 4.18: Código del constructor Icp.....	51
Figura 4.19: Código para crear un árbol kd.	51
Figura 4.20: Código para inicializar la transformación.	52

Figura 5.1: Fase de selección de 20 parejas de puntos.....	57
Figura 5.2: Ejecución del algoritmo ICP con 20 parejas de puntos.....	58
Figura 5.3: Resultados de errores totales del algoritmo ICP.....	59
Figura 5.4: Fusión multiespectral (LWIR-VS) con el índice NDVI.	61
Figura 5.5: Fusión multiespectral (LWIR-VS) con el índice RVI.....	62
Figura 5.6: Fusión multiespectral (LWIR-VS) con el índice ENDVI.....	63
Figura 5.7: Fusión multiespectral (LWIR-VS) con el índice LWIR-IV1.	65
Figura 5.8: Fusión multiespectral (LWIR-VS) con el índice LWIR-IV2.	66
Figura 5.9: Fusión multiespectral (LWIR-VS) con el índice LWIR-IV3.	67
Figura 5.7: Fusión multiespectral (LWIR-VS) con el índice LWIR-IV4.	68
Figura 5.8: Fusión multiespectral (LWIR-VS) con el índice LWIR-IC1.....	71
Figura 5.9: Fusión multiespectral (LWIR-VS) con el índice LWIR-IC2.....	72
Figura 5.10: Fusión multiespectral (LWIR-VS) con el índice LWIR-IC3.	72
Figura 5.11: Fusión multiespectral (LWIR-VS) con el índice LWIR-IC4.	73
Figura 5.12: Fusión multiespectral (LWIR-VS) con el índice LWIR-ICA1.....	76
Figura 5.13: Fusión multiespectral (LWIR-VS) con el índice LWIR-ICA2.....	76
Figura A.1: Métodos de Fusión con el índice NDVI.	82
Figura A.2: Métodos de Fusión con el índice RVI.....	83
Figura A.3: Métodos de Fusión con el índice ENDVI.....	83
Figura G.1: Malla triangular.....	84
Figura G.2: Neighbor Search.	85
Figura G.3: Kd-tree.	86

ÍNDICE DE TABLAS

Tabla 1: Rangos de los índices de vegetación.....	69
Tabla 2: Rangos de detección de vegetación.	70
Tabla 3: Rangos de los índices de detección de cielo.	74
Tabla 4: Rangos de detección de cielo.	74
Tabla 5: Rangos de los índices de detección de calle.	77

ABREVIATURAS

VS	Visible Spectrum
LWIR	Long Wavelength Infrared
NIR	Near Infrared
NDVI	Normalized Difference Vegetation Index
RVI	Ratio Vegetation Index
ENDVI	Enhanced Normalized Difference Vegetation Index
LWIR-IV	Nuevo índice de vegetación
LWIR-IC	Nuevo índice de cielo
LWIR-ICA	Nuevo índice de calle

INTRODUCCIÓN

Uno de los campos más investigados en la rama de Visión por Computador es el que se refiere a la detección de características o encontrar algún patrón que permita localizar objetos comunes en todas las imágenes.

La mayoría de estudios realizados en este ámbito trabajan únicamente con imágenes del mismo espectro, en general visibles, logrando así por ejemplo la detección y el reconocimiento de rostros que se encuentren en las imágenes.

Por el contrario de lo mencionado anteriormente, en éste estudio se utilizarán imágenes visibles e infrarrojas que serán analizadas para encontrar una manera adecuada de fusionar la información de intensidad de cada una y así poder obtener índices que puedan ser generalizados a las imágenes multiespectrales.

Éste documento contiene 5 capítulos, cada uno con temas puntuales que serán brevemente explicados a continuación:

En el primer capítulo se describe cuáles son los objetivos de éste trabajo. Además, por medio de la aplicación implementada, se podrá explicar por qué la elección manual de los puntos facilita el trabajo para el algoritmo

ICP—*Iterative Closest Point*— [1] que posteriormente servirá para obtener índices de fusión con información más precisa.

En el segundo capítulo se presentará la información más relevante de la Literatura referente a los algoritmos utilizados en este proceso. A continuación, tanto el tercer como el cuarto capítulo están enfocados a la descripción de la aplicación desarrollada para realizar las diferentes pruebas con las imágenes.

Para terminar, el último capítulo analiza los resultados obtenidos de varias comparaciones que contribuyeron al hallazgo de conclusiones y recomendaciones para futuros trabajos.

CAPÍTULO 1

ANÁLISIS DEL CASO

1.1. Análisis del problema

Actualmente existen varias formas en las que se puede realizar la detección de puntos característicos. La mayor dificultad de estas, es que sólo han sido aplicados en imágenes del mismo espectro, en general imágenes del espectro visible.

En el presente trabajo se trabajará con imágenes multiespectrales, más en particular con imágenes visibles (VS) e imágenes infrarrojas de onda larga (LWIR), por lo que no es conveniente ejecutar algoritmos de detección de características que solo sean eficaces con imágenes VS. Por ejemplo, como muestra la Figura 1.1, si aplicamos el algoritmo de Harris a una imagen

LWIR, los puntos hallados no son los mismos que los encontrados en la imagen visible.

Probablemente esto podría ser mejorado modificando el valor de los parámetros que son requeridos para aplicar dicho algoritmo y a su vez que varíen para cada tipo de imagen, pero el principal inconveniente sería el desconocer cuáles deberían ser dichos valores para que los puntos detectados en ambas imágenes sean los mismos, teniendo en cuenta que tanto la imagen LWIR como la VS captan la misma escena.

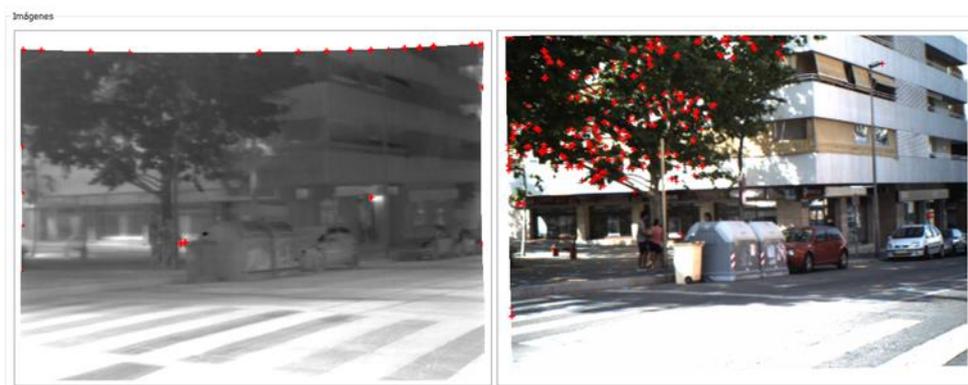


Figura 1.1: Aplicación del algoritmo de Harris en imágenes LWIR vs. VS.

Si por el contrario, las imágenes son tomadas desde diferentes perspectivas, surge otra dificultad, ya que es necesario encontrar la respectiva transformación para alinear correctamente ambas imágenes como se observa en la Figura 1.2.

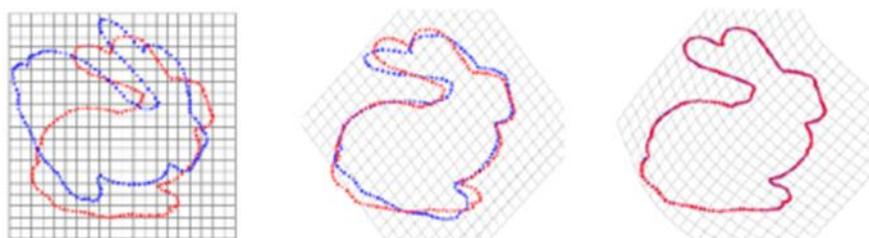


Figura 1.2: Alineación de imágenes con diferentes perspectivas.

De acuerdo a lo mencionado anteriormente, la mayoría de investigaciones realizadas han estudiado el caso en el que las imágenes pertenecen al mismo espectro, por ende no se tiene demasiados trabajos sobre la fusión multiespectral, refiriéndonos específicamente a fusionar la información visible junto a la infrarroja, ya que la información visible dista mucho de la información infrarroja como se puede observar en la Figura 1.3, por lo que no resulta tan fácil definir algún esquema general para detectar características u objetos.



Figura 1.3: Imagen LWIR vs VS.

1.2. Diseño de la solución

Dada la naturaleza multispectral de las imágenes a procesar, la selección de puntos característicos se realizará de forma manual, completando así una de las etapas del registrado de información multispectral.

Luego, se aplicará el algoritmo ICP con los puntos elegidos en cada imagen. De esta manera se procede a alinear ambas imágenes.

Finalmente, se tomarán de referencia índices ya existentes para la fusión de imágenes, aunque dichos índices sólo trabajen con imágenes visibles e imágenes infrarrojas pero de onda corta denominadas NIR, nos servirán como punto de partida para poder encontrar alguna combinación que logre localizar ciertos objetos o características representativas de la escena.

1.3. Objetivos del proyecto

El objetivo general y los objetivos específicos de este proyecto son descritos a continuación:

1.3.1. Objetivo general

Desarrollar una aplicación que permita realizar el registrado y fusión en imágenes VS y LWIR, de tal manera que realizando varias pruebas, se puedan obtener índices eficaces y funcionales para la detección de objetos y características.

1.3.2. Objetivos específicos

- Mostrar paso a paso como trabaja el algoritmo ICP y exponer porque éste se convierte en una excelente opción para alinear imágenes teniendo únicamente como datos de entrada puntos comunes, es decir que describan al mismo objeto tanto en imágenes VS como en LWIR.
- Encontrar índices que identifiquen objetos relevantes a través del análisis de la información multiespectral fusionada.

1.4. Justificación

Actualmente existen varios trabajos que realizan un exhaustivo análisis de imágenes en diferentes espectros, con la finalidad de encontrar características u objetos o aplicando diversos algoritmos que permitan encontrar puntos característicos en dichas imágenes.

La motivación principal de este trabajo es hallar distintas metodologías para fusionar imágenes multiespectrales y de esta forma colaborar en el ámbito investigativo y que sea un punto de partida para el desarrollo de futuros trabajos.

CAPÍTULO 2

REVISIÓN BIBLIOGRÁFICA

Actualmente existen diferentes formas en llevar a cabo el proceso de registrado y fusión multiespectral. En este capítulo se realiza una breve descripción del estado del arte de los temas abordados en el desarrollo del presente trabajo.

2.1. Registrado de información multiespectral

En general, el proceso de registrado de información consiste en los siguientes pasos:

- **Detección de características:** implica la obtención de puntos de interés que localicen objetos relevantes o distintivos en una imagen, ya sea de manera manual o automática. Por ejemplo, los contornos, esquinas, líneas de intersección, etc.

- **Correspondencia de características:** Una vez que se tienen los puntos de interés en ambas imágenes, tanto en la imagen que servirá de modelo, como en la imagen donde se aplicará la transformación, se establecen los puntos que coinciden correctamente en dichas imágenes.
- **Estimación del modelo de transformación:** Se estiman los parámetros de la alineación de las imágenes, en donde se asocia cada punto de un grupo dado hacia el grupo de puntos que sirve de referencia.
- **Remuestreo y transformación de imagen:** El conjunto de datos es transformado por medio de las funciones de mapeo.

2.2. Iterative Closest Point (ICP)

El algoritmo ICP—*Iterative Closest Point*—se basa esencialmente en alinear geoméricamente dos nubes de puntos, una nube de puntos que será el “target”, mientras que la otra será el “source” de tal forma que se pueda minimizar la distancia entre cada par de puntos asociados en cada iteración.

El ICP es utilizado frecuentemente para el registrado de información, ya sea para reconstruir objetos en 2D y 3D a partir del análisis de un par de imágenes desde diferentes ángulos y perspectivas de una misma escena.

Es necesario tener como datos de entrada, las respectivas nubes de puntos: el “target” que se mantendrá fijo y el “source” que se transformará para

adaptarse lo mejor posible al modelo o “target” por medio de la ejecución repetitiva del algoritmo ICP, obteniendo finalmente las matrices de transformación pertinentes como datos de salida.

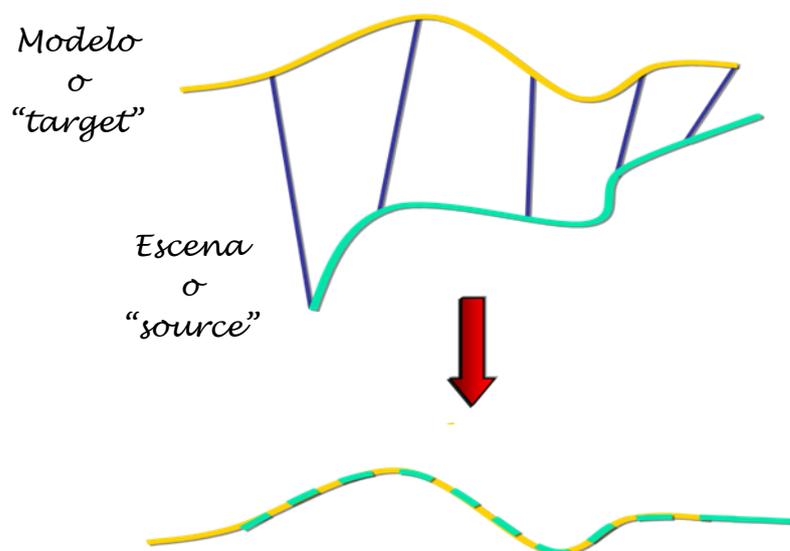


Figura 2.1: Modelo ideal del ICP.

Se supone que la transformación del “source” al “target” es lineal con una matriz de rotación R y un vector de traslación t .

El objetivo del algoritmo ICP es encontrar los parámetros de transformación, para que el error, en su mayoría estimado por el método de los mínimos cuadrados, entre los puntos transformados del “source” y los puntos más cercanos del modelo sea mínimo. Ésta característica es descrita en la siguiente ecuación:

$$\min_{R,t,j \in \{1,2,\dots,N_m\}} \left(\sum_{i=1}^{N_p} \|Rp_i + t - m_j\| \right)$$

Ecuación 1: Minimización del error por el método de mínimos cuadrados.

Donde:

$M \triangleq \{m_j\}_{j=1}^{N_m}$ nube “target” o modelo con N_m puntos

$P \triangleq \{p_i\}_{i=1}^{N_p}$ nube “source” con N_p puntos

R matriz de Rotación

t vector de traslación

De manera detallada, el ICP responde a las etapas o puntos mostrados en la siguiente figura:

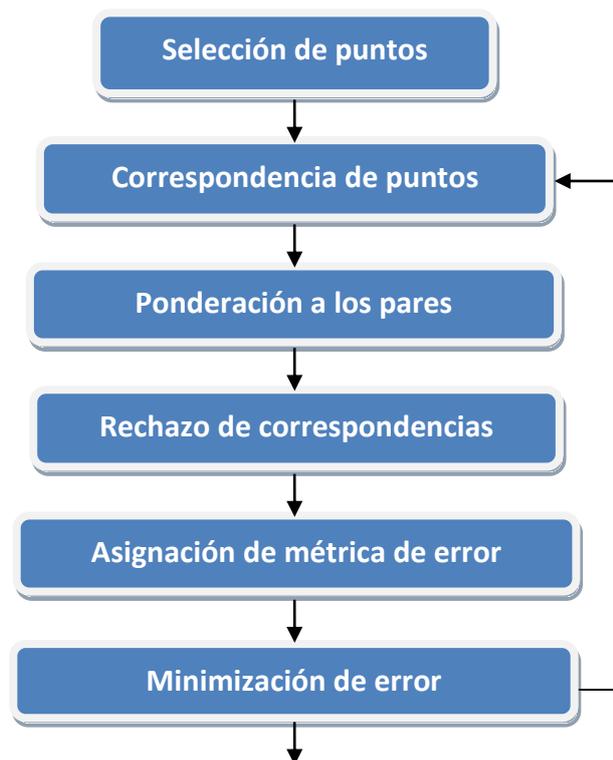


Figura 2.2: Etapas del ICP.

Seguidamente se presentan las distintas etapas del ICP detallando brevemente su funcionalidad:

2.2.1. Inicialización o selección de las nubes de puntos

La importancia de la inicialización reside en el hecho de que el algoritmo ICP converge únicamente a un mínimo local, por consiguiente aplicando buenos métodos de inicialización reduce la probabilidad de una correspondencia incorrecta.

Para la selección de puntos se pueden utilizar varios métodos o formas, entre las cuales se encuentran:

- Uso de todos los puntos disponibles [1].
- Submuestreo uniforme de los puntos disponibles [2].
- Sólo puntos de los bordes o nodos de una malla triangular [3].
- Muestreo aleatorio (con una muestra de puntos diferente en cada iteración) [4].
- Selección de puntos con gradiente de alta intensidad [5].
- Garantizar que las muestras tienen distribuidas las normales tan uniformemente como sea posible [6].

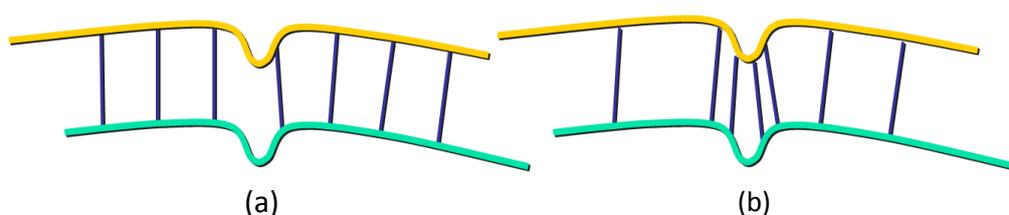


Figura 2.3: (a) Muestreo Uniforme. (b) Muestreo en el espacio de las normales.

2.2.2. Correspondencia de un punto

El propósito de este paso es encontrar el punto más cercano para cada punto transformado del “source”.

Esto puede ser formulado por la siguiente ecuación:

$$c(i) = \arg \min_{j \in \{1, 2, \dots, N_m\}} (||m_j - (Rp_i + t)||)$$

Ecuación 2: Correspondencia de un punto.

La correspondencia de puntos es el paso computacionalmente más caro del algoritmo ICP, pues necesita en el peor de los casos un tiempo de cálculo con una complejidad de $\mathcal{O}(N_m \cdot N_p)$, es decir realizando una búsqueda por fuerza bruta. Por esta razón, esta etapa frecuentemente sufre varias modificaciones que logran optimizarla.

El ICP propuesto por Besl y McKay [1] calcula la distancia de cada punto hacia el punto del otro conjunto para encontrar la asociación que corresponde al punto más cercano.

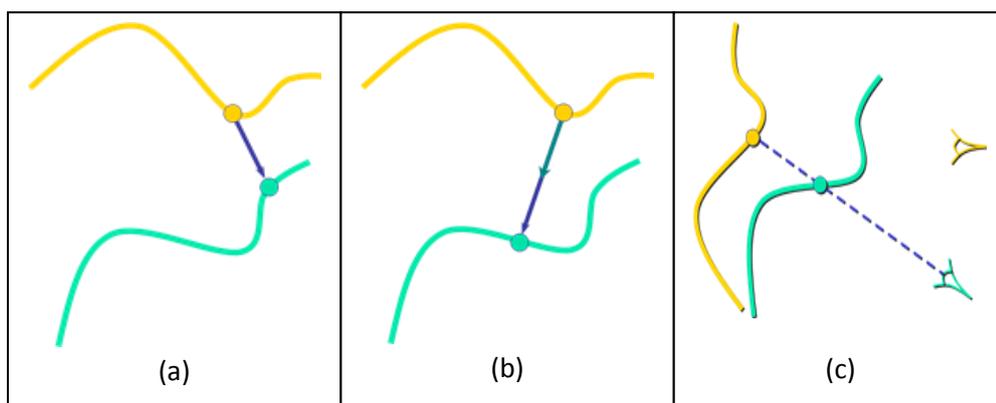


Figura 2.4: (a) Punto más cercano. (b) “Normal shooting”. (c) Proyección de un punto.

Chen en el 91 [10], propuso encontrar la intersección del rayo que se origina en el punto de origen en la dirección de la normal a la superficie/contorno que pasa por el punto destino. Esto se lo conoce como “normal shooting”.

Otra forma de llevar a cabo esta etapa es por medio de la proyección, también conocido como calibración reversa, método propuesto por Blais en 1995 [7][8], en donde se proyecta el punto de muestra sobre la malla destino desde el punto de vista de la cámara de la malla destino. No se requiere pre calcular datos espaciales estructurados y tiene un tiempo constante.

Una variante al método explicado anteriormente, es proyectar el punto sobre la imagen destino, luego realizar una búsqueda en el rango de la imagen de destino. La búsqueda podría usar una métrica basada en la distancia point-to-point, point-to-ray o compatibilidad de intensidad.

Actualmente, la búsqueda es implementada como un recorrido descendiente “neighbor-to-neighbor” en la malla de destino que intenta encontrar el punto más cercano.

En 2001 [11], el estado del arte es usar datos espaciales estructurados, como por ejemplo realizar una búsqueda a través de un kd-tree cuya complejidad es de $O(Nm \cdot \log Np)$, que resulta como una mejora a lo propuesto en [1].

2.2.3. Ponderación a los pares correspondientes

Se consideran diferentes algoritmos para la asignación de pesos:

- Peso constante.
- Asignación de pesos bajos a los pares con distancias point-to-point altas. Principalmente es elegido este método para eliminar aquellos pares cuya distancia point-to-point sea más grande que el valor de un umbral establecido. De acuerdo a [3] el valor del peso sería el siguiente:

$$Peso = 1 - \frac{Dist(p_1, p_2)}{Dist_{max}}$$

Ecuación 3: Asignación de pesos con distancias point-to-point.

- Asignación de pesos basado en la compatibilidad de las normales.

$$Peso = n1 \cdot n2$$

Ecuación 4: Asignación de pesos basado en la compatibilidad de normales.

- Pesos basados en la métrica del error esperado del sensor, calculado a través de la distancia point-to-plane.

2.2.4. Rechazo de pares para eliminar valores atípicos

La ejecución del paso anterior permite tener una mejor perspectiva acerca de esta etapa. El propósito es eliminar “outliers” o parejas que produzcan un gran efecto en el proceso de minimización.

A continuación, se listan algunas estrategias propuestas para este paso:

- Puntos correspondientes con una distancia *point-to-point* más alta que un umbral dado.
- Rechazo de parejas que posean puntos en los límites de las mallas, o sea pares que contengan puntos sobre los vértices finales. Este enfoque asume que la conectividad de los puntos de las imágenes es conocida.
- Rechazar el $n\%$ de las peores parejas. Usualmente la distancia entre los puntos es la métrica de determinación de la calidad de la pareja.
- Rechazar parejas que no sean consistentes con sus parejas vecinas. Este esquema clasifica dos parejas $(p1,p2)$ y $(q1,q2)$ como inconsistente si: $|(Dist(p1,p2) - Dist(q1,q2))|$ es mayor que un umbral [9].

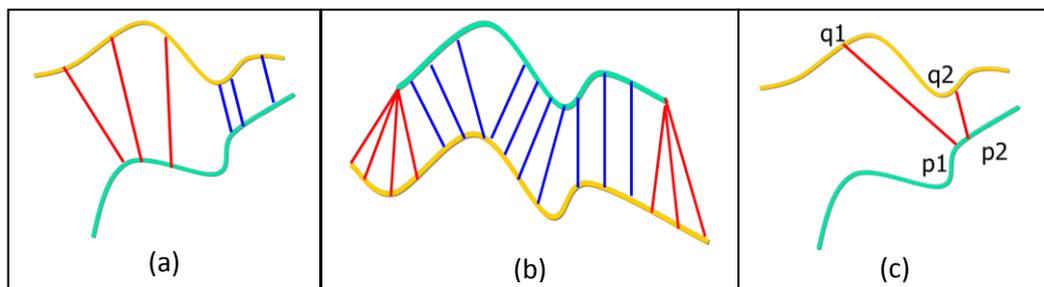


Figura 2.5: (a) Distancia comparada con el umbral. (b) Puntos sobre vértices finales. (c) Pares inconsistentes.

2.2.5. Asignación y minimización de la métrica de error

Es preciso seleccionar una métrica de error, cuyo objetivo es encontrar los nuevos parámetros de transformación en cada iteración, minimizando el error por medio del método de mínimos cuadrados [23]. Si bien existen distintas posibilidades, seguidamente se presentan los 2 métodos más usados:

El primer método utiliza como métrica a minimizar la distancia point-to-point entre los puntos correspondientes; siendo la función a minimizar la siguiente expresión:

$$(R, t) = \arg \min_{R, t} \left(\sum_{i=1}^{N_p} \|m_{c(i)} - (Rp_i + t)\|^2 \right)$$

Ecuación 5: Minimización de la distancia point-to-point.

Donde la ecuación del centro de masa para el vector de traslación más frecuentemente utilizada es:

$$t = \frac{1}{N_m} \sum_{i=1}^{N_m} m_{c(i)} - \frac{1}{N_p} \sum_{j=1}^{N_p} Rp_j$$

Ecuación 6: Centro de masa para el vector de traslación.

Cuaterniones, matrices ortonormales y descomposición de valores singulares (SVD) son soluciones que asimismo han sido propuestas.

Para una métrica de error de esta forma, existen soluciones “closed-form” para determinar la transformación de un cuerpo rígido que minimiza el error.

Otra métrica asociada a la anterior, se basa en no sólo tomar en cuenta la suma de cuadrados de las distancias entre puntos, sino que también considera la diferencia en colores.

El segundo método se basa en la distancia *point-to-plane*; en este caso los puntos de la nube modelo están conectados con la nube “source” por la normal a la tangente.

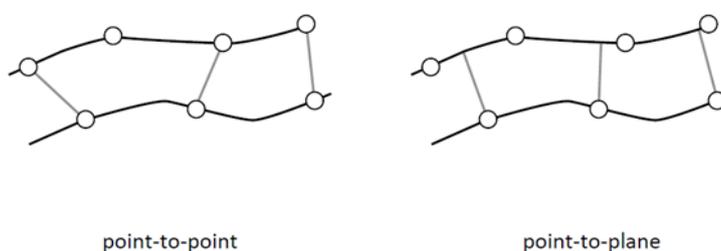


Figura 2.6: Métricas de error.

No hay una solución “*closed-form*”, por lo que éste problema debe ser resuelto a través de métodos no lineales genéricos o por linealización. La iteración dura hasta que el error está por debajo de un umbral [6].

2.3. Fusión de información multispectral

En los últimos años, se han venido desarrollando varios trabajos sobre técnicas que permitan combinar la información en una misma imagen de diferentes bandas espectrales.

La fusión de imágenes engloba a todos aquellos métodos utilizados con el objetivo de combinar o integrar varios tipos de imágenes de distintas características.

El presente trabajo estará enfocado a la combinación de bandas del espectro visible con el espectro infrarrojo lejano, de cara a obtener representaciones compactas con dicha información.

Como punto de partida, se estudiarán algunos índices de fusión que existen actualmente, pero que combinan la información NIR y visible de una misma escena. Como ejemplos de dichos índices, tenemos al NDVI (*Normalized Difference Vegetation Index*), ENDVI (*Enhanced Normalized Difference Vegetation Index*), RVI (*Relative Vigor Index*), entre otros, que son usados para detectar vegetación, tomando en cuenta que las plantas saludables reflejan mejor luz visible roja y NIR.

2.3.1. NDVI

El NDVI es un índice que proporciona una estimación de la salud de la vegetación, convirtiéndose en un medio para monitorear los cambios en la vegetación a través del tiempo.

Ésta fórmula tiene en cuenta la cantidad de infrarrojo reflejado por las plantas. Las plantas absorben radiación solar, que utilizan como fuente de energía en el proceso de la fotosíntesis.

Las hojas verdes tienen una reflectancia de 20% o menos en el rango de 0.5 a 0.7 μ (verde a rojo) y alrededor del 60% en el rango 0.7 a 1.3 μ (NIR).

Estas reflectancias espectrales son coeficientes propios de la radiación reflejada sobre la radiación entrante en cada banda espectral individualmente. Por lo tanto, el NDVI varía entre -1.0 y 1.0

Los valores negativos del índice que se acercan a -1 corresponden a aguas profundas, mientras que los valores que se acercan a 0, entre -0.1 a 0.1 por lo general corresponden a áreas de rocas, arena o nieve. Los valores positivos representan arbustos y pastizales, aproximadamente entre 0.2 a 0.4 y finalmente los valores cercanos a 1 representan selvas templadas y tropicales.

El rango típico se encuentra entre -0.1 para una zona no muy verde a 0.6 para una zona muy verde.

El NDVI se calcula de la siguiente forma:

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Ecuación 7: Índice NDVI.

Donde NIR corresponde a la imagen obtenida en el espectro infrarrojo cercano y RED corresponde al canal rojo de la imagen del espectro visible (asumiendo la imagen en el formato RGB, RED corresponde al canal R).

2.3.2. RVI

El RVI fue descrito originalmente por Birth y McVey en 1968 [19]. Se calcula simplemente dividiendo los valores de reflectancia de la banda del infrarrojo cercano por los de la banda roja.

$$RVI = \frac{NIR}{RED}$$

Ecuación 8: Índice RVI.

NIR y RED corresponden a las imágenes del espectro infrarrojo cercano y al canal rojo de la imagen del espectro visible como se indicó anteriormente.

El resultado capta claramente el contraste entre las bandas rojas e infrarrojas para los píxeles con vegetación, con altos valores de índice siendo producidos por combinaciones de baja reflectancia en la banda roja (debido a la absorción por la clorofila) y alta reflectancia en el infrarrojo (como resultado de la estructura de la hoja). Además, debido a

que el índice se construye como una relación, se reducen al mínimo los problemas de iluminación variable como resultado de la topografía.

Sin embargo, el índice es susceptible de errores de división por cero y la escala de medición resultante no es lineal.

2.3.3. ENDVI

Se pueden obtener mejores resultados si se utiliza el NIR y verde como canales reflexivos mientras se usa el azul como el canal de absorción. El ENDVI es la mejora de la fórmula del NDVI.

$$ENDVI = \frac{((NIR + GREEN) - (2 * BLUE))}{((NIR + GREEN) + (2 * BLUE))}$$

Ecuación 9: Índice ENDVI.

Donde NIR se refiere a la imagen del espectro infrarrojo cercano, mientras que GREEN y BLUE a los canales G y B de la imagen en el espectro visible.

En la Figura 2.7 se muestra el resultado de aplicar estos índices en la fusión multiespectral de una imagen VS de una escena con la correspondiente imagen NIR de la misma escena.

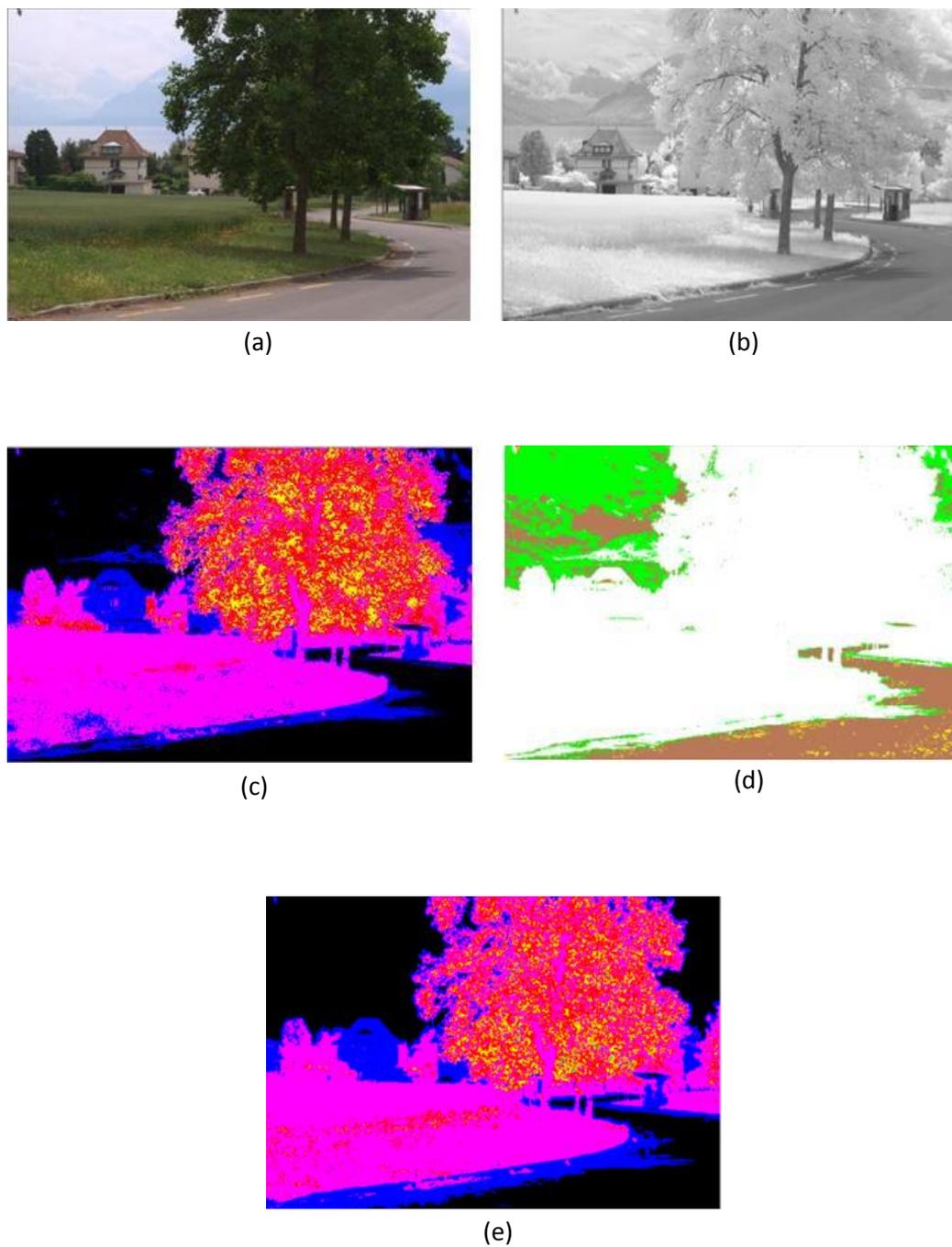


Figura 2.7: Aplicación de Índices de Vegetación. (a) Imagen VS. (b) Imagen NIR. (c) NDVI. (d) RVI. (e) ENDVI.

La Figura 2.8 muestra la paleta de colores utilizada para el mapeo de los valores de los índices en el rango de -1 a 1.



Figura 2.8: Paleta de colores utilizada para el mapeo de los índices de vegetación.

Los valores fuera de este rango se representarán a través del color blanco.

CAPÍTULO 3

INSTALACIÓN Y CONFIGURACIÓN DE LA APLICACIÓN

En este capítulo se presentan las herramientas utilizadas para el desarrollo de la solución propuesta, describiendo los puntos más relevantes como: características de las mismas y detalles de la instalación.

3.1. Herramientas de desarrollo

En esta primera sección se especificarán las herramientas que han sido utilizadas para el desarrollo de la aplicación.

Qt Creator

Qt Creator es un entorno de desarrollo integrado, llamado también IDE, multiplataforma C++, JavaScript y QML, que forma parte del SDK para el

desarrollo de aplicaciones con las bibliotecas QT, requiriendo su versión 4.x., creado por Trolltech.

Qt Creator incluye un project manager que puede utilizar una variedad de formatos de proyecto como .pro, CMake, autotools y otros. Un archivo de proyecto puede contener información como qué archivos están incluidos en el proyecto, pasos de generación personalizada y configuraciones para la ejecución de las aplicaciones.

Qt Creator incluye un editor de código e integra Qt Designer para el diseño y construcción de una interfaz gráfica de usuario (GUI). Las características del editor incluyen resaltado de sintaxis y autocompletado, pero no pestañas.

Qt Creator utiliza el compilador de C ++ de la colección de compiladores de GNU en Linux y FreeBSD. En Windows se puede utilizar MinGW o MSVC con la instalación por defecto y también puede utilizar cdb cuando se compila desde el código fuente.

OpenCv

OpenCV (*Open Source Computer Vision*) es una biblioteca de funciones de programación dirigidas principalmente al campo de visión por computador en tiempo real, desarrollado por el centro de investigación de Intel Rusia en Nizhny Novgorod, y ahora con el apoyo de Willow Garage y Itseez. Es gratuito para uso bajo la licencia BSD de código abierto. La biblioteca es

multiplataforma. Se centra principalmente en el procesamiento de imágenes en tiempo real.

Las áreas de aplicación de OpenCV incluyen: Kits de herramientas de funciones en 2D y 3D, sistema de reconocimiento facial, el reconocimiento de gestos, la interacción hombre-máquina (IHM), robótica móvil, identificación de objetos, segmentación y reconocimiento, la percepción de profundidad de 2 cámaras, el seguimiento del movimiento, realidad aumentada, etc.

Qt Widgets for Technical Applications (Qwt)

Es un conjunto de widgets QT personalizados, componentes GUI y utility classes que son principalmente útiles para programas con un perfil técnico. En comparación con un framework para gráficos 2D, éste provee escalas, deslizadores, diales, brújulas, termómetros, ruedas y botones para controlar o visualizar valores, matrices o rangos de tipo double.

Qwt 6.x podría ser usado en todos los ambientes donde se encuentre Qt. Éste es compatible con Qt ≥ 4.4 .

QCustomPlot

Es un widget de QT C++ para el trazado y visualización de datos. No tiene otras dependencias y está totalmente documentado. Esta biblioteca se enfoca en un buen aspecto, gráficos 2D y cuadros de alta calidad, además

de ofrecer un alto rendimiento para aplicaciones de visualización en tiempo real.

QCustomPlot puede exportar a diversos formatos tales como archivos PDF y vectorizar imágenes como PNG, JPG y BMP.

QCustomPlot es útil tanto para visualización de datos en tiempo real dentro de la aplicación así como para la producción de gráficos de alta calidad para otros medios.

Boost

Es un conjunto de bibliotecas de software libre y revisión por pares preparadas para extender las capacidades del lenguaje de programación C++. Su licencia, de tipo BSD, permite que sea utilizada en cualquier tipo de proyectos, ya sean comerciales o no.

Abarca desde bibliotecas de propósito general hasta abstracciones del sistema operativo. Con el objetivo de alcanzar el mayor rendimiento y flexibilidad se hace un uso intensivo de plantillas.

Actualmente Boost está formada por más de 80 bibliotecas individuales, incluidas las bibliotecas de álgebra lineal, la generación de números pseudoaleatorios, multihilos, procesamiento de imágenes, expresiones regulares, pruebas unitarias, entre otros. La mayoría de las bibliotecas Boost

están basadas en cabeceras, funciones en línea y plantillas, por lo que no tienen que ser construidas antes de su uso.

3.2. Instalación y configuración de la aplicación

Primeramente se debe descargar todas las herramientas anteriormente mencionadas.

En los siguientes enlaces se pueden realizar la descarga de cada una.

Qt Creator:

<https://qt-project.org/downloads>

OpenCV:

<http://sourceforge.net/projects/opencvlibrary/files/>

Qwt:

<http://sourceforge.net/projects/qwt/files/>

QCustomPlot:

<http://www.qcustomplot.com/index.php/download>

Boost:

<http://sourceforge.net/projects/boost/files/>

Fase de instalación y configuración de herramientas

1. La primera herramienta a ser instalada será el IDE Qt Creator a través del ejecutable provisto en el link de descarga. Para la aplicación se ha usado la versión Qt Creator 3.0.0 que incluyen las bibliotecas de Qt 5.2.0 con el compilador MSVC.
2. Luego se instalará la librería OpenCV. Para el proyecto se ha utilizado la versión 2.4.7. El ejecutable instalará la librería en el directorio C:\

Se continúa con la configuración de la variable de entorno.

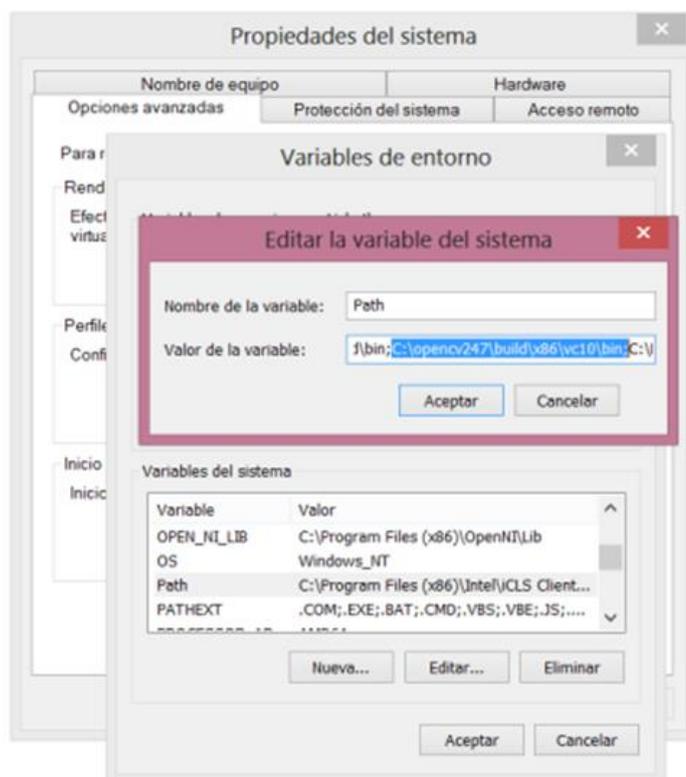


Figura 3.1: Configuración de la variable de entorno.

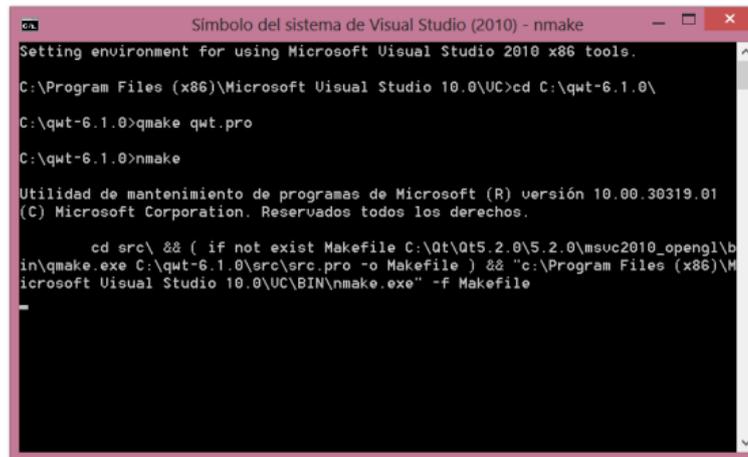
Para esto se debe ingresar a Sistema – Configuración avanzada del sistema – Opciones avanzadas – Variables de entorno

Se edita la variable del sistema PATH y se agrega al valor de la variable el directorio C:\opencv247\build\x86\vc10\bin, como lo muestra la Figura 3.1.

3. En el caso del Qwt, no hay paquetes binarios disponibles, por lo que Qwt necesita ser compilado e instalado en el sistema de destino. Asumiendo que el SDK Qt está instalado, se procede a realizar la instalación del Qwt como se explica a continuación:

- a. Luego de descargar el qwt-6.1.0.zip, se lo descomprime en C:\
- b. Antes de ejecutar el siguiente paso, hay que asegurarse que el qmake.exe se encuentre configurado en la variable de entorno PATH. Para ellos se agregará lo siguiente en el valor de la variable C:\Qt\Qt5.2.0\5.2.0\msvc2010_opengl\bin
- c. Se procederá a abrir la interfaz de líneas de comandos de MSVC y se escribirán los siguientes comandos, como muestra la Figura 3.2

```
cd C:\qwt-6.1.0\  
qmake qwt.pro  
nmake
```



```
Símbolo del sistema de Visual Studio (2010) - nmake
Setting environment for using Microsoft Visual Studio 2010 x86 tools.
C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC>cd C:\qwt-6.1.0\
C:\qwt-6.1.0>qmake qwt.pro
C:\qwt-6.1.0>nmake

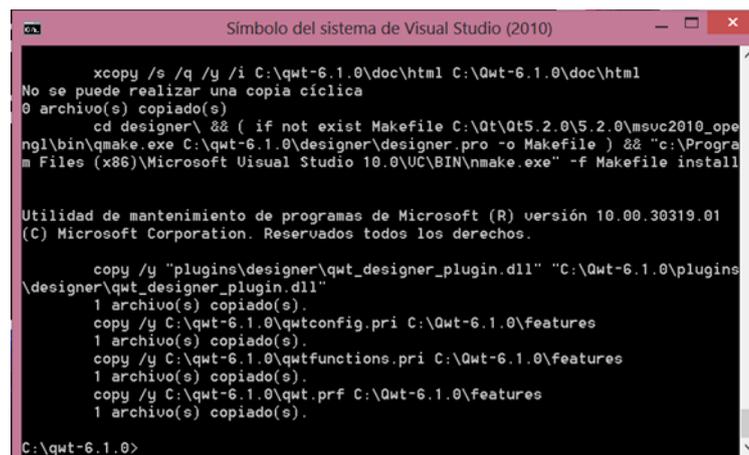
Utilidad de mantenimiento de programas de Microsoft (R) versión 10.00.30319.01
(C) Microsoft Corporation. Reservados todos los derechos.

cd src\ && ( if not exist Makefile C:\Qt\Qt5.2.0\5.2.0\msvc2010_opengl\b
in\qmake.exe C:\qwt-6.1.0\src\src.pro -o Makefile ) && "c:\Program Files (x86)\M
icrosoft Visual Studio 10.0\VC\BIN\nmake.exe" -f Makefile
```

Figura 3.2: Instalación de Qwt(1).

Concluyendo con la instalación, se ejecutará el siguiente comando:

nmake install



```
Símbolo del sistema de Visual Studio (2010)

xcopy /s /q /y /i C:\qwt-6.1.0\doc\html C:\Qwt-6.1.0\doc\html
No se puede realizar una copia ciclica
0 archivo(s) copiado(s)
cd designer\ && ( if not exist Makefile C:\Qt\Qt5.2.0\5.2.0\msvc2010_op
engl\bin\qmake.exe C:\qwt-6.1.0\designer\designer.pro -o Makefile ) && "c:\Progra
m Files (x86)\Microsoft Visual Studio 10.0\VC\BIN\nmake.exe" -f Makefile install

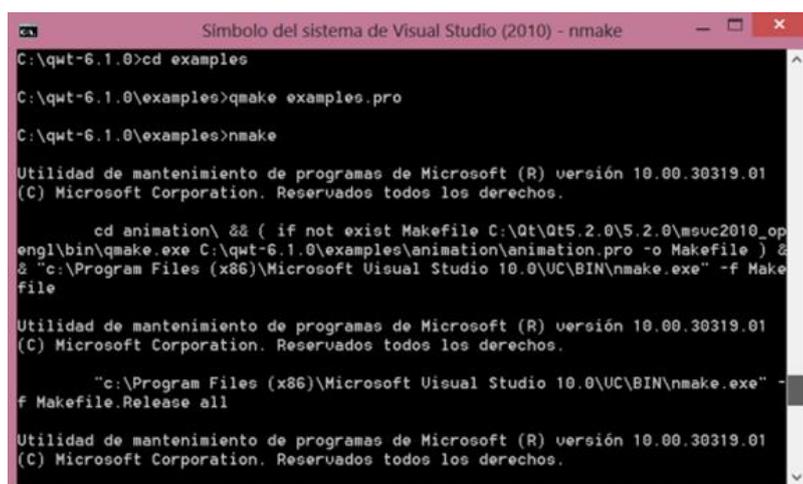
Utilidad de mantenimiento de programas de Microsoft (R) versión 10.00.30319.01
(C) Microsoft Corporation. Reservados todos los derechos.

copy /y "plugins\designer\qwt_designer_plugin.dll" "C:\Qwt-6.1.0\plugins
\designer\qwt_designer_plugin.dll"
1 archivo(s) copiado(s).
copy /y C:\qwt-6.1.0\qwtconfig.pri C:\Qwt-6.1.0\features
1 archivo(s) copiado(s).
copy /y C:\qwt-6.1.0\qwtfunctions.pri C:\Qwt-6.1.0\features
1 archivo(s) copiado(s).
copy /y C:\qwt-6.1.0\qwt.prf C:\Qwt-6.1.0\features
1 archivo(s) copiado(s).
C:\qwt-6.1.0>
```

Figura 3.3: Instalación de Qwt(2).

- d. Si no se activó el autobuilding de los ejemplos en qwtconfig.pri, se tiene que construir los ejemplos de esta manera:

```
cd examples
qmake examples.pro
nmake
```



```

C:\qwt-6.1.0>cd examples
C:\qwt-6.1.0\examples>qmake examples.pro
C:\qwt-6.1.0\examples>nmake

Utilidad de mantenimiento de programas de Microsoft (R) versión 10.00.30319.01
(C) Microsoft Corporation. Reservados todos los derechos.

    cd animation\ && ( if not exist Makefile C:\Qt\Qt5.2.0\5.2.0\msvc2010_op
    engl\bin\qmake.exe C:\qwt-6.1.0\examples\animation\animation.pro -o Makefile ) &
    & "c:\Program Files (x86)\Microsoft Visual Studio 10.0\UC\BIN\nmake.exe" -f Make
    file

Utilidad de mantenimiento de programas de Microsoft (R) versión 10.00.30319.01
(C) Microsoft Corporation. Reservados todos los derechos.

    "c:\Program Files (x86)\Microsoft Visual Studio 10.0\UC\BIN\nmake.exe" -
    f Makefile.Release all

Utilidad de mantenimiento de programas de Microsoft (R) versión 10.00.30319.01
(C) Microsoft Corporation. Reservados todos los derechos.

```

Figura 3.4: Instalación de los ejemplos de Qwt.

Ahora es posible abrir los ejemplos(C:\qwt-6.1-0\examples) en Qt Creator, construir y ejecutar sin errores. Este paso confirma que se ha instalado correctamente Qwt.

4. Finalmente se procede a realizar la instalación de la librería Boost. Para este caso, la versión utilizada es la 1.55.0. El archivo descargado está en formato .zip, por lo que se debe extraer su contenido y colocarlo en el directorio C:\

Fase de configuración del proyecto

Nos dirigimos a abrir el proyecto en Qt Creator, para luego editar el Project manager que se encuentra en formato .pro

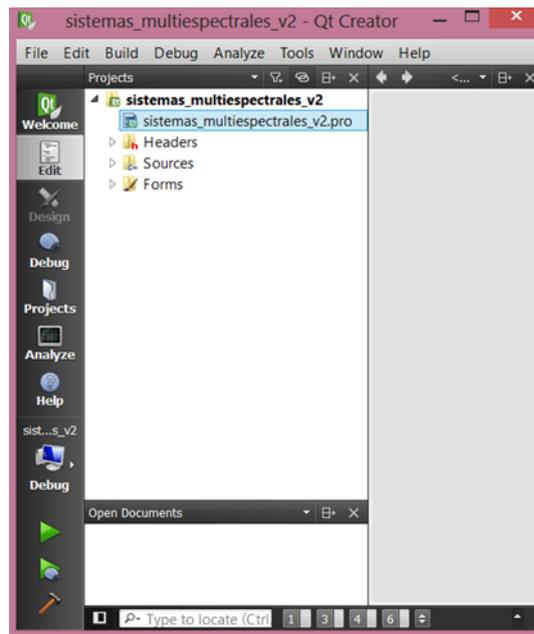


Figura 3.5: Project Manager de la aplicación.

Se procede a configurar las librerías Boost y OpenCv agregando las siguientes líneas a este archivo.

```
INCLUDEPATH += C:/boost_1_55_0
```

```
INCLUDEPATH += C:/opencv247/build/include
```

```
LIBS += C:/opencv247/build/x86/vc10/lib/opencv_core247d.lib \
```

```
C:/opencv247/build/x86/vc10/lib/opencv_highgui247d.lib \
```

```
C:/opencv247/build/x86/vc10/lib/opencv_imgproc247d.lib \
```

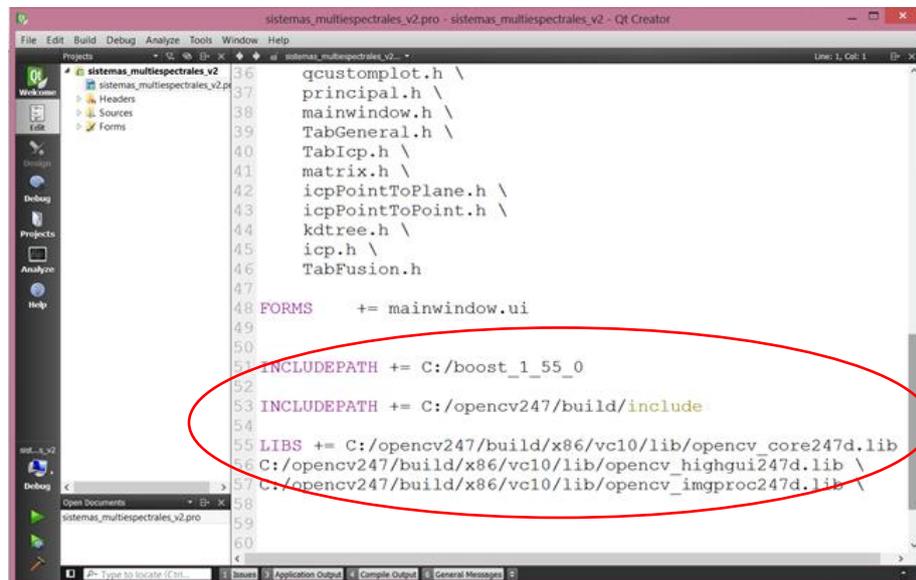


Figura 3.6: Configuraciones de OpenCV y Boost.

Además, para el uso de la librería Qwt en el proyecto, se debe agregar la siguiente línea dentro del Project manager.

CONFIG += qwt

Respecto a la librería QCustomPlot, ésta pueda ser utilizada en el proyecto agregando el archivo llamado qcustomplot.cpp con su respectiva cabecera qcustomplot.h como muestra la Figura 3.7.

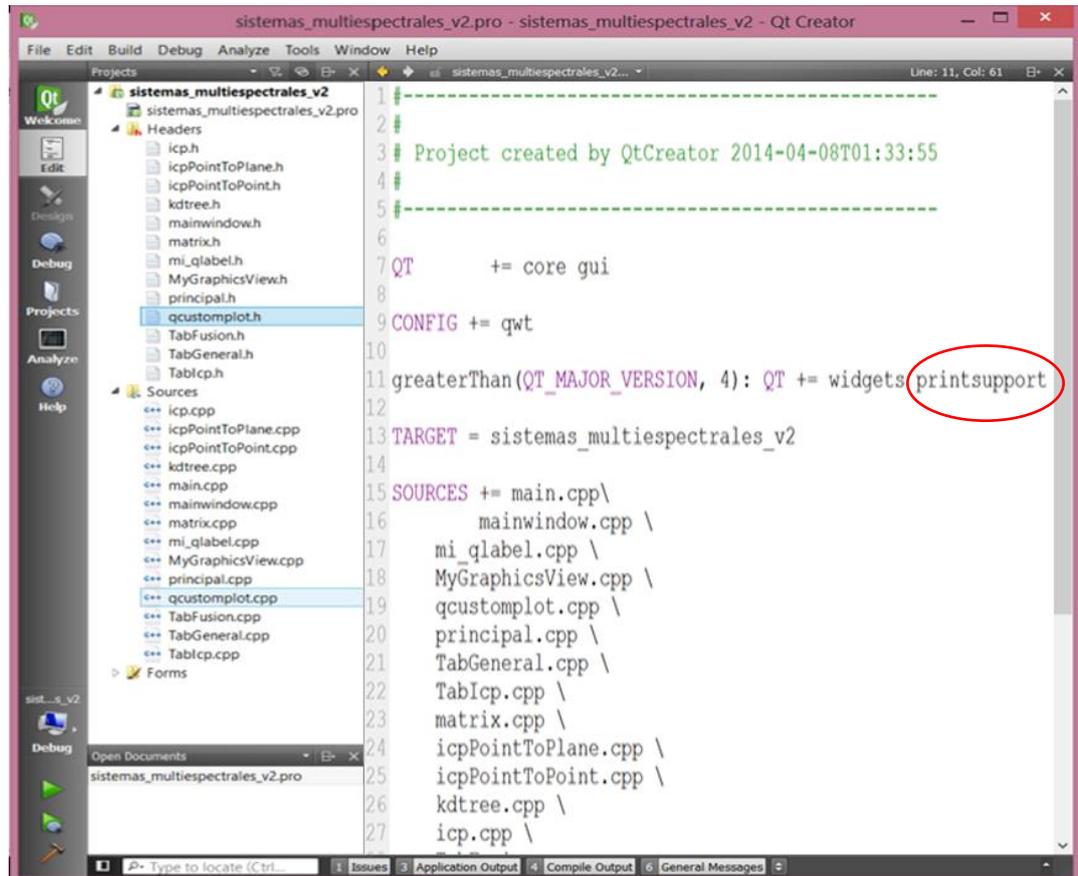


Figura 3.7: Configuración de la librería QCustomPlot.

Si se está utilizando una versión QT mayor a la 5.0, es necesario agregar `printsupport` a la variable `QT` en su archivo `.pro`. Esto se hace después de un condicional `greaterThan(QT_MAJOR_VERSION, 4)`. Esto hace que el `printsupport` y `widgets` no se agreguen al usar versiones anteriores de Qt.

CAPÍTULO 4

DISEÑO Y DESARROLLO DEL SISTEMA EXPERIMENTAL

Luego de haber realizado las configuraciones explicadas en el capítulo anterior, este capítulo mostrará específicamente cómo está compuesto el sistema y cuáles son sus interfaces con las que se interactuará.

4.1. Diseño del sistema (GUI)

Esta sección mostrará las interfaces del sistema y a su vez se detallará cuál es el flujo de trabajo en la aplicación.

La Figura 4.1 muestra cómo está dividido el código del proyecto.

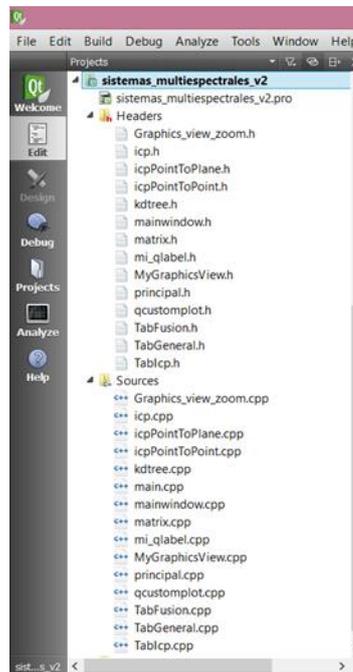


Figura 4.1: Distribución del código del proyecto.

La interfaz principal del sistema experimental es la mostrada en la Figura 4.2:

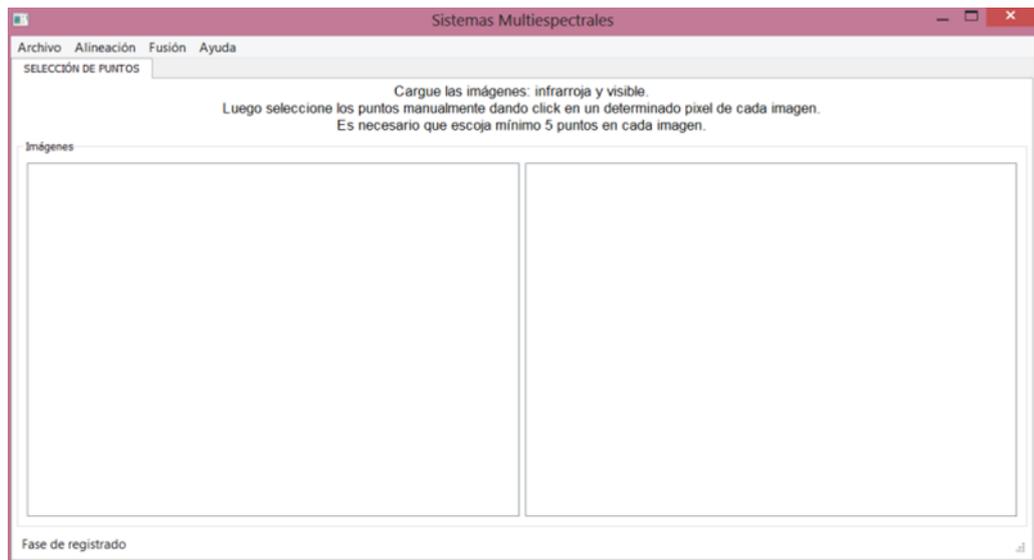


Figura 4.2: Interfaz principal del sistema.

El sistema está dividido en los siguientes módulos:

- Archivo
 1. Abrir Imagen LWIR
 2. Abrir Imagen VS
 3. Guardar los puntos
 4. Borrar los puntos
 5. Salir
- Alineación
 1. Aplicar ICP
 2. Iniciar ICP iterativo
- Fusión
 1. Aplicar Transformación
- Ayuda
 1. Acerca del Sistema



Figura 4.3: Menú del sistema.

4.1.1. Módulo de archivo

En esta opción del menú, se gestionará la carga de las imágenes con las cuales se trabajará. Además, en este módulo se deberá realizar la selección de puntos por lo que se habilitarán las opciones de *borrar todos los puntos* que se hayan seleccionado en caso de alguna equivocación y *guardar los puntos* elegidos en las imágenes LWIR y VS lo que conllevará a mostrar la gráfica posicional de dichos puntos .

4.1.1.1. Abrir imagen

En estas opciones se procederá a cargar las imágenes LWIR y VS respectivamente. También se podrá hacer este paso por medio de los shorcuts dispuestos para cada imagen.



Figura 4.4: Opción Abrir Imagen. (a) LWIR. (b) VS.

Una vez seleccionada la opción, aparecerá un cuadro de diálogo, donde se podrá navegar a través de los directorios con el fin de seleccionar cualquier imagen guardada en el ordenador.

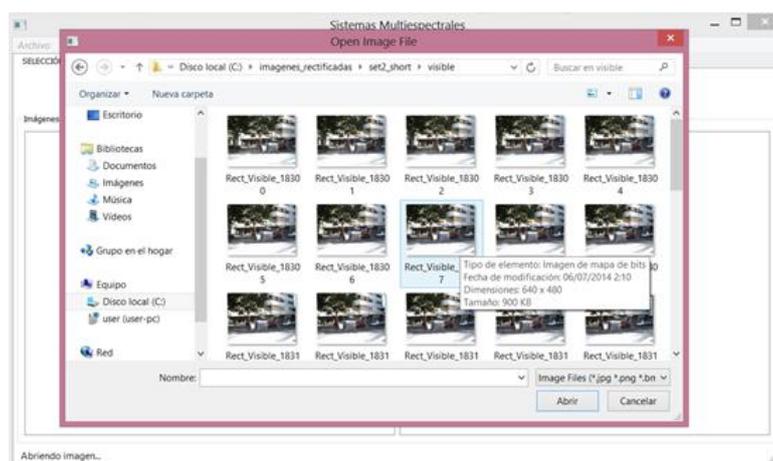


Figura 4.5: Cuadro de diálogo con las imágenes existentes en el PC.

Al seleccionar la imagen, haciendo doble click o presionando la tecla “Enter” sobre ésta, inmediatamente se cargará en el panel de imágenes, quedando listas para seleccionar los puntos característicos de cada una de ellas.



Figura 4.6: Carga de imágenes LWIR y VS.

Luego de este paso, las opciones *Borrar todos los puntos* y *Guardar los puntos* se habilitarán.

4.1.1.2. Borrar todos los puntos

Con esta opción, se podrán eliminar todos los puntos que se han seleccionado, pudiendo así empezar nuevamente el proceso de selección de puntos característicos.

4.1.1.3. Guardar los puntos

Esta opción sirve para guardar la nube de puntos seleccionados de cada imagen. En este caso la nube de puntos en la imagen VS será el “target” mientras que la nube de puntos en la imagen LWIR será el “source”. Aunque para el algoritmo ICP, este hecho no tiene importancia, ya que es un algoritmo geométrico, en donde lo que importa son las posiciones de los puntos, más no su información de intensidad, es necesario establecer las reglas de estandarización correspondientes.

Al seleccionar esta opción, se pueden presentar algunos errores. A continuación se detallan dichos errores:

1. *Guardar los puntos* sin primeramente haberlos seleccionado.

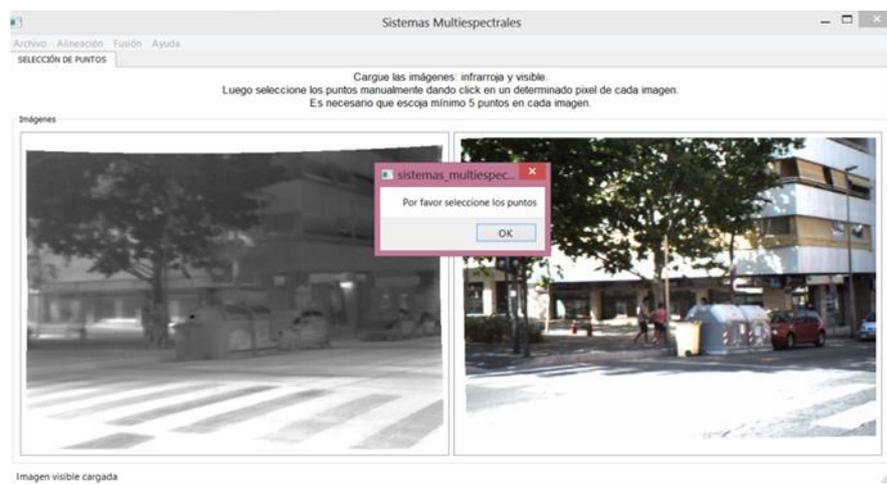


Figura 4.7: Error #1 en opción Guardar los puntos.

2. Seleccionar menos de 4 puntos en cada imagen.

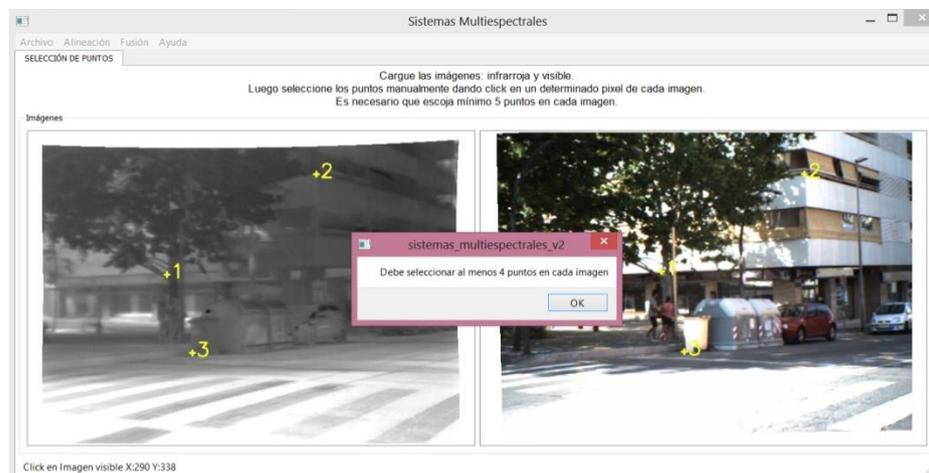


Figura 4.8: Error #2 en opción Guardar los puntos.

3. Seleccionar más de 4 puntos pero diferente cantidad en cada imagen.

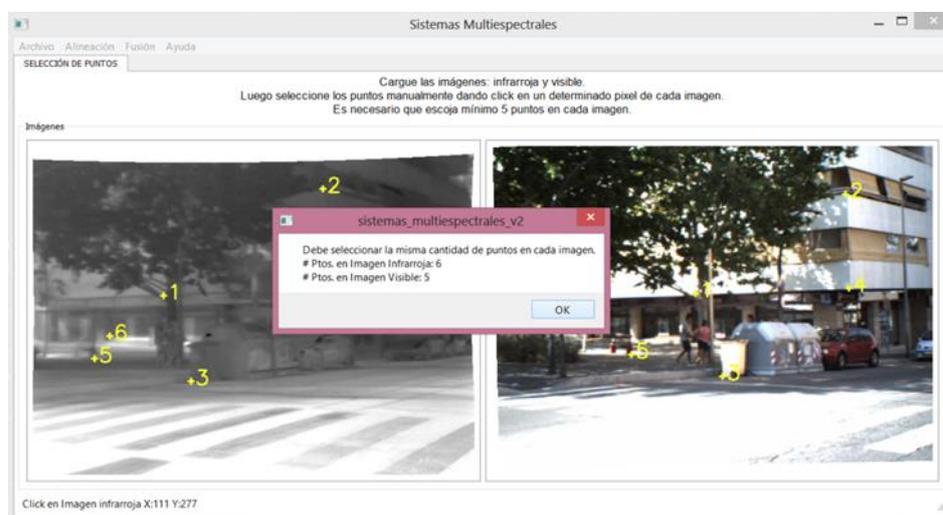


Figura 4.9: Error #3 en opción Guardar los puntos.

Si por el contrario, se ha procedido a seleccionar los puntos de interés en cada imagen, haciendo click sobre éstos y sin cometer alguno de los errores, como muestra la Figura 4.9, entonces el sistema procederá a dibujar la gráfica posicional de los puntos seleccionados que se observa en la Figura 4.10.



Figura 4.10: Selección de puntos de interés.

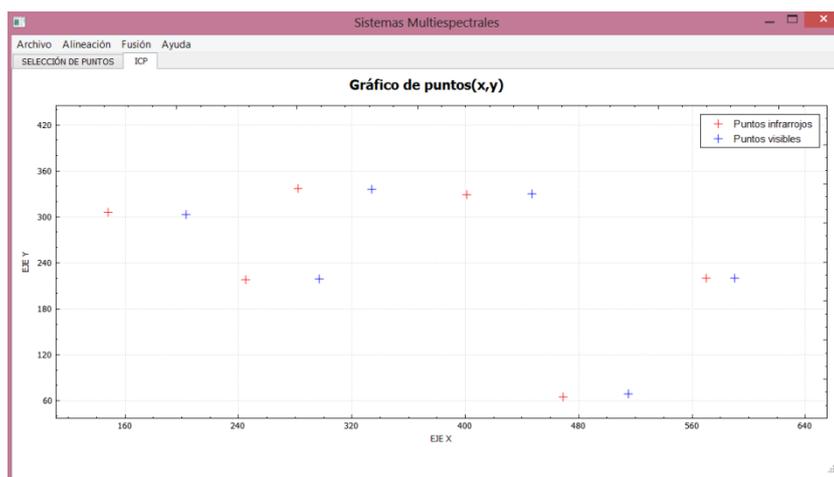


Figura 4.11: Gráfico posicional de puntos de interés.

4.1.2. Módulo alineación de imágenes

En este módulo se ejecutará el algoritmo ICP, en donde la opción *Aplicar ICP* ejecutará de forma directa el algoritmo, mostrando inmediatamente la transformación final, mientras que la opción *Iniciar ICP iterativo* mostrará cada una de sus iteraciones.

4.1.2.1. Iniciar ICP iterativo

Al seleccionarla, el algoritmo ICP empezará a encontrar una transformación inicial para realizar la alineación de las imágenes LWIR y VS, como se muestra en la Figura 4.12.

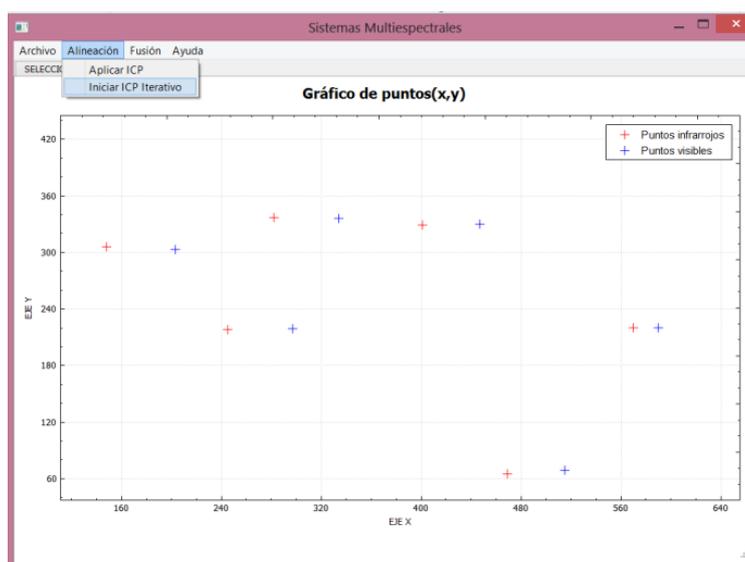


Figura 4.12: Iniciar ICP iterativo.

Para continuar ejecutando cada iteración, se debe dar doble click sobre la gráfica posicional de los puntos. Este proceso se muestra en la Figura 4.13.

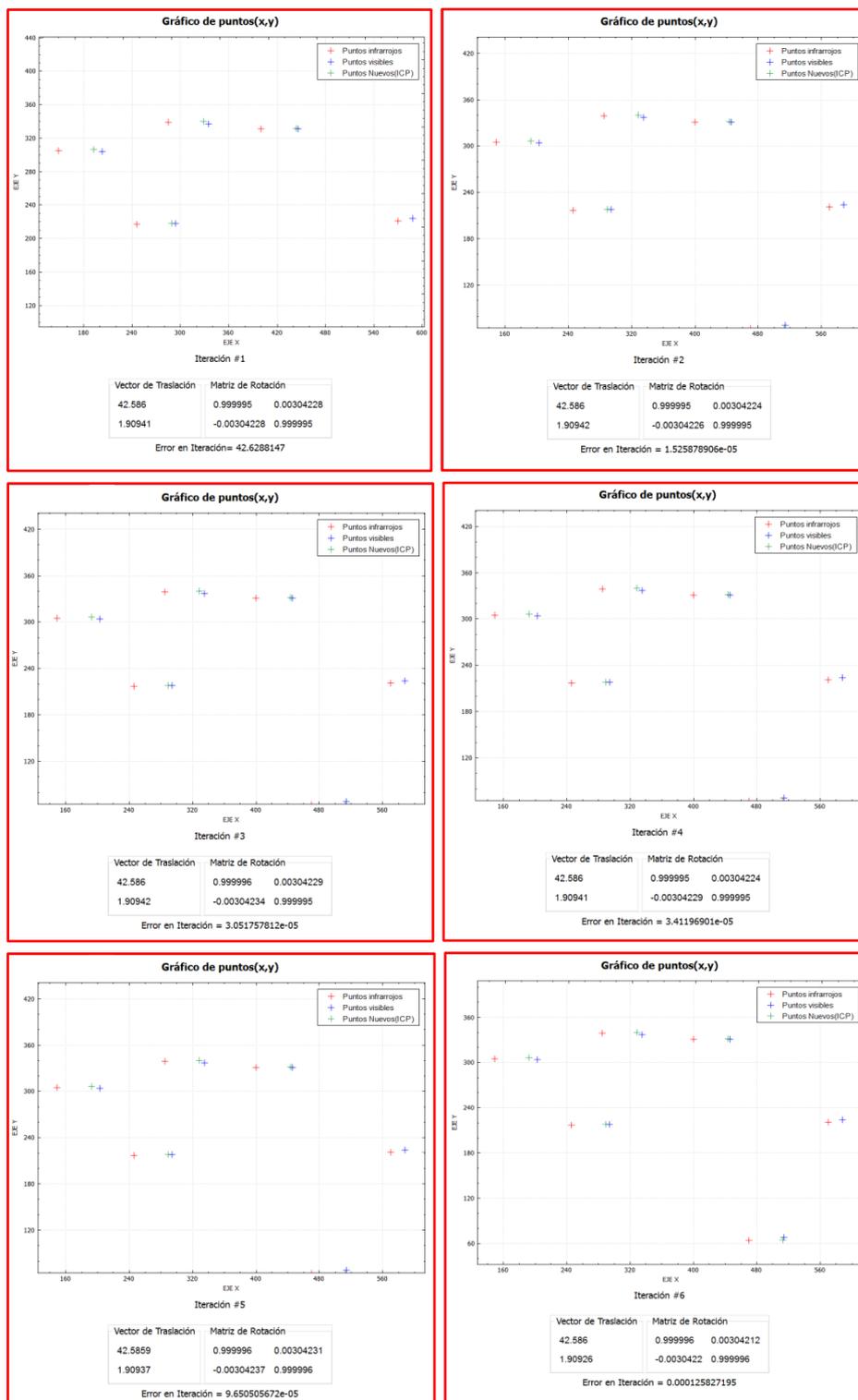


Figura 4.13: Proceso iterativo del ICP.

Finalmente, el algoritmo ICP logra encontrar la transformación para alinear correctamente las imágenes, mostrándose lo siguiente:

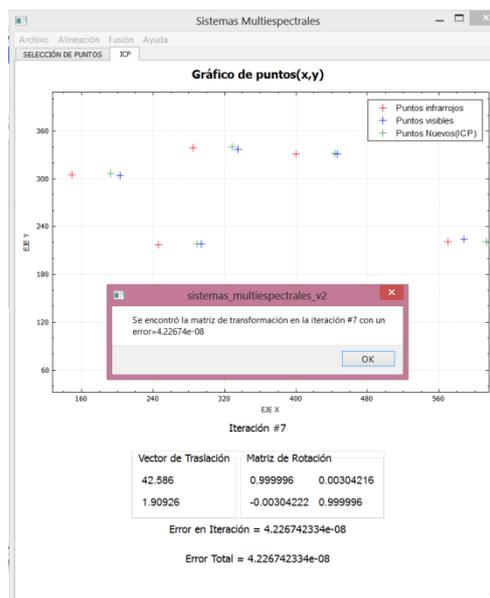


Figura 4.14: Fin del algoritmo ICP.

4.1.3. Módulo fusión de información multiespectral

En esta sección se presentan los diferentes índices hallados como resultado de la fusión multiespectral realizada, por lo que se aplica la transformación a la imagen “source” o LWIR para que se alinee con la imagen “target” o VS.

4.1.3.1. Aplicar la transformación

Seleccionando esta opción del menú, se realizará la alineación de ambas imágenes, pudiendo generar los índices de fusión y los distintos métodos de representar dicha información.

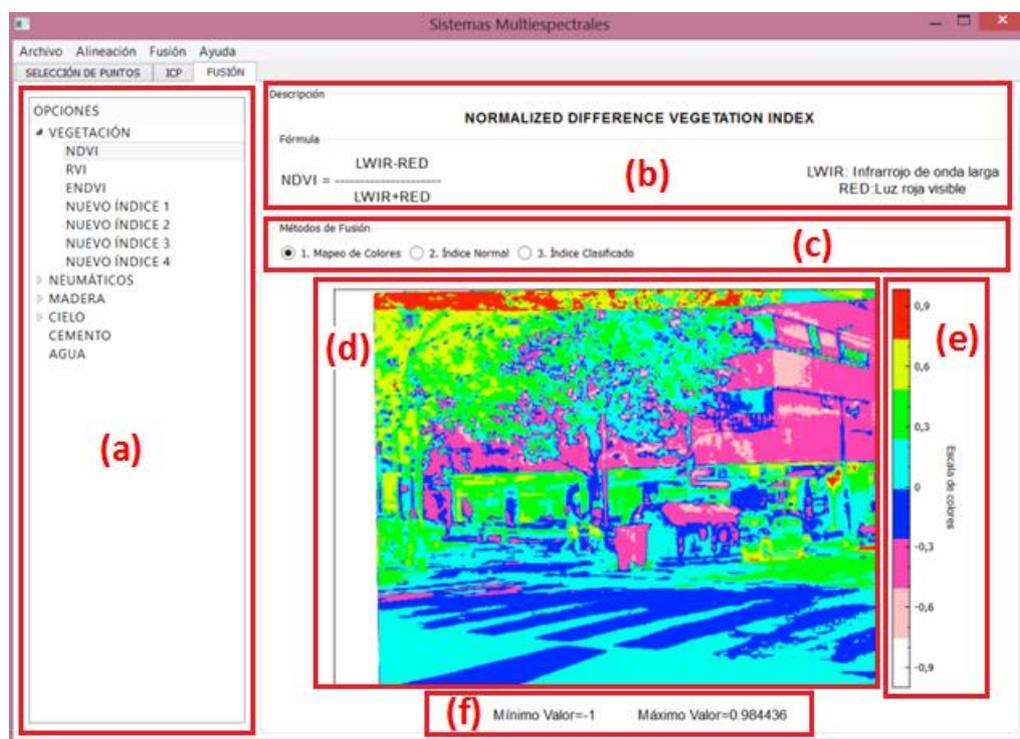


Figura 4.15: Interfaz de la Fusión Multiespectral.

- (a) Panel con las opciones de diferentes características con sus respectivos índices.
- (b) Información acerca del índice seleccionado con la fórmula utilizada.
- (c) Panel con las opciones de métodos de fusión.
- (d) Imagen fusionada.
- (e) Paleta de colores asociada a la imagen.
- (f) Rango de valores de la imagen fusionada.

4.1.4. Ayuda

Se describe brevemente cómo interactuar con el sistema.



Figura 4.16: Interfaz de la Ayuda del Sistema.

4.2. Desarrollo del sistema (ICP)

En este sistema experimental se ha utilizado la librería LIBICP [18], en donde el algoritmo ICP implementa la métrica *point-to-point* basado en SVD así como el *point-to-plane* linearizado. Además, la LIBICP es compatible con el rechazo de valores atípicos y ha sido acelerado con el uso de árboles Kd.

A continuación se describe la implementación de esta librería conjuntamente con las funciones más relevantes de la misma. Así también, se explicará cómo trabaja el algoritmo ICP y cuáles son los parámetros necesarios para su correcto desempeño.

Para empezar mencionaremos los archivos C++ que forman parte de LIBICP, cada uno con sus respectivos archivos cabecera. Estos son:

- matrix.cpp
- icp.cpp
- kdtree.cpp
- icpPointToPoint.cpp
- icpPointToPlane.cpp

De estos archivos, el icpPointToPlane no será utilizado, ya que que en este caso el algoritmo será aplicado a imágenes 2D, por lo que el icpPointToPoint será la métrica de error usada.

Para poder utilizar la librería LIBICP, primero se debe crear un objeto de la métrica escogida, es decir de la clase IcpPointToPoint, que a su vez es una clase derivada de la clase base Icp. Para esto es necesario ejecutar la siguiente instrucción:

```
IcpPointToPoint icp(M, num, dim)
```

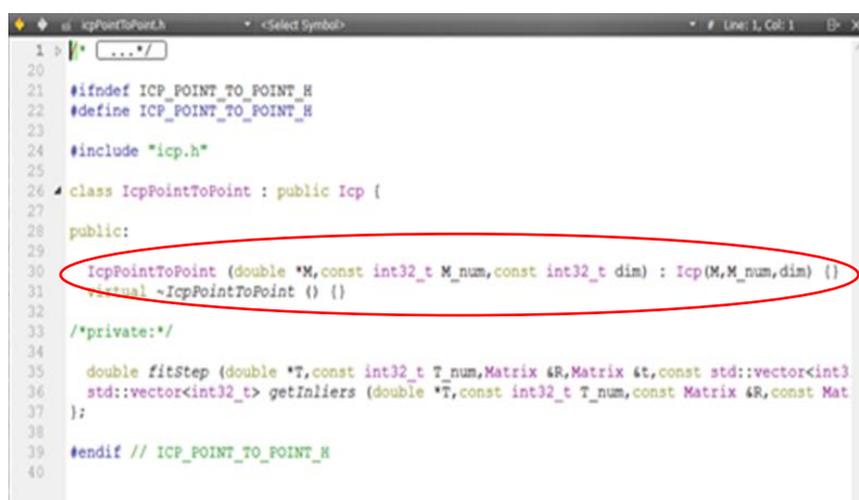
Donde:

M (double *): nube de puntos modelo o “target”.

M_num (int32_t): número de puntos en el modelo o “target”

dim (int32_t): dimensión de los puntos en “target”

Debido a la herencia, al invocar al constructor del IcpPointToPoint, éste lo que hace es crear un objeto Icp, ya que también se invoca al constructor de la clase Icp como lo muestra la Figura 4.17.



```

1 > |* ...*/
20
21 #ifndef ICP_POINT_TO_POINT_H
22 #define ICP_POINT_TO_POINT_H
23
24 #include "icp.h"
25
26 class IcpPointToPoint : public Icp {
27
28 public:
29
30 IcpPointToPoint (double *M,const int32_t M_num,const int32_t dim) : Icp(M,M_num,dim) {}
31 virtual ~IcpPointToPoint () {}
32
33 /*private:*/
34
35 double fitStep (double *T,const int32_t T_num,Matrix &R,Matrix &t,const std::vector<int3
36 std::vector<int32_t> getInliers (double *T,const int32_t T_num,const Matrix &R,const Mat
37 );
38
39 #endif // ICP_POINT_TO_POINT_H
40

```

Figura 4.17: Código del constructor del IcpPointToPoint.

El constructor Icp realiza algunas validaciones respecto a la dimensión de los puntos y a la cantidad de puntos que hay en la nube “target”. Si no se incumple alguna validación, el algoritmo copia todos los puntos del modelo a un arreglo multidimensional que es creado con la ayuda de la librería Boost. Este arreglo llamado M_data cambia su tamaño de acuerdo a M_num y dim, para luego ser llenado con cada uno de los puntos del modelo como se observa en la Figura 4.18.

```

1 > //+ ...*/
20
21 #include "icp.h"
22
23 using namespace std;
24
25 Icp::Icp (double *M, const int32_t M_num, const int32_t dim) :
26     dim(dim), sub_step(10), max_iter(200), min_delta(1e-6) {
27
28     // check for correct dimensionality
29     if (dim!=2 && dim!=3) {
30         cout << "ERROR: LIBICP works only for data of dimensionality 2 or 3" << endl;
31         M_tree = 0;
32         return;
33     }
34
35     // check for minimum number of points
36     if (M_num<4) {
37         cout << "ERROR: LIBICP works only with at least 4 model points" << endl;
38         M_tree = 0;
39         return;
40     }
41
42     // copy model points to M_data
43     M_data.resize(boost::extents[M_num][dim]);
44     for (int32_t m=0; m<M_num; m++)
45         for (int32_t n=0; n<dim; n++)

```

Figura 4.18: Código del constructor Icp.

Finalmente se crea el árbol kd del arreglo multidimensional M_data. Al ser invocado dicho constructor, se procede con la asignación de valores a ciertas variables que serán utilizadas más adelante, mientras que internamente se hace el llamado a la función build_tree que trabaja conjuntamente con la función build_tree_for_range para formar el árbol kd.

```

67 //
68 //     KDTREE2 implementation
69 //
70
71 // constructor
72 KDTree::KDTree(KDTreeArray& data_in, bool rearrange_in, int dim_in)
73 : the_data(data_in),
74   N ( data_in.shape()[0] ),
75   dim( data_in.shape()[1] ),
76   sort_results(false),
77   rearrange(rearrange_in),
78   root(NULL),
79   data(NULL),
80   ind(N) {
81
82     // initialize the constant references using this unusual C++
83     // feature.
84     //
85     if (dim_in > 0)
86         dim = dim_in;
87
88     build_tree();
89
90     if (rearrange) {
91         // if we have a rearranged tree.
92         // allocate the memory for it.
93         rearranged_data.resize( boost::extents[N][dim] );

```

Figura 4.19: Código para crear un árbol kd.

Continuando con la ejecución del algoritmo ICP, luego de crear la estructura indispensable para la nube de puntos modelo, el siguiente paso es alinear la nube de puntos “source” por medio de la siguiente instrucción:

```
icp.fit(T,T_num,R,t,-1)
```

Donde:

T (double *): nube de puntos “source”

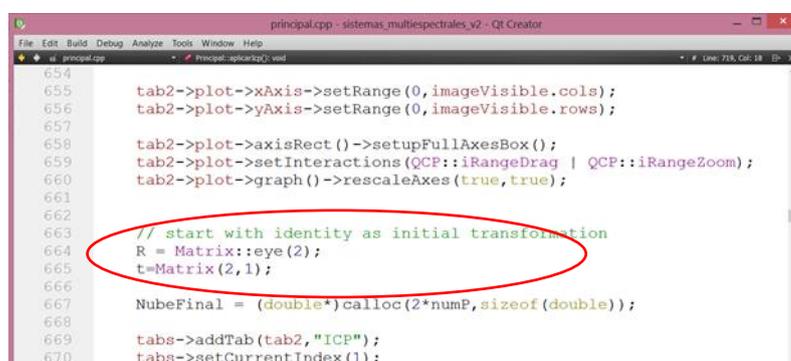
T_num (int32_t): número de puntos en el “source”

R (Matrix): matriz de Rotación

t (Matrix): vector de traslación

-1: No hay un umbral de valor atípico

Cabe mencionar que tanto la matriz de rotación como el vector de traslación de tipo Matrix, han sido previamente inicializados como una matriz identidad y un vector con 1s respectivamente. La Figura 4.20 lo muestra claramente.



```

654
655     tab2->plot->xAxis->setRange(0, imageVisible.cols);
656     tab2->plot->yAxis->setRange(0, imageVisible.rows);
657
658     tab2->plot->axisRect()->setupFullAxesBox();
659     tab2->plot->setInteractions(QCP::iRangeDrag | QCP::iRangeZoom);
660     tab2->plot->graph()->rescaleAxes(true, true);
661
662
663     // start with identity as initial transformation
664     R = Matrix::eye(2);
665     t=Matrix(2,1);
666
667     NubeFinal = (double*) calloc(2*numP, sizeof(double));
668
669     tabs->addTab(tab2, "ICP");
670     tabs->setCurrentIndex(1);

```

Figura 4.20: Código para inicializar la transformación.

Los resultados finales de la transformación se guardarán en las variables **R** y **t** que pueden variar de las iniciales debido a la ejecución del algoritmo ICP.

Una vez que ya se hayan obtenido **R** y **t**, se aplicará una “*Affine Transformation*” de acuerdo a [22], con el fin de alinear el “target” y el “source”.

CAPÍTULO 5

RESULTADOS EXPERIMENTALES

En este capítulo se analizarán las imágenes resultantes obtenidas de las diversas combinaciones de los valores de las bandas multiespectrales. Dichas imágenes han sido adaptadas a una escala de colores para cada fusión realizada.

5.1. Resumen

Para realizar las pruebas se cuenta con una base de imágenes que consta de 3 grupos de diferentes imágenes y con 58 pares de imágenes LWIR-VS en total.

Respecto a la selección de puntos, éstos podrán ser seleccionados de forma manual, en este caso se ha aceptado la sugerencia en la literatura, en donde los puntos de las nubes “target” y “source” son aquellos puntos de los bordes o nodos de una malla triangular en lugar de todos los puntos de la imagen.

En nuestro caso, como no contamos con mallas triangulares, los puntos seleccionados corresponden a puntos característicos de la imagen (en general corners).

Luego se procederá a ejecutar el algoritmo ICP con los puntos elegidos para hallar la matriz de rotación y el vector de traslación. La métrica de error utilizada para este trabajo será la distancia *point-to-point*.

Finalmente se procede a combinar de diferentes formas los valores de intensidad de cada una de las imágenes por medio de un método de fuerza bruta. En este caso, las imágenes visibles serán analizadas con un formato RGB mientras que las imágenes infrarrojas en escala de grises.

5.2. Metodología de las pruebas

Para empezar a realizar las pruebas, es necesario que se seleccionen como mínimo 4 puntos con sus correspondientes en cada una de las imágenes.

Una vez escogidos los puntos, se ejecuta el algoritmo ICP, culminando con la fusión de la información de las imágenes VIS-LWIR de acuerdo a los índices mencionados en el capítulo 2, que serán el punto de partida para la formación de nuevos índices.

Las pruebas fueron realizadas con un par de imágenes VIS-LWIR de cada grupo con los que se cuentan.

Los índices obtenidos han sido normalizados tomando en cuenta el valor mínimo y máximo como los límites del rango de valores representados en la paleta de colores.

Debido a que los rangos de cada índice son desconocidos, la aplicación permite ajustar la escala de colores, admitiendo realizar un zoom a dicha escala con la finalidad de tener una mejor visualización y poder detectar con facilidad algún rango de detección de algún objeto que coincida en cualquier otra imagen.

5.3. Resultados

En el caso del algoritmo ICP, se procederá a analizar el valor del error generado en cada una de la pruebas y para realizar un análisis comparativo en la fusión de imágenes es necesario ajustar los resultados de los índices a una escala de colores para realizar una evaluación visual.

5.3.1. Algoritmo ICP

Para hacer una evaluación del comportamiento del algoritmo ICP, se ha realizado un pequeño estudio. Dicho estudio consiste en ejecutar el algoritmo sobre un mismo par de imágenes variando la cantidad de parejas de puntos seleccionados, para así poder observar el error

expuesto en cada prueba, en donde se calcula las distancias *point-to-point* de los puntos correspondientes.

El algoritmo ICP encuentra los parámetros de la matriz de rotación y del vector de traslación sí y solo sí el error total es menor al valor del umbral. En este estudio el valor del umbral es de 0.000001.

A continuación se muestra un ejemplo de las pruebas realizadas. En la Figura 5.1, se presenta la fase de selección de 20 parejas de puntos.



Figura 5.1: Fase de selección de 20 parejas de puntos.

Se guardan los puntos y se procede a ejecutar el algoritmo ICP, donde se halla la matriz de transformación con el respectivo error total. Este proceso se observa en la Figura 5.2.

Esta fue la metodología utilizada en cada una de las pruebas para obtener el error total del ICP con un mismo par de imágenes,

incrementando poco a poco la cantidad de pares de puntos seleccionados, hasta llegar a los 20 pares.

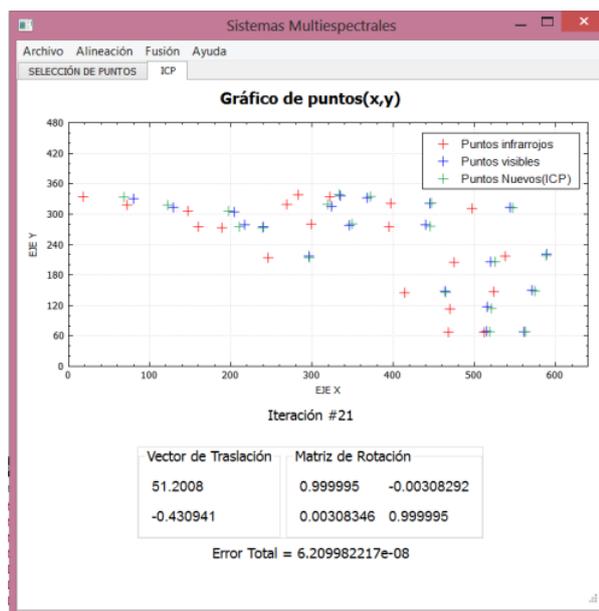


Figura 5.2: Ejecución del algoritmo ICP con 20 parejas de puntos.

Graficando los errores obtenidos aplicando el algoritmo ICP con los puntos seleccionados de la imagen LWIR Rect_Infra18304.jpeg y los puntos de la imagen VS Rect_Visible_18300.bmp, se puede distinguir que el error total disminuye a medida que aumenta el número de pares de puntos seleccionados, como se visualiza en la Figura 5.3.

Tal y como lo muestra el gráfico, el error disminuye mínimamente en cada prueba, pero es preferible que se seleccionen al menos entre 7 a 10 pares de puntos.

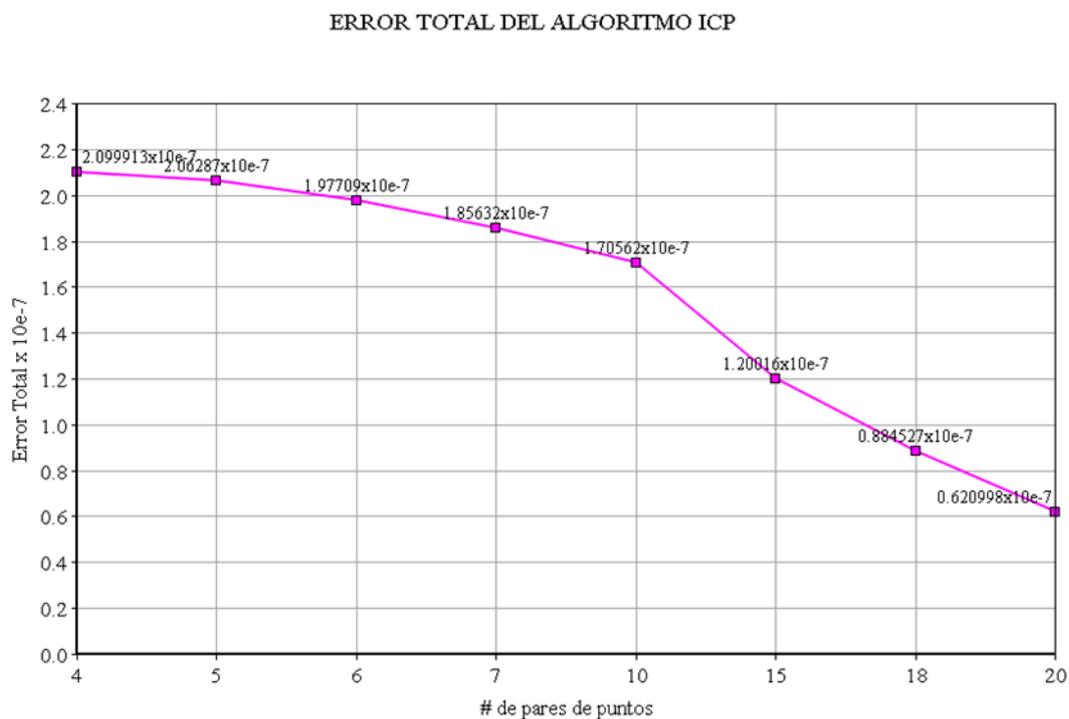


Figura 5.3: Resultados de errores totales del algoritmo ICP.

5.3.2. Índices de detección de vegetación

A continuación, se presentan las imágenes obtenidas a partir de la fusión multiespectral.

Para empezar se tomará como base los índices ya existentes para la fusión entre una imagen NIR y VS, que serán probados con una nueva fusión entre imágenes LWIR y VS, con la finalidad de observar algún comportamiento de estos índices que pudiera aportar en éste estudio,

por lo que para los índices de vegetación existentes tales como el NDVI, ENDVI y RVI se reemplazará el NIR por el LWIR.

NORMALIZED DIFFERENCE VEGETATION INDEX (NDVI)

De acuerdo a la literatura, la fórmula del NDVI tiene como fundamentos principales la condición de la vegetación sana que absorbe la mayor parte de luz visible que cae sobre ella y refleja una gran porción de luz del infrarrojo cercano, mientras que la vegetación poco saludable refleja más luz visible y menos luz del infrarrojo cercano. Por lo tanto, se puede notar que las bandas más sensibles a la vegetación son el NIR y el rojo.

Una gran diferencia entre el NIR y la reflectancia del rojo corresponde a la presencia de vegetación.

Sin embargo, pueden surgir inconvenientes si sólo se toma en cuenta la diferencia entre estas dos bandas, ya que dos píxeles en la misma ubicación de una imagen podrían tener valores diferentes si se toma en cuenta la iluminación del sol, ya que los valores podrían diferir enormemente. Es por esta razón que para evitar este problema, es preciso que la diferencia se divida para la suma de las reflectancias.

$$NDVI = \frac{LWIR - RED}{LWIR + RED}$$

Donde LWIR corresponde al valor del pixel en la imagen del infrarrojo lejano y RED al canal rojo del espectro visible.

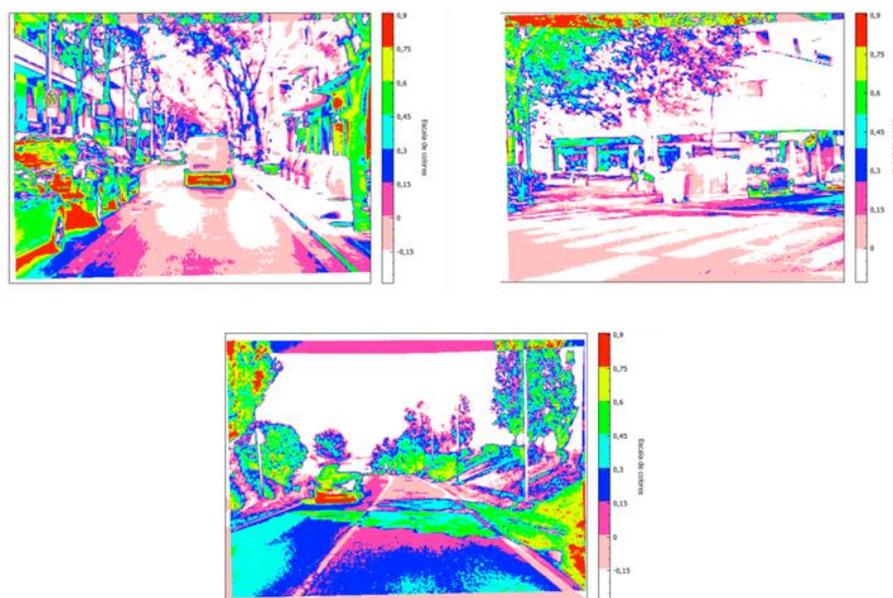


Figura 5.4: Fusión multiespectral (LWIR-VS) con el índice NDVI.

RATIO VEGETATION INDEX (RVI)

A través de la división de la banda NIR y la banda visible roja, se puede obtener el contraste entre dichas bandas para los píxeles de vegetación, en donde se consiguen altos valores por combinaciones de rojo bajo debido a la absorción de la clorofila y alta reflectancia en el infrarrojo por la propia estructura de la hoja.

Puesto que esta fórmula se construye como una relación, se reducen los problemas de iluminación que el NDVI podría presentar. No obstante, el

riesgo principal en este índice será la división por cero y la escala de medición resultante no es lineal.

$$RVI = \frac{LWIR}{RED}$$

LWIR y RED corresponden a la notación indicada anteriormente.

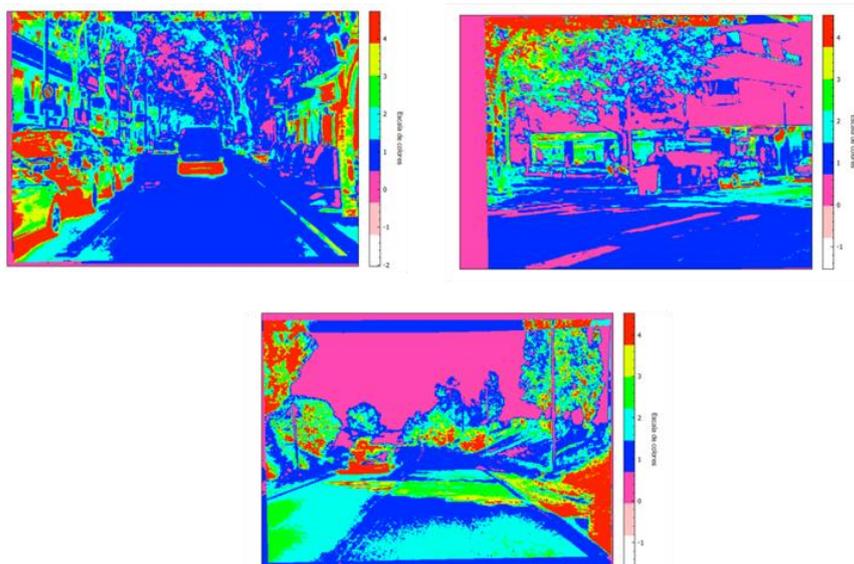


Figura 5.5: Fusión multiespectral (LWIR-VS) con el índice RVI.

ENHANCED NORMALIZED DIFFERENCE VEGETATION INDEX (ENDVI)

El ENDVI está basado en el NDVI, con la diferencia de que este índice ya no toma en cuenta la banda roja, sino más bien los canales utilizados además del NIR son la banda azul y verde.

Siendo así, los canales reflexivos serán el NIR y el verde, por medio de la suma de ambos y el canal azul es multiplicado por dos para compensar la suma anteriormente mencionada.

$$ENDVI = \frac{LWIR + GREEN - 2 * BLUE}{LWIR + GREEN + 2 * BLUE}$$

Donde LWIR corresponde al valor en la imagen infrarroja lejana, GREEN al canal verde y BLUE al canal azul en el espectro visible respectivamente.

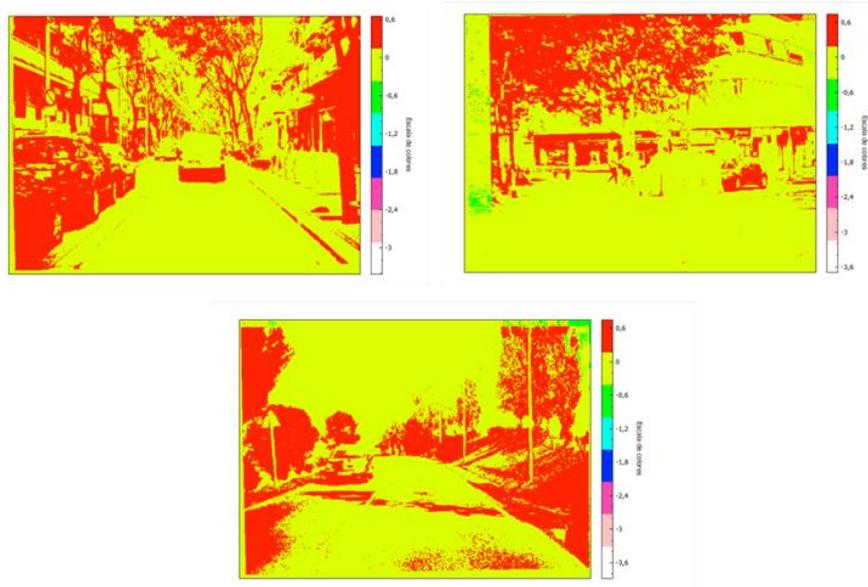


Figura 5.6: Fusión multiespectral (LWIR-VS) con el índice ENDVI.

Los siguientes índices LWIR-IV1, LWIR-IV2, LWIR-IV3 y LWIR-IV4 han sido creados de manera empírica a partir del índice ENDVI, ya que fue el índice que mejores resultados arrojó de las pruebas con los índices de vegetación que ya existían.

Para cada prueba realizada, primeramente se comenzó a aumentar el valor del coeficiente que multiplica a la banda LWIR. A medida que se aumentaba ésta, era necesario ir compensando las bandas visibles verde y azul.

En cada prueba se procedió a aumentar los coeficientes con factores de ± 0.5 para cada canal. A pesar de incrementar los coeficientes tanto para los canales en el numerador como en el denominador, la detección de vegetación era cada vez menos visible, por lo que se optó por intercambiar los coeficientes para cada banda.

NUEVO ÍNDICE DE VEGETACIÓN 1 (LWIR-IV1)

En el caso del índice LWIR-IV1, se intercambiaron los índices de los canales reflexivos LWIR y GREEN, ya que en el denominador se sumaría el canal BLUE. De ésta forma los valores probablemente resulten muy cercanos y con un rango pequeño. Además el factor 0.2 adicionado ha sido únicamente utilizado para efectos de que el rango sea un poco más alto y se aproxime a 1.

$$LWIR - IV1 = \frac{4 * LWIR + 6 * GREEN - 2.5 * BLUE}{6 * LWIR + 4 * GREEN + 2.5 * BLUE} + 0.2$$

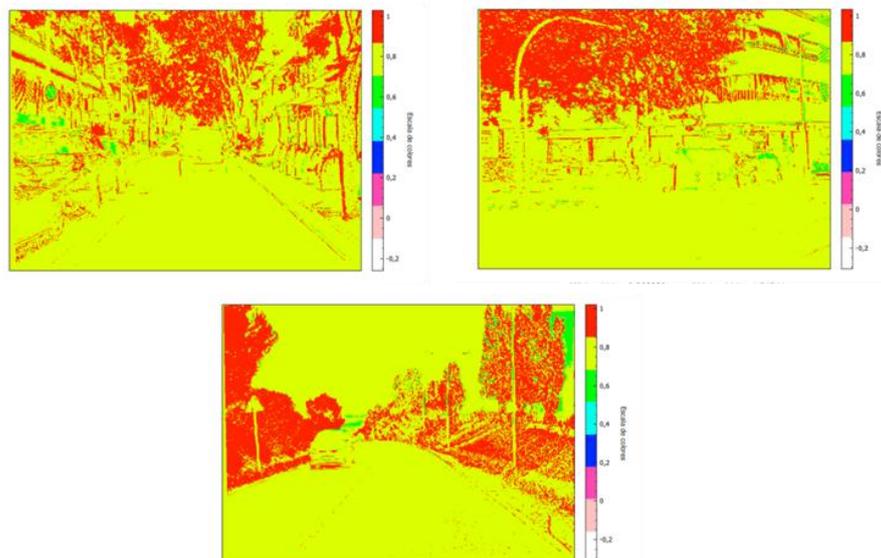


Figura 5.7: Fusión multiespectral (LWIR-VS) con el índice LWIR-IV1.

NUEVO ÍNDICE DE VEGETACIÓN 2 (LWIR-IV2)

Al igual que el LWIR-IV1, este índice fue creado a través de un proceso de prueba y error. Se procedió a aumentar cada coeficiente, así al LWIR se le adicionó 2.5, al GREEN 3.5 y al BLUE 1, tomando en cuenta únicamente los valores que se encuentran en el numerador, para finalmente utilizar el mismo método de intercambio de factores utilizado en el NDVI1, así como también fue utilizado el factor 0.2 para que el rango se aproxime a 1.

$$LWIR - IV2 = \frac{6.5 * LWIR + 9.5 * GREEN - 3.5 * BLUE}{9.5 * LWIR + 6.5 * GREEN + 3.5 * BLUE} + 0.2$$

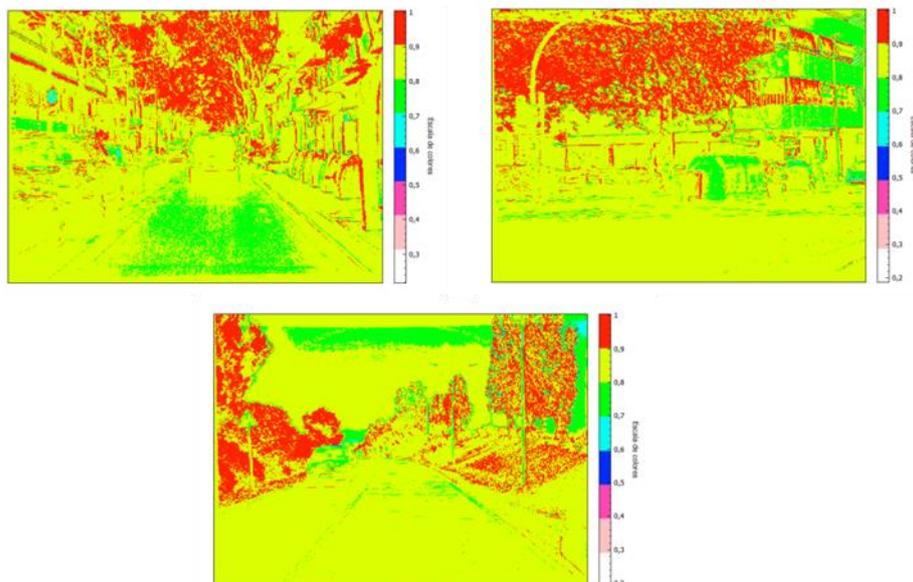


Figura 5.8: Fusión multiespectral (LWIR-VS) con el índice LWIR-IV2.

NUEVO ÍNDICE DE VEGETACIÓN 3 (LWIR-IV3)

Básicamente tiene el mismo enfoque que el LWIR-IV1, pero con la diferencia de un valor de -0.5 en la banda BLUE.

Se puede notar que este índice detecta mayor cantidad de vegetación, pero también detecta otros objetos que no son requeridos.

$$LWIR - IV3 = \frac{4 * LWIR + 6 * GREEN - 2 * BLUE}{6 * LWIR + 4 * GREEN + 2 * BLUE} + 0.2$$

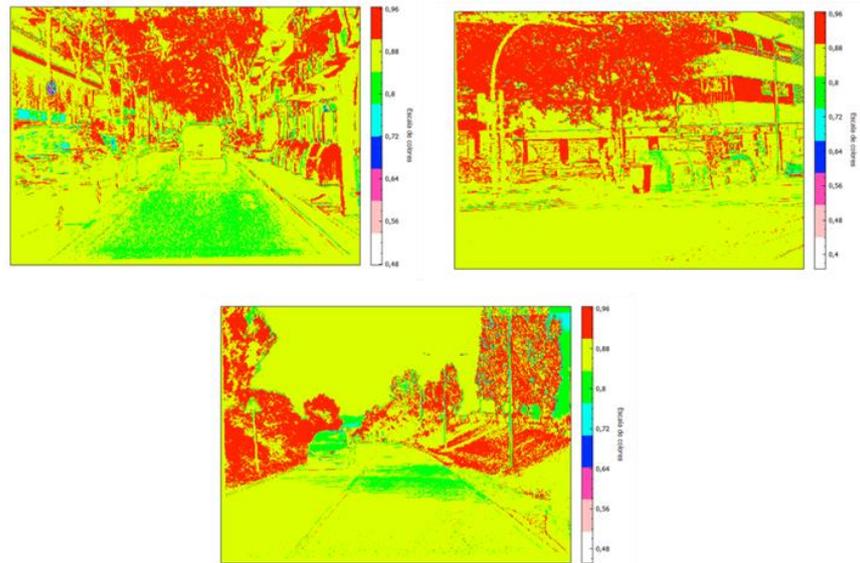


Figura 5.9: Fusión multiespectral (LWIR-VS) con el índice LWIR-IV3.

NUEVO ÍNDICE DE VEGETACIÓN 4 (LWIR-IV4)

El LWIR-IV4 aplica el intercambio de coeficientes entre todas las bandas. Así mismo, empíricamente se procedió a realizar la mezcla de los valores de cada canal.

$$LWIR - IV4 = \frac{6.5 * LWIR + 8 * GREEN - 5 * BLUE}{8 * LWIR + 5 * GREEN + 6.5 * BLUE} + 0.2$$

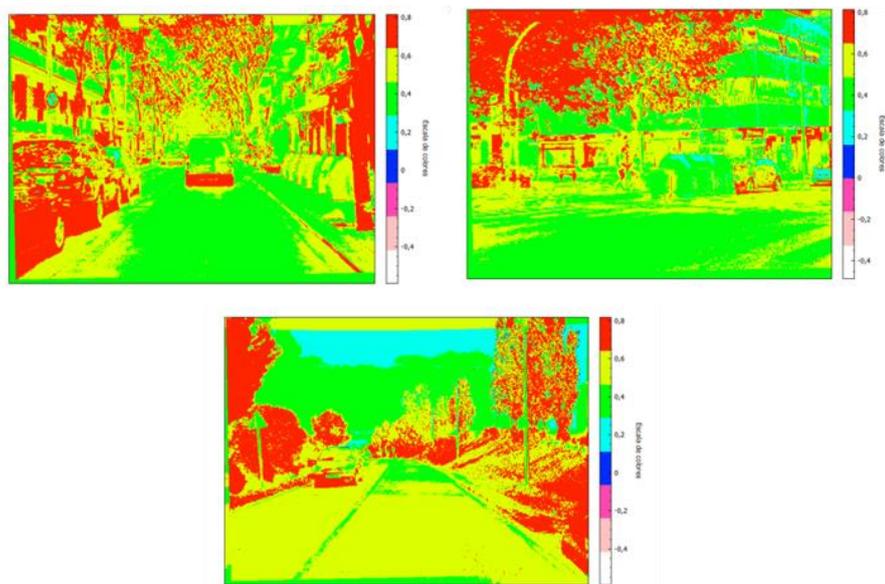


Figura 5.7: Fusión multispectral (LWIR-VS) con el índice LWIR-IV4.

La Tabla 1 presenta los rangos entre los que se encuentra cada índice calculado.

De acuerdo a esta información, se puede observar que el rango del índice RVI es mucho más amplio en comparación con los demás.

También se puede notar que tanto el índice NDVI como el ENDVI son los que constan de valores negativos.

Además los valores obtenidos en los índices de vegetación creados se localizan en rangos cercanos.

Imagen LWIR	Imagen VS	Índice	Mínimo Valor	Máximo Valor
Rect_Infra21805	Rect_Visible_21810	NDVI	-1	0.951613
		RVI	0	40.3333
		ENDVI	-0.535632	1
		LWIR-IV1	0.334421	1.28919
		LWIR-IV2	0.396109	1.28365
		LWIR-IV3	0.431029	1.29003
		LWIR-IV4	0.00405983	1.19509
Rect_Infra18310	Rect_Visible_18300	NDVI	-1	0.992188
		RVI	0	255
		ENDVI	-0.673759	1
		LWIR-IV1	0.280658	1.34214
		LWIR-IV2	0.347609	1.33466
		LWIR-IV3	0.393772	1.34859
		LWIR-IV4	-0.0802292	1.21906
Rect_Infra2905	Rect_Visible_2900	NDVI	-1	0.946565
		RVI	0	36.4266
		ENDVI	-0.749049	1
		LWIR-IV1	0.107859	1.58281
		LWIR-IV2	0.180338	1.56034
		LWIR-IV3	0.240373	1.60157
		LWIR-IV4	-0.259473	1.39671

Tabla 1: Rangos de los índices de vegetación.

Volviendo al análisis visual, claramente nos podemos fijar que los valores que corresponden a la vegetación en el NDVI se encuentran muy dispersos y no presentan algún patrón al igual que el RVI. Por el contrario, el índice ENDVI si permite identificar los árboles de manera

notoria, mientras que los índices recién creados, arrojan resultados positivos respecto a la detección de otras características.

Sintetizando lo expuesto anteriormente se presentan los resultados respecto al intervalo que identifica la vegetación en cada índice en la Tabla 2.

Índice	Rango de Vegetación
NDVI	0 - 0.9
RVI	1.2 – 4.5
ENDVI	0.2 – 0.8
LWIR-IV1	0.85 – 1.1
LWIR-IV2	0.9 - 1
LWIR-IV3	0.89 – 0.96
LWIR-IV4	0.65 – 0.8

Tabla 2: Rangos de detección de vegetación.

Adicionalmente, en estas pruebas se aplicaron dos métodos de fusión utilizados en la detección de vegetación cuando la fusión es entre una imagen NIR y VS, pero fueron aplicadas para la fusión entre una imagen LWIR y VS, con el objetivo de verificar que efectivamente el rango de los índices no es entre -1 a 1 como lo dice la revisión bibliográfica para el caso de las NIR-VS. Dichas pruebas están en el Anexo 1. Sin embargo, estos métodos no han sido considerados para los nuevos índices, ya que no se conocen los límites de cada nuevo índice para cualquier imagen y por lo tanto no se hallaría una forma correcta de normalización.

5.3.3. Índices de detección de cielo

Todos los índices posteriores han sido creados de manera experimental, producto de varias pruebas realizadas a lo largo de este trabajo, tratando de encontrar algún patrón en las fórmulas que sirva como el punto de inicio de tal forma que puedan derivar variantes. A pesar de los intentos, en ninguna de estas fórmulas se puede observar alguna similitud o patrón, ya que éstas fueron halladas en el proceso de prueba y error que se realizaba para los índices de detección de vegetación.

A continuación se muestran las imágenes fusionadas de los índices de detección de cielo:

$$LWIR - IC1 = \frac{GREEN}{2 * LWIR + BLUE}$$

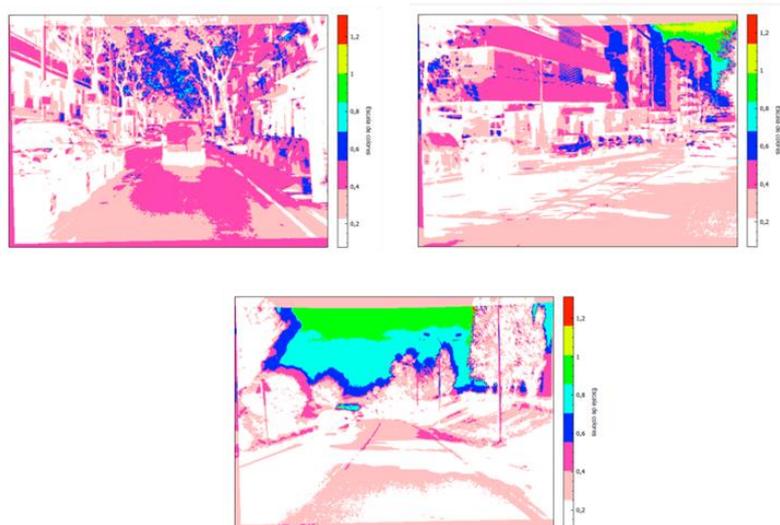


Figura 5.8: Fusión multispectral (LWIR-VS) con el índice LWIR-IC1.

$$LWIR - IC2 = \frac{3*RED-LWIR}{3*RED+GREEN}$$

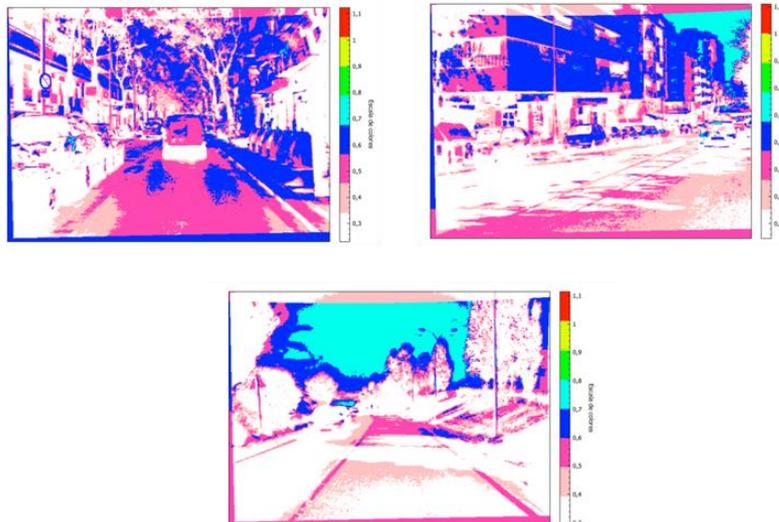


Figura 5.9: Fusión multispectral (LWIR-VS) con el índice LWIR-IC2.

$$LWIR - IC3 = \frac{RED+BLUE}{2*GREEN+2*LWIR}$$

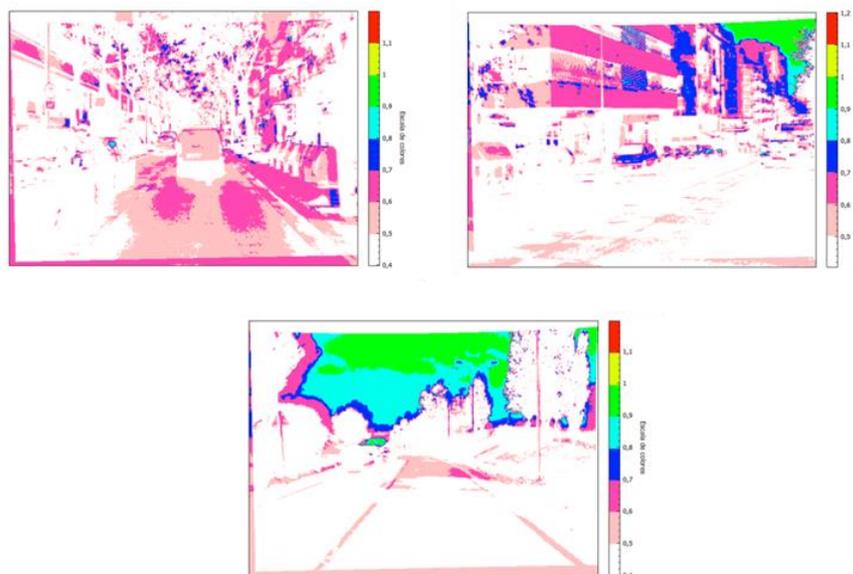


Figura 5.10: Fusión multispectral (LWIR-VS) con el índice LWIR-IC3.

$$LWIR - IC4 = \frac{RED}{2*GREEN+2*LWIR-RED}$$

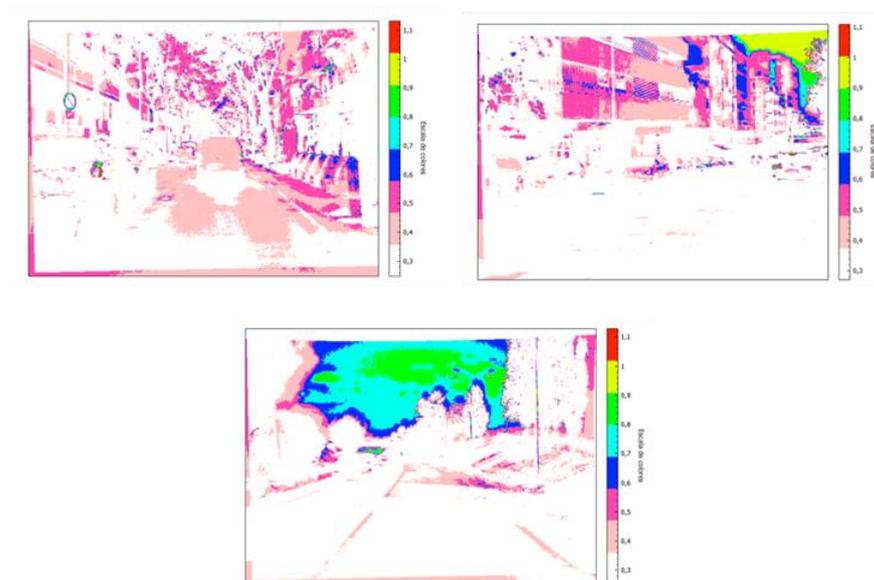


Figura 5.11: Fusión multiespectral (LWIR-VS) con el índice LWIR-IC4.

La Tabla 3 muestra los rangos de valores para cada índice creado para la detección de cielo.

Como se puede observar, el LWIR-IC2 y LWIR-IC4 tienen valores negativos en su rango, con la diferencia de que el LWIR-IC4 tiene un rango más amplio que el del LWIR-IC2, exceptuando la prueba realizada con el primer par de imágenes.

Además el índice LWIR-IC3 nos muestra un rango de valores parecido en cada prueba.

Imagen LWIR	Imagen VS	Índice	Mínimo Valor	Máximo Valor
Rect_Infra21805	Rect_Visible_21810	LWIR-IC1	0	0.894737
		LWIR-IC2	-9.41667	0.79771
		LWIR-IC3	0.22290008	0.941489
		LWIR-IC4	0.00763359	3.81818
Rect_Infra18335	Rect_Visible_18339	LWIR-IC1	0.00414938	3.82051
		LWIR-IC2	-7.53571	0.876777
		LWIR-IC3	0.026455	1.7931
		LWIR-IC4	-97	32.2
Rect_Infra2905	Rect_Visible_2900	LWIR-IC1	0.0129199	10
		LWIR-IC2	-7.125	0.876
		LWIR-IC3	0.0263158	1.79104
		LWIR-IC4	-78	57

Tabla 3: Rangos de los índices de detección de cielo.

Haciendo los respectivos ajustes en la escala, se puede detectar el cielo de las imágenes en los rangos de valores aproximados. Estos rangos son dados en la Tabla 4.

Índice	Rango de Vegetación
LWIR-IC1	0.65 - 1
LWIR-IC2	0.68 – 0.8
LWIR-IC3	0.8 – 1
LWIR-IC4	0.7 – 1

Tabla 4: Rangos de detección de cielo.

5.3.4. Índices de detección de calle

Para detectar las calles en una imagen, realizando varias combinaciones de las bandas a través de la fuerza bruta y pudiendo observar previamente el comportamiento de las fórmulas lineales en los anteriores índices, en donde no se pudo detectar las calles, es preciso modificar éstas fórmulas.

En estos casos se han procedido a utilizar un canal con una potencia de 2 y para compensar el alto valor que se podría obtener de esto, se realiza la multiplicación entre otros 2 canales.

NUEVO ÍNDICE DE CALLE 1 (LWIR-ICA1)

$$LWIR - ICA1 = \frac{RED^2 - LWIR + BLUE}{BLUE * GREEN}$$

NUEVO ÍNDICE DE CALLE 2 (LWIR-ICA2)

$$LWIR - ICA2 = \frac{GREEN * RED + LWIR}{BLUE^2}$$

Las siguientes figuras presentan las imágenes de fusión obtenidas de los índices de detección de calle.

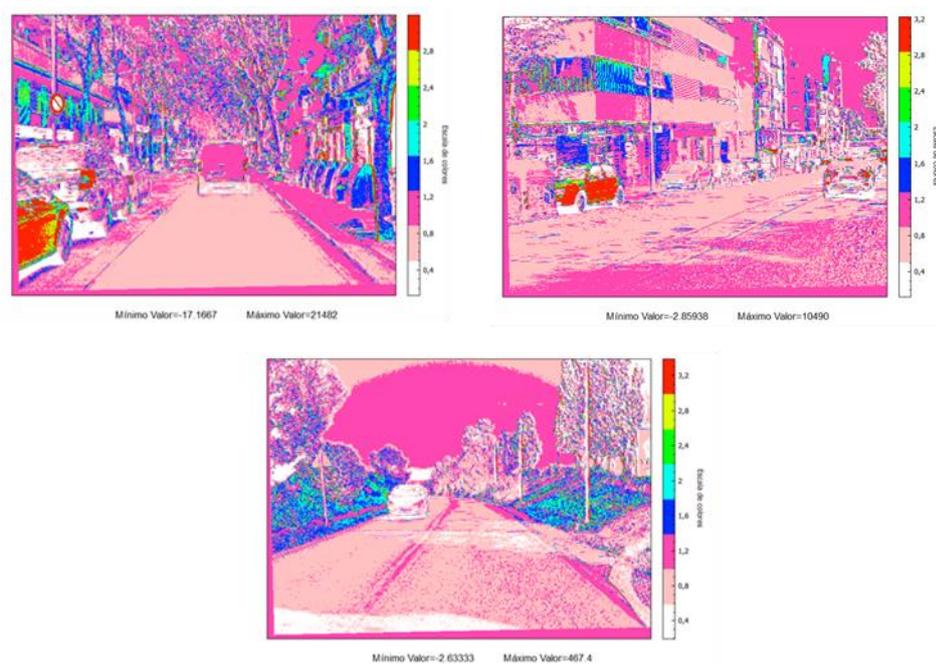


Figura 5.12: Fusión multispectral (LWIR-VS) con el índice LWIR-ICA1.

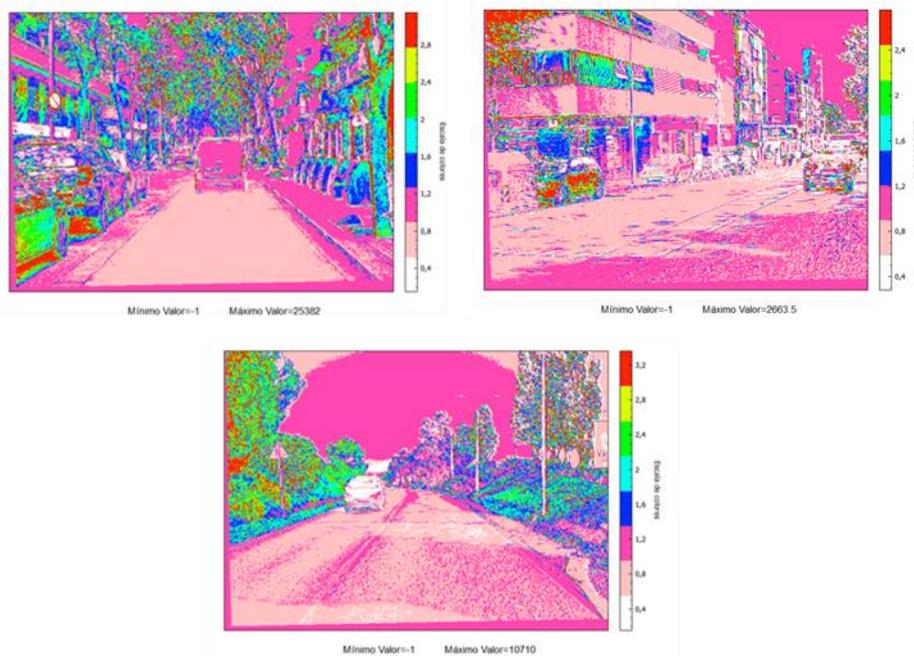


Figura 5.13: Fusión multispectral (LWIR-VS) con el índice LWIR-ICA2.

La Tabla 5 resume la información extraída de las imágenes resultantes de los índices de detección de calle. Aquí se muestran los valores mínimos y máximos.

Imagen LWIR	Imagen VS	Índice	Mínimo Valor	Máximo Valor
Rect_Infra21805	Rect_Visible_21810	LWIR-ICA1	-17.1667	21482
		LWIR-ICA2	-1	25382
Rect_Infra18335	Rect_Visible_18339	LWIR-ICA1	-2.85938	10490
		LWIR-ICA2	-1	2663.5
Rect_Infra2905	Rect_Visible_2900	LWIR-ICA1	-2.63333	467.4
		LWIR-ICA2	-1	10710

Tabla 5: Rangos de los índices de detección de calle.

Tanto el LWIR-ICA1 como el LWIR-ICA2 tienen un amplio rango de valores, que incluyen valores negativos y valores positivos muy altos.

Sin embargo, realizando el ajuste necesario de la escala de colores, se puede notar que se detecta la calle entre 0.6 a 0.95 en ambos índices, aunque vale resaltar que también se detectan otros objetos.

CONCLUSIONES

Este trabajo se realiza con el objetivo de analizar cómo trabaja el algoritmo ICP y cómo éste afecta en la fusión multiespectral, específicamente entre una imagen en el infrarrojo lejano y una imagen visible.

Se puede concluir que el ICP es un algoritmo eficiente y eficaz para realizar la correspondencia entre dos nubes de puntos. Además se puede notar en el pequeño estudio expuesto anteriormente, que a medida que se aumenta el número de pares de las nubes, el error total del ICP disminuye.

Otro de los objetivos era hallar índices de detección. Efectivamente, se ha logrado hallar índices para detectar la vegetación, el cielo y las calles, aunque éstos índices podrían estar sujetos a errores, ya que se desconocen los rangos en los que cada índice se encuentra.

Los índices ya existentes para detectar vegetación, no resultaron útiles para detectar objetos, pero sin duda alguna sirvió como un punto de partida para obtener los nuevos índices.

En base a las imágenes, el índice de detección de vegetación que muestra mejores resultados es el LWIR-IV1.

Referente al índice de detección de cielo, todos los índices presentados permiten una excelente detección.

Finalmente, para la detección de las calles, ambos índices no resultan eficaces, ya que además de detectar lo requerido, también se detectan otros objetos. A pesar de esto, resulta útil conocer que para este caso no se utilizó una fórmula lineal, puesto que no se obtenían resultados óptimos, por ende se procedió a usar la potencia de uno de los canales y la multiplicación de otros dos.

RECOMENDACIONES

1. Tras la experiencia obtenida en las pruebas, se puede utilizar otra metodología para normalizar los valores de los índices o elegir otra forma de visualización para la detección de características.
2. Respecto al ICP, es recomendable tener las nubes de puntos con al menos 7 puntos y es preferible que tampoco sea una cantidad excesiva de puntos, ya que si bien es cierto el error total disminuye, también es cierto que el proceso resulta costoso computacionalmente.
3. El valor del umbral seleccionado en este estudio es de 0.000001, porque es relativamente bajo para que algoritmo encuentre la transformación correcta para lograr la fusión. Este valor también puede ser disminuido y puede generar otros estudios.

ANEXOS

Anexo 1

Métodos de Fusión para los índices de vegetación

Los índices de vegetación NDVI, RVI y ENDVI han intercambiado el valor del NIR por el LWIR.

Las imágenes utilizadas para las pruebas han sido:

VS: Rect_Visible_2180.bmp

LWIR: Rect_Infra21804.jpg

Índice NDVI

$$NDVI = \frac{LWIR - RED}{LWIR + RED}$$

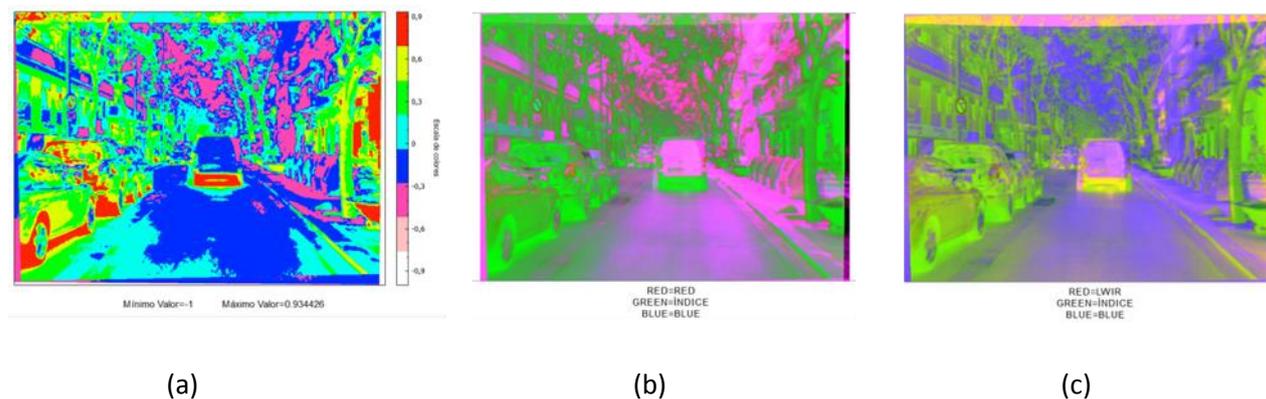


Figura A.1: Métodos de Fusión con el índice NDVI.

(a) Paleta de colores. (b) Método de fusión #1. (c) Método de fusión #2.

Índice RVI

$$RVI = \frac{LWIR}{RED}$$

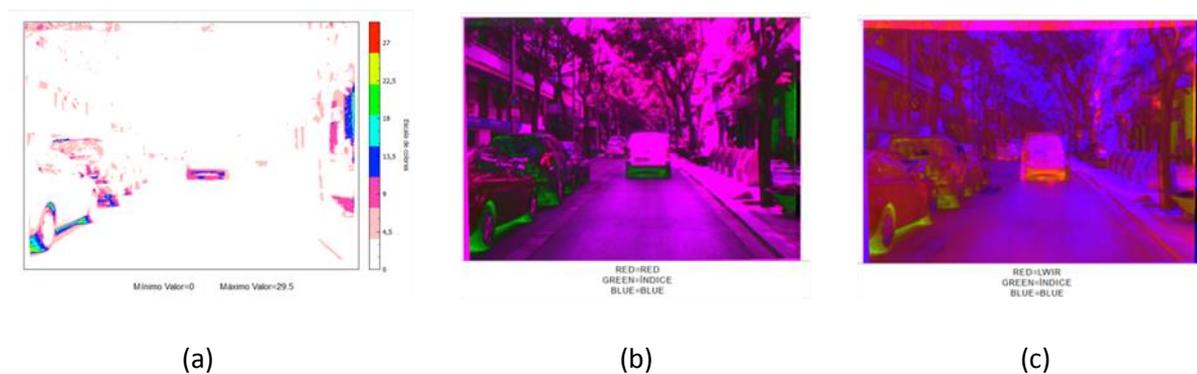


Figura A.2: Métodos de Fusión con el índice RVI.

(a) Paleta de colores. (b) Método de fusión #1. (c) Método de fusión #2.

Índice ENDVI

$$ENDVI = \frac{LWIR + GREEN - 2 * BLUE}{LWIR + GREEN + 2 * BLUE}$$

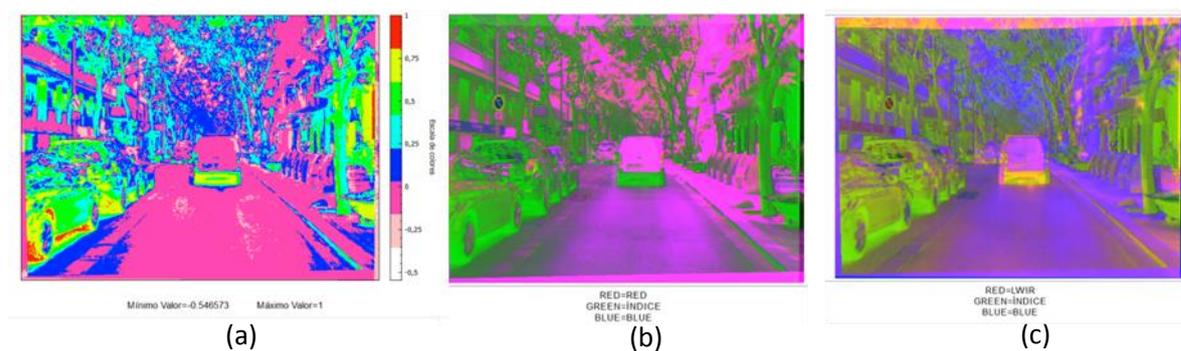


Figura A.3: Métodos de Fusión con el índice ENDVI.

(a) Paleta de colores. (b) Método de fusión #1. (c) Método de fusión #2.

Glosario

Datos Estructurados

Suponen que los puntos están coordinados, o sea por cada punto, los vecinos son conocidos. Las aplicaciones que utilizan registros 3D se basan en imágenes de rango, nubes de puntos o mallas de triángulos como datos de entrada.

En las imágenes de rango, los puntos vecinos en el “source” también son puntos vecinos sobre la superficie. La nube de puntos puede ser estructurada por algoritmos de triangulación para obtener una malla de triángulos, donde los puntos de un triángulo son vecinos.

Malla triangular

Es una colección de triángulos y vértices que aproximan una superficie en 3D.

Una utilidad de las mallas de triángulos podría ser para sistemas de reconocimiento de objetos, la compresión de una escena y las características del modelado.

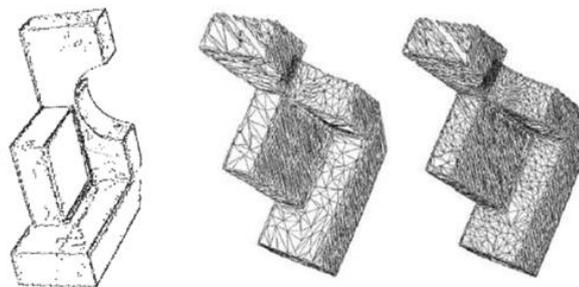


Figura G.1: Malla triangular.

Neighbor Search

Es una modificación del algoritmo de correspondencia. Los vecinos de cada punto tienen que ser conocidos en ambas nubes. Por lo tanto se necesita de datos estructurados. Si el punto correspondiente a m_k es p_k es conocido y p_k se encuentra en un vecindario V' de p_i , entonces se asume que el correspondiente punto m_k está en el vecindario V' de m_i .

La complejidad del algoritmo se reduce a $\mathcal{O}(N_p)$, porque sólo para el primer punto, la nube modelo completa ha sido considerada.

Dado que se comprueba sólo un pequeño número de puntos, se puede producir falsa coincidencia. Al aumentar el tamaño de V' , la correspondencia exitosa es mejorada, pero al mismo tiempo aumenta el tiempo de cálculo.

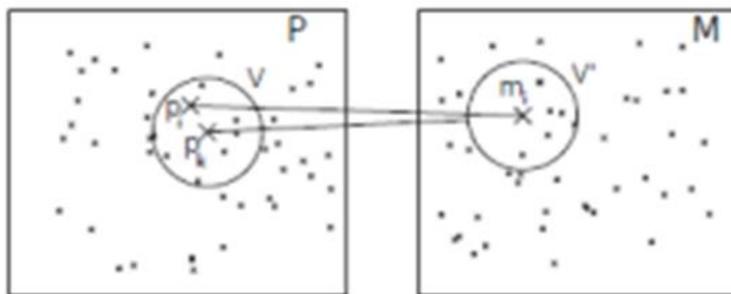


Figura G.2: Neighbor Search.

Solución Closed-Form

Una expresión “closed-form” es una expresión matemática que puede ser evaluada en un número finito de operaciones. Ésta debería contener constantes, variables, ciertas operaciones bien conocidas y funciones.

Resuelve un problema dado en términos de funciones y operaciones matemáticas elegidas de un conjunto limitado y generalmente aceptado

Kd-tree

Un árbol Kd es un árbol binario de búsqueda en el que cada nodo representa una partición del espacio de dimensión d , para crear las divisiones en el espacio en cada nodo sólo una de las k dimensiones es utilizada como discriminador.

El nodo raíz representa el espacio entero y las hojas representan subespacios que contienen subconjuntos del grupo de puntos de entrada, los cuales son mutuamente excluyentes.

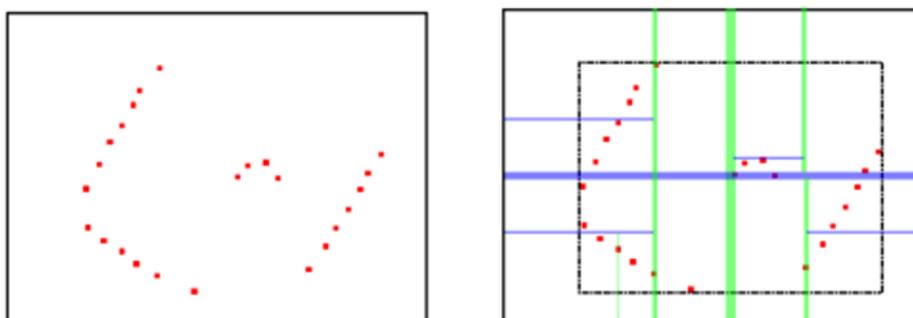


Figura G.3: Kd-tree.

Linearización

La linearización se refiere al proceso eficaz de encontrar la aproximación lineal a una función en un punto dado.

Bibliografía

- [1] Besl P.J. and McKay N., "A Method for Registration of 3-D Shapes", Pattern Analysis and Machine Intelligence (PAMI), Vol. 14 No. 2, 1992.
- [2] Turk G. and Levoy M., "Zippered Polygon Meshes from Range Images", Proc. SIGGRAPH, 1994.
- [3] Godin G., Rioux M. and Baribeau R., "Three-dimensional Registration Using Range and Intensity Information", Proc. SPIE: Videometrics III, Vol. 2350, 1994.
- [4] Masuda T., Sakaue K. and Yokoya N., "Registration and Integration of Multiple Range images for 3-D Model Construction", Proc. CVPR, 1996.
- [5] Weik S., "Registration of 3-D Partial Surface Models Using Luminance and Depth Information", Proc. 3D Digital Imaging and Modeling (3DIM), 1997.
- [6] Rusinkiewicz S. and Levoy M., "Efficient variants of the ICP algorithm", Proc. 3D Digital Imaging and Modeling (3DIM), 2001.
- [7] Blais G. and Levine M., "Registering Multiview Range Data to Create 3D Computer Objects", Trans. PAMI, Vol. 17 No. 8, 1995.

- [8] Neugebauer P., "Geometrical Cloning of 3D Objects via Simultaneous Registration of Multiple Range Images", Proc. SMA, 1997.
- [9] Dorai C., Weing J. and Jain A., "Registration and Integration of Multiple Object Views for 3D Model Construction", Trans. PAMI, Vol.20 No. 1, 1998.
- [10] Chen Y. and Medioni G., "Object Modeling by Registration of Multiple Range Images", Proc. IEEE Conf. on Robotics and Automation, 1991.
- [11] Simon, "Fast and Accurate Shape-Bases Registration, Ph. D. Dissertation, Camegie Mellon University, 1996.
- [12] Wild Michael, "Recent Development of the Iterative Closest Point (ICP) Algorithm", URL: http://students.asl.ethz.ch/upl_pdf/314-report.pdf, fecha de consulta: marzo del 2014.
- [13] Halchenko Yaroslav, "Iterative Closest Point (ICP)", URL: <http://www.onerussian.com/classes/cis780/icp-slides.pdf>, fecha de consulta: marzo del 2014.
- [14] Mitra Niloy, "Dynamic Geometry Processing", URL: <http://resources.mpi->

inf.mpg.de/deformableShapeMatching/EG2012_Tutorial/slides/1.2%20CP_+_TPS_%28NM%29.pdf, fecha de consulta: junio del 2014.

[15] Colas Francis, "Iterative Closest Point Algorithm", URL: <http://www.asl.ethz.ch/education/master/info-process-rob/ICP.pdf>, fecha de consulta: julio del 2014.

[16] Ruiz Luis, "Métodos Básicos de fusión de imágenes", URL: http://riunet.upv.es/bitstream/handle/10251/30096/Fusion_imagenes.pdf?sequence=1, fecha de consulta: septiembre del 2014.

[17] "OpenCV Documentation", URL: <http://docs.opencv.org/2.4.7/index.html>, fecha de consulta: diciembre del 2013.

[18] "LIBICP: C++ Library for Iterative Closest Point Matching", URL: <http://www.cvlibs.net/software/libicp/>, fecha de consulta: febrero del 2014.

[19] "Vegetation Indices", URL: <http://books.google.com.ec/books?id=PqgQZG8DvNcC&pg=PA25&lpg=PA25&dq=RVI+McVey&source=bl&ots=ZjFLXHHHtV&sig=uUIYRp65HPk-DX7jX-GDZoNTt-Y&hl=es&sa=X&ei=PLZhVJD4HoynNujpg8gN&ved=0CBwQ6AEwAA#>

v=onepage&q=RVI%20McVey&f=false, fecha de consulta: octubre del 2014.

[20] “Índices de vegetación”, URL:
http://www.biblioteca.udep.edu.pe/BibVirUDEP/tesis/pdf/1_59_185_24_508.pdf, fecha de consulta: octubre del 2014.

[21] “Closed-Form Solution”, URL:
<http://mathworld.wolfram.com/Closed-FormSolution.html>, fecha de consulta: octubre del 2014.

[22] “Affine Transformations”, URL:
http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/warp_affine/warp_affine.html, fecha de consulta: octubre del 2014.

[23] “The Method of Least Squares” URL:
http://web.williams.edu/Mathematics/sjmillier/public_html/BrownClasses/54/handouts/MethodLeastSquares.pdf, fecha de consulta: octubre del 2014.