



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

**ESCUELA ACADÉMICA PROFESIONAL DE INGENIERIA
DE SISTEMAS**

TESIS

**STITCHING DE IMÁGENES DIGITALES TOMADAS POR
UN VEHÍCULO AÉREO NO TRIPULADO (UAV)**

**PARA OPTAR POR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

Autor(es):

Bach. Ruiz Ordinola José Antonio

Asesor:

Mg. Tuesta Monteza Víctor Alexci

**Línea de Investigación:
Ciencias de la Computación**

**Pimentel – Perú
2018**

**STITCHING DE IMAGENES DIGITALES TOMADAS POR UN VEHICULO
AEREO NO TRIPULADO (UAV)**

Aprobación del informe de investigación

**Dr. Gutiérrez Gutiérrez Jorge Luis
Presidente del jurado de tesis**

**Ing Mejía Cabrera Heber Iván
Secretario del jurado de tesis**

**Mg Tuesta Monteza Víctor Alexci
Vocal del jurado de tesis**

INFORMACIÓN GENERAL

1.1 Título del Informe de Investigación:

“STITCHING DE IMAGENES DIGITALES TOMADAS POR UN
VEHICULO AEREO NO TRIPULADO (UAV)”

1.2 Línea de Investigación:

Ciencias de la computación

1.3 Autor:

Ruiz Ordinola Jose Antonio

1.4 Asesor Metodólogo:

Mg Vidaurre Flores Miguel Angel

1.5 Tipo y diseño de investigación.

Tipo experimental, metodología cuantitativa.

1.6 Facultad y Escuela Académico Profesional:

Facultad de Ingeniería, Arquitectura y Urbanismo
Escuela Profesional de Ingeniería de Sistemas

1.7 Periodo: 2016-II

1.8 Fecha de inicio y término del proyecto:

Abril – Diciembre 2016

1.9 Firma de los autores del proyecto:

Ruiz Ordinola Jose Antonio
AUTOR

1.10 Aprobado:

Mg Vidaurre Flores Miguel Angel
ASESOR METODÓLOGO

Mg. Tuesta Monteza Víctor Alexci
ASESOR ESPECIALISTA

1.11 Fecha de Presentación:

2018

DEDICATORIA

A mis padres.

A mis padres por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.

Todo este trabajo ha sido posible gracias a ellos

.

A mis maestros.

Por su gran apoyo y motivación para la culminación de nuestros estudios profesionales y para la elaboración de esta tesis

Ruiz Ordinola José Antonio

AGRADECIMIENTO

A Dios: .

Por estar con nosotros en cada paso que damos por guiarnos y por fortalecer a nuestro corazón e iluminar nuestra mente durante todo el periodo de estudio, gracias a él pude lograr uno de mis mas grandes objetivos.

.

A mi familia.

A mi familia que siempre me apoyo de manera incondicional y fueron el motor para impulsarme a terminar con éxito el presente proyecto de tesis.

INDICE

RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN	3
1.CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN	4
1.1. Situación Problemática.....	4
1.2. Formulación del problema	5
1.3. Delimitación de la investigación	5
1.4. Justificación e importancia de la investigación	5
1.5. Limitaciones de la Investigación	6
1.6. Objetivos de la investigación	6
1.6.1. Objetivos Generales	6
1.6.2. Objetivos Específicos	6
2.CAPÍTULO II: MARCO TEÓRICO	7
2.1. Antecedentes de estudios	7
2.2. Estado del arte	9
2.3. Base teórica científica	11
2.3.1. Imagen Digital	11
2.3.2. Elementos de un sistema de procesado de imágenes	13
2.3.3. Proceso de Stitching de Imágenes	13
2.3.4. Métodos y Técnicas de Stitching	14
2.3.5. Algoritmo SIFT.....	16
2.3.6. Algoritmo RANSAC (Random Sample Consensus)	22
2.3.7. Algoritmo de Levenberg-Marquardt.....	26
2.3.8. Herramientas	27
2.4. Definición de los términos básicos.....	29
3.CAPÍTULO III: MARCO METODOLÓGICO.....	32
3.1. Tipo de investigación	32
3.1.1. Tipo de estudio.....	32
3.1.2. Diseño de estudio	32
3.2. Población y Muestra.....	32
3.2.1. Población.....	32
3.2.2. Muestra	32
3.3. Hipótesis.....	32
3.4. Variables.....	33
3.4.1. Variables independientes:	33
3.4.2. Variable dependiente:	33

3.5.	Operacionalización.....	33
3.6.	Abordaje metodológico, técnicas e instrumentos de recolección de datos	34
3.7.	Procedimiento para la recolección de datos	35
3.8.	Análisis estadístico e interpretación de los datos	36
3.9.	Principios éticos	36
3.10.	Criterios de rigor científico.....	37
4.	CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS.....	39
4.1.	Resultados en tablas y gráficos	39
4.2.	Discusión de resultados	44
5.	CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN	48
5.1.	Adquisición imágenes	48
5.2.	Determinar algoritmos para procesamiento de imágenes	53
5.3.	Aplicar los Algoritmos de Stitching.....	60
5.4.	Optimizar la imagen panorámica	76
6.	CAPÍTULO VI: CONSIDERACIONES FINALES Y RECOMENDACIONES	82
6.1.	Conclusiones	82
6.2.	Recomendaciones.....	82
6.3.	Referencias	83

INDICE DE FIGURAS

<i>Figura 1.</i> El proceso de cálculo de la matriz de transformación afín para una imagen arbitraria.....	15
<i>Figura 2.</i> Secuencia de la imagen original (6 imágenes)	16
<i>Figura 3.</i> El panorama obtenido por el método de costura tradicional	16
<i>Figura 4.</i> Fuente (Lowe, 2004)	17
<i>Figura 5.</i> Fuente (Lowe, 2004)	19
<i>Figura 6.</i> Fuente (Lowe, 2004)	19
<i>Figura 7.</i> A la izquierda: la ventana Gaussiana pondera los valores de módulo y fase en la vecindad de los puntos de interés. A la derecha: los histogramas con 8 direcciones posibles realizados para cada subregión. (Lowe, 2004)	20
<i>Figura 8.</i> El pseudo código para implementar el algoritmo SIFT paralelo.....	21
<i>Figura 9.</i> Proceso del algoritmo SIFT. (Zanchettin & Cenedese, 2012-2013)	22
<i>Figura 10.</i> Escenario problemático implementando SIFT.	24
<i>Figura 11.</i> Análisis del Algoritmo RANSAC.	24
<i>Figura 12.</i> Pseudocódigo Algoritmo RANSAC.....	25
<i>Figura 13.</i> Pseudocódigo Algoritmo RANSAC.....	25
<i>Figura 14.</i> Después de realizar el procesamiento de RANSAC.....	26
<i>Figura 15.</i> Es una imagen donde se grafica la Tabla 02. Fuente: Elaboración Propia.....	39
<i>Figura 16.</i> . Es una imagen donde se grafica la Tabla 03. Fuente: Elaboración Propia.....	40
<i>Figura 17.</i> Es una imagen donde se grafica la Tabla 04. Fuente: Elaboración Propia.....	41
<i>Figura 18.</i> Es una imagen donde se grafica la Tabla 05. Fuente: Elaboración Propia.....	41
<i>Figura 19.</i> Es una imagen donde se grafica la Tabla 06. Fuente: Elaboración Propia.....	43
<i>Figura 20.</i> Es una imagen donde se grafica la Tabla 07. Fuente: Elaboración Propia.....	43
<i>Figura 21.</i> UAV Utilizado para las capturas de las imágenes digitales. Fuente:Elaboracion Propia.....	48
<i>Figura 22.</i> Imagen “A” del primer conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia	49
<i>Figura 23.</i> Imagen “B” del primer conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.	49
<i>Figura 24.</i> Imagen “C” del primer conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.	50

<i>Figura 25.</i> Imagen “W” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.	50
<i>Figura 26.</i> Imagen “X” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia	51
<i>Figura 27.</i> Imagen “Y” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia	51
<i>Figura 28.</i> Imagen “Z” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.	52
<i>Figura 29.</i> Descriptor SURF puntos de interés. (Bay, Tuytelaars, & Van Gool)	56
<i>Figura 30.</i> Las entradas de descripción de una subregión representan la naturaleza del patrón de intensidad subyacente. (Bay, Tuytelaars, & Van Gool)	56
<i>Figura 31.</i> Visión general del Algoritmo ASIFT. (Morel & Yu, 2011)	56
<i>Figura 32.</i> Interpretación de la formula geométrica de ASIFT. (Morel & Yu, 2011)	57
<i>Figura 33.</i> Definiciones utilizadas matemáticamente. (J. Matas, O. Chum, M. Urban, & T. Pajdla, 2002).....	57
<i>Figura 34.</i> Descripción grafica "Feature Descriptor. (Yuan's, 2013).....	61
<i>Figura 35.</i> Matriz de puntos de Surf. (Yuan's, 2013).....	62
<i>Figura 36.</i> El algoritmo SURF implementado Visual Studio y OpenCV. Fuente: Elaboración Propia.	63
<i>Figura 37.</i> Resultado del proceso del algoritmo SURF de Imagen A e Imagen B. Fuente: Elaboración Propia.	64
<i>Figura 38.</i> Resultado del proceso del algoritmo SURF de Imagen B e Imagen C. Fuente: Elaboración Propia	64
<i>Figura 39.</i> Resultado del proceso del algoritmo SURF de Imagen W e Imagen X. Fuente: Elaboración Propia.	65
<i>Figura 40.</i> Resultado del proceso del algoritmo SURF de Imagen X e Imagen Y. Fuente: Elaboración Propia.	65
<i>Figura 41.</i> Resultado del proceso del algoritmo SURF de Imagen Y e Imagen Z.	65
<i>Figura 42.</i> El algoritmo SIFT implementado Visual Studio y OpenCV. Fuente: Elaboración Propia.	68
<i>Figura 43.</i> Resultado del proceso del algoritmo SIFT de Imagen A e Imagen B. Fuente: Elaboración Propia.	69

<i>Figura 44.</i> Resultado del proceso del algoritmo SIFT de Imagen B e Imagen C. Fuente: Elaboración Propia.	69
<i>Figura 45.</i> Resultado del proceso del algoritmo SIFT de Imagen W e Imagen X. Fuente: Elaboración Propia.	70
<i>Figura 46.</i> Resultado del proceso del algoritmo SIFT de Imagen X e Imagen Y. Fuente:Elaboracion Propia.	70
<i>Figura 47.</i> Resultado del proceso del algoritmo SIFT de Imagen Y e Imagen Z. Fuente:Elaboracion Propia.	70
<i>Figura 48.</i> Escenario problemático implementando SIFT. Fuente: Elaboración Propia	71
<i>Figura 49.</i> Análisis del Algoritmo RANSAC. (Zanchettin & Cenedese, 2012-2013).....	72
<i>Figura 50.</i> El algoritmo RANSAC implementado Visual Studio y OpenCV PARTE 1. Fuente: Elaboración Propia	72
<i>Figura 51.</i> El algoritmo RANSAC implementado Visual Studio y OpenCV PARTE 2. Fuente: Elaboración Propia.	73
<i>Figura 52.</i> Resultado del proceso del algoritmo RANSAC de Imagen A e Imagen B. Fuente: Elaboración Propia.	74
<i>Figura 53.</i> Resultado del proceso del algoritmo RANSAC de Imagen B e Imagen C. Fuente: Elaboración Propia.	74
<i>Figura 54.</i> Resultado del proceso del algoritmo RANSAC de Imagen W e Imagen X. Fuente: Elaboración Propia.	75
<i>Figura 55.</i> Resultado del proceso del algoritmo RANSAC de Imagen X e Imagen Y. Fuente: Elaboración Propia.	75
<i>Figura 56.</i> Resultado del proceso del algoritmo RANSAC de Imagen Y e Imagen Z. Fuente: Elaboración Propia.	76
<i>Figura 57.</i> Formula y diseño de funciones de L-M. (Chunyang Zhao & Huaici Zhao, 2016).....	77
<i>Figura 58.</i> El algoritmo implementado en Visual Studio y OpenCV. Fuente: Elaboración Propia.....	79
<i>Figura 59.</i> Imagen Panorámica optimizada con el algoritmo L-M. Fuente:Elaboracion Propia.....	80
<i>Figura 60.</i> Imagen Panorámica optimizada con el algoritmo L-M. Fuente:Elaboracion Propia.....	81

INDICE DE TABLAS

Tabla 1 Esta tabla muestra la resolución de imagen mínima recomendada al imprimir en diferentes tamaños y calidad (Montabone, 2010).....	12
Tabla 2 Número de puntos “Matching” en el “PRIMER CONJUNTO DE IMAGENES”	39
Tabla 3 Muestra medida por Número de puntos “Matching” en el “SEGUNDO CONJUNTO DE IMAGENES”	40
Tabla 4 Muestra medida por Número de puntos “Matching” en el “PRIMER CONJUNTO DE IMAGENES”	40
Tabla 5 Muestra medida por Número de puntos “Matching” en el “SEGUNDO CONJUNTO DE IMAGENES”	41
Tabla 6 Muestra medida en Tiempo de Stitching.....	42
Tabla 7 Muestra medida en Tiempo de Stitching.....	43
Tabla 8 Muestra medida en Tiempo de Stitching del Primer conjunto de Imágenes.....	44
Tabla 9 Muestra medida en Tiempo de Stitching del Segundo conjunto de imágenes.....	44
Tabla 10 The comparison data of distortion degree between the traditional method and the improved method.....	45
Tabla 11 Experimental data in image registration process.....	46

RESUMEN

En este trabajo de investigación analizamos las técnicas para el proceso de Stitching de imágenes digitales que incluyen algoritmos que implementan modelos matemáticos y matrices con la finalidad de obtener una imagen panorámica con menor distorsión; se utilizó el método que consta: en primer paso la obtención de las imágenes, a las cuales se le realizó la conversión de las imágenes a puntos mediante el algoritmo Sift y Surf, luego la respectiva correspondencia de puntos usando el Algoritmo Ransac y se optimiza el resultado mediante el algoritmo L-M.

Todo esto se implementó utilizando las herramientas OpenCV y Visual Studio en el lenguaje de programación C++.

Como resultado tuvimos que el porcentaje de información útil en la imagen panorámica alcanzado es de 83,197 %, la validación se realizó usando fórmulas matemáticas y matrices. También se evaluó los puntos Matching en los diferentes algoritmos implementados la validación se realizó mediante el uso de tablas y gráficos. Los algoritmos determinados para el procesamiento de imágenes en esta investigación aumentaron el número de puntos Matching, estos algoritmos SURF, SIFT, RANSAC Y L-M mejoraron la imagen panorámica.

PALABRAS CLAVE:

Procesamiento de Stitching de Imagen, Puntos de coincidencia, Errores de distorsión, Variable invariante.

ABSTRACT

This research presents an analysis of the techniques for processing digital images using Stitching methods and algorithms in order to obtain a panoramic image with less distortion; to implement the programming language C ++ was used; which are based on point conversion techniques mainly use Sift algorithm and Surf in second case also points matching techniques using RANSAC Algorithm

the percentage of useful information in the panoramic image is reached 83,197 % , validation was performed using mathematical formulas and matrices.

The problem actually responds to the fact that there are many method for panoramic images in a particular sector or group of images, and the majority of cases leads even costly investment of money, so an investigation is presented using various techniques image processing for obtaining panoramic images in our reality, this is considered Stitching process images.

KEY WORDS:

Image Stitching processing, Matching points, Distortion Errors, Variable invariant

INTRODUCCIÓN

La obtención de imágenes panorámicas de un determinado sector ya sea geográfica o de grupos de imágenes con el fin de obtener información para un fin en específico se tiene que recurrir a elementos muy costosos como son satélites, torres de control, globo aerostático, aeronaves pilotadas, ect.

Si optamos por el proceso de Stitching de imágenes el cual consiste en la costura de las imágenes digitales mediante métodos y algoritmos de procesamiento de imágenes y así obtener una imagen panorámica. Reduciendo el costo de inversión y obtener la información deseada.

Otro problema a considerar en las imágenes panorámica son los errores de distorsión, por el cual mediante algoritmo que implementan formulas matemáticas y métodos de selección de puntos de referencias en las imágenes poder reducir el error de distorsión.

Los resultados experimentales muestran que la técnica de proceso de (Lowe, 2004)Stitching implementado nos otorga un 83% de información útil en nuestra imagen panorámica obtenida.

Nuestra investigación se organiza por capitulos de la siguiente manera. Capitulo 1 Problema de la investigación, Capitulo 2 Marco teorico donde se describe las diferentes investigaciones sobre procesamiento de Stitching de imágenes digitales, también describe el algoritmo SIFT para la extracción de características de la imagen y el algoritmo RANSAC para la purificación de la característica coincidente puntos Matching, Capitulo 3 Marco Metodologico, Capitulo 4 muestra los resultados experimentales y el análisis cuantitativo para evaluar las tecnicas y métodos investigados, Capitulo 5 Propuesta de Investigación contiene todos los métodos, algoritmos, programación para el proceso de unión de imágenes, Capitulo 6 actúa como la conclusión de nuestra investigación.

1. CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Situación Problemática

El apogeo de la tecnología digital se ha convertido en un medio de apoyo para los diferentes sectores como son la medicina, agricultura, arquitectura, etc siendo las fotografías una herramienta eficaz para lograr coherente interpretación, análisis y diagnóstico en los distintos sectores implementados.

Para obtener imágenes remotas de la superficie de la Tierra se realizan mediante satélites, torres de control, globo aerostático, aeronaves pilotadas y etc. pero estos instrumentos son muy costosos por lo cual se realiza el uso de las plataformas no tripuladas flexibles, tales como vehículos aéreos no tripulados (UAV) de bajo costo.

En la investigación “Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images”. (Candiago, Remondino, De la Giglio, Dubbini, & Gattelli, 2015) Nos demuestra el gran potencial de las imágenes tomadas por UAV de alta resolución y técnicas de fotogrametría aplicada en el marco de la agricultura exactamente en los cultivos (viñedos y tomates) para recoger imágenes multiespectrales y evaluar de manera rápida y fiable.

Donde implementando una mayor automatización permitirá localizar y orientar al UAV en la captura de imágenes de diferentes alturas y distintas direcciones lo que hace posible trabajar con varios tipos de imágenes (por ejemplo visibles, infrarrojos y / o térmicas) mediante cualquier tipo de cámara (calibrada o no calibrada) lo cual ofrece una gran exactitud y fiabilidad en los resultados.

Por lo cual surge no solo obtener como resultado una imagen, ni varias imágenes distintas sino una imagen panorámica con la cual le se debe implementar métodos y algoritmo para el proceso de “Stitching” de las imágenes, el cual consiste en la costura de dichas imágenes, donde nace un obstáculo que se considera como “error de distorsión de la imagen panorámica” durante el proceso de stitching de las imágenes.

Este obstáculo surge mayormente cuando se desea obtener una imagen panorámica de un lugar geográfico, área de cultivo, espacios amplios, entre otros. Entonces se debe aplicar métodos los cuales modifican el proceso de Stitching de imágenes por ejemplo en el paper “The Improved Algorithm of Fast Panorama Stitching for Image Sequence and Reducing the Distortion Errors” (Zhong Qu, Si-Peng Lin, Fang-Rong Ju, & Ling Liu, 2015) en el cual nos demuestra como implementa un algoritmo con el cual reduce las

distorsiones en el proceso de Stitching de imágenes y hace una comparación entre proceso tradicional y su implementación de su método.

El presente trabajo de investigación pretende realizar un Stitching de imágenes digitales tomadas por un vehículo aéreo no tripulado.

1.2. Formulación del problema

¿ Que algoritmos permitirán realizar stitching de imágenes digitales de manera eficiente y reduciendo los errores de distorsión?

1.3. Delimitación de la investigación

- a. Se realizará en el laboratorio de Investigación de Sistemas ubicado en la Universidad Señor de Sipan.
- b. Se usara una Base de Datos externa para la obtencion de imágenes digitales.
- c. Realizado en el año 2016.

1.4. Justificación e importancia de la investigación

En lo personal el resultado obtenido será una imagen panorámica con mayor reducción de errores de distorsión, el cual será un gran apoyo para el análisis y evaluación de dicha área.

En lo social la imagen panorámica obtenida por un UAV es importante hoy para las empresas y/o organizaciones lo cual le permitirá visualizar un determinado área para su respectivo análisis y evaluación.

En lo tecnológico el registro de imágenes es un paso importante en proceso de “STITCHING” de imágenes. La calidad del stitching de la imagen en gran medida depende de la precisión de registro de imágenes. Por eso hay diferentes métodos para el registro de imagen, los algoritmos de costura de imágenes tradicionales y la mejora de este proceso con lo que se desea la obtención de imágenes menos distorsionada.

En lo económico esta investigación muestra una forma menos costosa de obtener una imagen panorámica mediante el uso de UAV y el proceso de STITCHING de las imágenes.

1.5. Limitaciones de la Investigación

La captura de imágenes digitales obtenidas por un UAV en un ambiente de vuelo no controlado.

1.6. Objetivos de la investigación

1.6.1. Objetivos Generales

Evaluar algoritmos de Stitching de Imágenes digitales tomadas por un vehículo aéreo no tripulado.

1.6.2. Objetivos Específicos

- a) Capturar imágenes digitales mediante el uso de UAV.
- b) Determinar los Algoritmos para el procesamiento de imágenes.
- c) Aplicar los algoritmos de STITCHING para unir las imágenes digitales.
- d) Optimizar imagen panorámica de imágenes digitales.

2. CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de estudios

(Zhong Qu, Si-Peng Lin, Fang-Rong Ju, & Ling Liu, 2015) En su paper “*The Improved Algorithm of Fast Panorama Stitching for Image Sequence and Reducing the Distortion Errors*” el objetivo es mejorar el proceso de stitching de imagen tradicional donde se implementa primer el registro de todas las imágenes en la secuencia y se calcula la matriz de transformación entre las imágenes y luego se toma la imagen del centro de la secuencia como la imagen de referencia, el método mejorado puede conseguir una imagen en cualquier lugar en la secuencia transforma en el espacio de la imagen de referencia de la coordenada. Luego mediante algoritmos como: algoritmo de Sift utilizado para la extracción de característica de la Imagen, algoritmo Ransac para purificar los puntos de características coincidentes y algoritmo de Levenberg-Marquardt que se utiliza para ajustar los parámetros en la matriz de transformación.

Este método mejorado reduce los errores de la distorsión del panorama y ahorra tiempo en el proceso de stitching de imágenes, además mejora la calidad del resultado del proceso.

Se puede observar 8 imágenes originales donde aplica el proceso tradicional de stitching y el nuevo proceso mejorado, donde se realiza una comparación de las imágenes panorámica obtenido por ambos procesos, notando una gran mejor en la calidad de la imagen panorámica aplicando el método mejorado.

En lo personal, en este paper nos explica detalladamente el proceso de stitching de imágenes, que aplicando funciones y modelos matemáticos reduce la distorsión, y mejora localidad de la imagen panorámica que deseamos obtener.

(Minchen Zhu, Weizhi Wang, Bingham Liu, & Jingshan Huang, 2013) En su paper “*Efficient Video Panoramic Image Stitching Based on an Improved Selection of Harris Corners and a Multiple Constraint Corner Matching*” el objetivo se ha centrado en algoritmos basados en características, que hacen uso de un pequeño número de invariantes puntos, líneas o bordes para alinear las imágenes, el proceso de stitching de las imágenes depende en gran medida de la precisión y solidez. En donde esta investigación nos presenta un nuevo algoritmo basado en funciones y está motivado por la necesidad de unir imágenes adyacentes. Tiene dos características primordiales una

selección de auto-adaptativa (self-adaptive) del esquinas de Harris y un matching de esquina múltiple restricción. Con las esquinas de Harris obtenemos baja complejidad computacional y alta robustez en las características para alinear las imágenes. Dicha selección auto-adaptativa (self-adaptive) garantiza que las esquinas puedan estar distribuidos de tal manera que haya una mayor selección de las regiones con la información de textura. Además, nuestro algoritmo evita la posible agrupación de las esquinas. Después se aplica una filtración de un gran número de esquinas inapropiadas de acuerdo a su posición de información, generamos un conjunto inicial de pares de matching de esquina basados en escalas de grises de las regiones adyacentes en cada esquina.

Se puede observar 3 imágenes originales donde se aplica su algoritmo, el cual es robusto a errores causado por el movimiento de la cámara, porque los valores óptimos de los parámetros se obtenerse de una manera auto-adaptativa (self-adaptive) durante la fusión de imágenes.

En lo personal, en este paper nos explica y se basa en los errores ocasionados por el movimiento de la cámara donde implementan su propio algoritmo para la solución de estos errores.

(Minchen Zhu, Weizhi Wang, Binghan Liu, Jingshan Huang, 2013) En su paper “*A Fast Image Stitching Algorithm vía Multiple-Constraint Corner Matching*” el objetivo es mostrarnos un nuevo algoritmo para manejar el proceso de stitching de un conjunto de datos de vídeo aérea a gran escala con el fin de obtener una imagen panorámica, Enfocándose en las pequeñas regiones, superpuestas y en proceso de stitching extremadamente lento. Consiste en matching de esquina múltiple restricción y un proceso de stitching de la imagen más eficiente. Para superar el ineficiente algoritmo Ransac tradicional, generado un conjunto inicial de los correspondientes pares de esquina basados en escalas de grises de las regiones adyacentes en cada esquina. Siendo capaces de reducir significativamente el número de iteraciones necesarias en el algoritmo Ransac, lo que resulta en un proceso de stitching de imágenes mucho más eficiente. Los resultados experimentales demostraron que el método nuevo aplicado es tres veces más rápido que el algoritmo Ransac tradicional y los panoramas generados a partir del nuevo algoritmo Ransac cuentan con una suave transición en la superposición de áreas de imagen y satisfacer los requerimientos visuales humanos.

En lo personal, en este paper nos explica cómo mejorar los métodos tradicionales para la proceso de stitching de imágenes pero aplica en los videos aéreos a gran escala para hacer que dichos procesos sean más rápidos y eficientes.

(Tao Yang, Jing Li, Jingyi Yu, Sibing Wang, & Yanning Zhang, 2015) En su “*Diverse Scene Stitching from a Large-Scale Aerial Video Dataset*” el objetivo se centra principalmente sobre métodos de sutura relacionadas con la vigilancia de vídeo aérea, y los métodos de sutura relacionados donde incluyendo el proceso de stitching de imagenes secuencial de recuperación. Puede utilizarse para estimar de manera eficiente la transformación de la imagen entre tramas consecutivas. El problema de stitching que presentan es al tratar de suturar las imágenes captadas en el video de vigilancia el cual debe resultar una imagen panorámica. Como soluciones se aplican método de unión de recuperación se basa en la característica de indexación. Aunque el uso de indexación es mucho menos caro, se requiere suficientes características de la imagen distintivos. Si la escena observada carece de suficiente único características, la varianza de diferentes imágenes no es suficiente para que el método para encontrar de forma fiable coincidencias correctas.

Como resultado, el método de recuperación de stitching puede encontrar todas las imágenes aéreas superpuestas de la misma escena y los rendimientos bajo rendimiento: por lo general alrededor del 80% de las concordancias correctas se pueden encontrar.

2.2. Estado del arte

(Jinwook Jeong & Kyungkoo Jun, 2016) En su paper “*A Novel Seam Finding Method Using Downscaling and Cost for Image Stitching*” se propuso una técnica basada en la estimación de movimiento de la cámara para mejorar el rendimiento de la inscripción. Algoritmo de constatación de la costura fue propuesta por el uso de parches de imagen obtenidos por el algoritmo gráfico de corte. El rendimiento de la constatación de la costura fue mejorada por la segmentación de imágenes basado en el algoritmo de cuencas. Técnica de programación dinámica fue capaz de mejorar tanto la velocidad y precisión a pesar de que trabajó en el nivel de píxel.

En este paper se propone un método de algoritmo de constatación de la costura para la etapa de mezcla, que es el segundo paso de la creación del panorama. Como objetivo tienen el mejorar el rendimiento preservando al mismo tiempo la precisión

del sitio de soldadura. Para obtener un rendimiento, hemos de bajar imágenes, cambiar el tamaño a la mitad de su tamaño, lo que resulta en menos tiempo de cálculo. La línea de unión que encontramos en las pequeñas imágenes de tamaño es sólo la aproximación de la línea de costura perfecta. Sobre la base de la aproximación, ya que la región en la que existe la línea de costura que se conoce, se puede limitar el área donde hallazgo costura debe ser realizada lugar de aplicarse a toda la zona. Esto resulta en un mejor rendimiento, mientras se mantiene la precisión.

(Jung-Hee Suk, Chun-Gi Lyuh, Sanghoon Yoon, & Tae Moon Roh, 2015). En su paper “*Fixed Homography-Based Real-Time SW/HW Image Stitching Engine for Motor Vehicles*” tiene como objetivo diseñar un software eficiente (SO) / hardware (HW) motor de fusión de imágenes para vehículos de motor que pueden hacer que las imágenes panorámicas para ampliar la distancia de detección de obstáculos y el área de la seguridad en tiempo real con una pequeña zona de HW. Mejorando las imágenes están deformadas sobre la superficie panorámica mediante la matriz homografía estimado (matriz H) entre la superficie y el panorama de coordenadas de imagen. Las imágenes panorámicas ofrecen a los usuarios de ancho escenas que no pueden ser capturados por una sola imagen de una cámara normal. Muchos algoritmos y sistemas comerciales para la costura de la imagen se ha informado. Los primeros movimientos del sistemas panorama asumidos en la cámara fija, como las rotaciones horizontales con ángulos fijos, utilizando interfaces de usuario con limitaciones. Esto simplifica los cálculos de una matriz de transformación, pero los grados de libertad para manejar imágenes panorámicas. En los algoritmos panorama, coincidente función y la estimación de transformación son los procedimientos más importantes ya que las imágenes se deforman espacialmente de las transformaciones resultantes. Brown y Lowe aprovechó una característica basada en el descriptor, cribar, para que coincida con las correspondencias de imágenes y movimientos de cámara estimación arbitraria de forma automática. características basadas en descriptor SIFT y SURF mejoran el rendimiento de la síntesis de panorama automático. Sin embargo, ya que la detección de características y función automática de juego tienen una alta carga computacional, estos enfoques no son adecuados para sistemas con poca potencia de cálculo.

2.3. Base teórica científica

2.3.1. Imagen Digital

Concepto

(Montabone, 2010) Es un conjunto de bits (1 y 0) que representa una imagen. Ese conjunto de bits se crea siguiendo un formato de imagen digital específico para que los diferentes programas puedan entenderlo. Hubo dos categorías principales de imágenes digitales:

a) Imágenes vectoriales

Este se forman mediante la ecuación matemática, tales como líneas, puntos, polígonos, y otros. Puesto que cada objeto en la imagen se define mediante una ecuación, haciendo cambios a la imagen es relativamente fácil; sólo los parámetros de la ecuación se cambian. Si desea cambiar el tamaño, cambiar el color, o mover los objetos, sólo se debe cambiar sus propiedades a los nuevos valores.

Este tipo de formato se utiliza comúnmente para los diseñadores gráficos para crear imágenes con un nivel de detalle no tan rico como una fotografía, pero con la capacidad de mantener su calidad en diferentes tamaños, tales como logotipos, iconos, fuentes, diagramas, y así sucesivamente,

Si usted está interesado en la edición de imágenes vectoriales, recomiendo que utilice "Inkscape", que es un excelente software libre para este fin (revisar anexo).

b) Imágenes de mapa de bits

Por otro lado, están representados con píxeles. Un píxel, también llamado un elemento de imagen, es la unidad más pequeña medible en imágenes de mapa de bits. Puede ser considerado como un punto de color. Cuando todos los píxeles de la imagen se analizaron conjuntamente, la imagen se percibe. Realizar cambios en este tipo de imágenes se realiza cambiando el color de los píxeles o cambiar el número de píxeles que representan la imagen.

Resolución

(Montabone, 2010) Para la impresión, se necesita generalmente una fotografía de alta resolución. El número recomendado de píxeles utilizados para imprimir la fotografía depende del tamaño de la impresión. Impresiones más grandes requieren resoluciones más grandes. Para impresiones de buena calidad, se necesita al menos 300 píxeles por pulgada (ppi). Por ejemplo, si desea imprimir una imagen de 6 MP, sería imprime bien en la ronda 9 x 7 pulgadas, aunque no es un tamaño de impresión común.

Para obtener una calidad aceptable, se necesita al menos 200 ppi. A menos de que imprimiría una foto mal definida. En el caso de grandes vallas publicitarias, que por lo general no puede obtener suficiente resolución para la impresión con alta calidad, sino porque se ven generalmente de muy lejos, puede salirse con la suya una imagen de menor resolución.

Tabla 1

Esta tabla muestra la resolución de imagen mínima recomendada al imprimir en diferentes tamaños y calidad (Montabone, 2010)

Common print sizes	Good quality	Acceptable quality
6 × 4 inches (15 × 10 cms)	2.2MP	1.0MP
7 × 5 inches (18 × 12 cms)	3.2MP	1.4MP
10 × 8 inches (25 × 20 cms)	7.2MP	3.2MP

El tamaño o la resolución de impresión es el aspecto que tendrán grandes o pequeñas las imágenes digitales en papel. Por defecto, la mayoría de las cámaras digitales establecen este 72 ppp, aunque esto no afecta a los píxeles de la imagen en absoluto. Este número es sólo para decirle a la impresora cuán grande o pequeña que desea imprimir su fotografía. Puede cambiar este número con facilidad en la mayoría del software moderno de procesamiento de imagen sin cambiar la resolución de la imagen.

2.3.2. Elementos de un sistema de procesamiento de imágenes

(Figueiras, 2002) De forma un tanto simplificada, pueden distinguirse tres grandes ámbitos dentro del procesamiento de imágenes:

1. mejora y acondicionamiento de imágenes para observación humana.
2. reconocimiento e interpretación automática o semi-automática de imágenes
3. gestión de información visual en formato digital.

Aunque cada ámbito se caracteriza por sus propias técnicas de procesamiento, la interrelación entre ellos es muy elevada: así, por ejemplo, los sistemas de visión artificial o de reconocimiento de imágenes suelen utilizar técnicas de mejora de imágenes en algunas etapas iniciales; también el indexado automático de imágenes características de los sistemas de interpretación; y hay otros muchos ejemplos.

2.3.3. Proceso de Stitching de Imágenes

Concepto

Según (Minchen Zhu, Weizhi Wang, Binghan Liu, & Jingshan Huang, 2013) nos dice proceso de stitching trata de combinar una secuencia de imágenes, que poseen mutuamente áreas superpuestas, lo que resulta en una imagen panorámica transparente.

La costura de imagen se ha convertido en un punto de acceso en el campo de la visión por computador, procesamiento de imágenes y gráficos por ordenador.

La calidad de la imagen depende en gran medida de la exactitud de registro de imágenes. Estos algoritmos de costuras se dividen:

- El registro de Imagen en relación de región: estudia la relación de los mismos bloques de cota entre la imagen de referencia y la imagen de entrada y calcula su grado de similitud.
- La imagen de registro basado en función: Utiliza modelos matemáticos para encontrar una descripción de las características de la importación útil de píxeles mediante las comparaciones de

características.

2.3.4. Métodos y Técnicas de Stitching

El método tradicional de costura

(Y.Xiong & K.Pulli, 2010) Se resume el metodo en los pasos siguientes.

Input “n”: se ordena la secuencia de imágenes S ($S_0, S_1, \dots, S_{(n-1)}$), que tienen áreas superpuestas entre sí:

- 1) Seleccionar las dos primeras imágenes del conjunto S para calcular los puntos de característica SIFT, respectivamente.
- 2) Seleccione la primera imagen de la secuencia S la imagen de referencia y la segunda imagen de la imagen de la nueva entrada. Utilice el algoritmo KNN (K Nearest-Neighbours) para buscar puntos de característica emparejan entre la nueva imagen de referencia de imagen de entrada y de acuerdo con la distancia euclidiana mínima.
- 3) De acuerdo con la característica de coincidencia de puntos conjunto de datos, utilice el algoritmo RANSAC para calcular la matriz afín H, que puede transformar la imagen de entrada nueva en el espacio de coordenadas de la imagen de referencia.
- 4) Usar el algoritmo de L-M (Levenberg-Marquardt) para optimizar la matriz afín H.
- 5) Utilice la H optimizado para transformar la imagen de entrada nueva.
- 6) Buscar la costura óptima entre el resultado transformación afín (5) y la imagen de referencia de la etapa; a continuación, a lo largo de la costura de combinarlos juntos sin problemas, obtener el resultado cosido imgResult.
- 7) Añadir la imgResult en S para reemplazar la imagen de referencia y la imagen de entrada. En el próximo proceso de costura del imgResult se selecciona la nueva imagen de referencia.
- 8) Volver a la etapa (1) para poner en práctica de forma continua el siguiente proceso de costura hasta que sólo hay una imagen existente en el S, que es el resultado panorámica.

Salida: Una imagen panorámica.

Sin embargo, en el método tradicional, cada vez que se completa una costura de la imagen, la dimensión de la imagen de referencia seguirá aumentando. El área de solapamiento entre la nueva imagen de referencia y la imagen de entrada representa la proporción más pequeña y más pequeña en comparación con la totalidad del área de la imagen de referencia. Por lo tanto, el cálculo de los puntos característicos SIFT de la imagen consumirá una gran cantidad de recursos del sistema y el tiempo de registro.

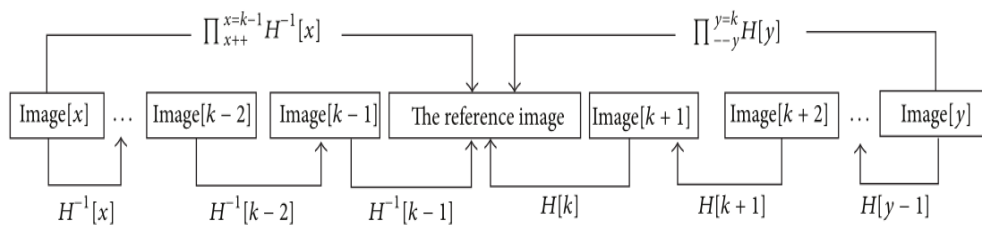


Figura 1. El proceso de cálculo de la matriz de transformación afín para una imagen arbitraria
(Y.Xiong & K.Pulli, 2010)

Además, el enfoque del algoritmo de costura tradicional a la selección de la imagen de entrada siguiente es relativamente sencillo. Cada vez que la imagen adyacente se selecciona la nueva imagen de entrada como, no da prioridad al tamaño de la zona de solapamiento o el número de punto característica coincidente imagen de referencia nueva imagen de entrada y el medio. Mientras tanto, si hay muchos errores que coinciden en proceso de registro de imágenes, los errores de concordancia se conservarán y afectan a todos los procesos de registro posterior debido a que la imagen de referencia está formado por los resultados anteriores cosidas. Finalmente, la imagen panorámica se verá seriamente distorsionada.

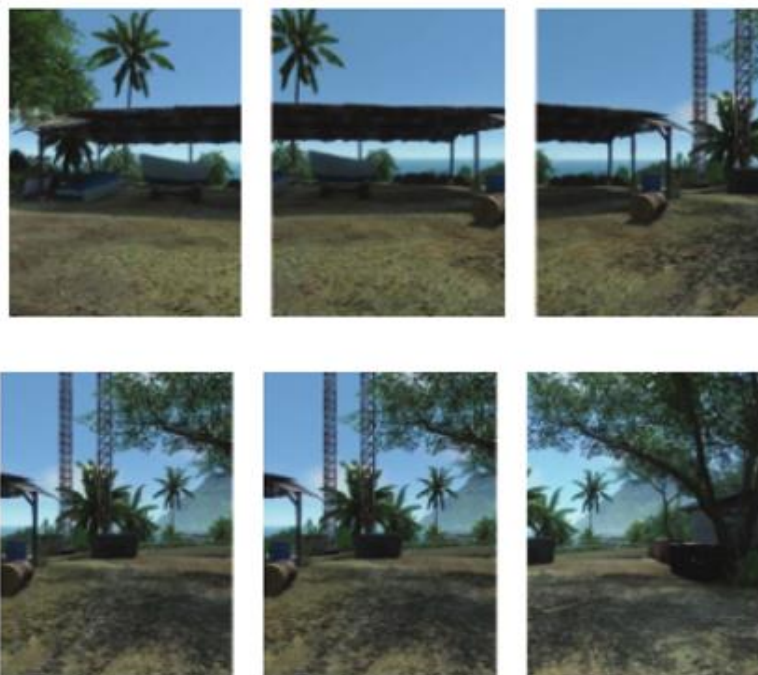


Figura 2. Secuencia de la imagen original (6 imágenes)
(Y.Xiong & K.Pulli, 2010)



Figura 3. El panorama obtenido por el método de costura tradicional
(Y.Xiong & K.Pulli, 2010)

2.3.5. Algoritmo SIFT

Concepto

(Lowe, 2004) Es un algoritmo de visión artificial publicado originalmente por David Lowe en 1999 que se encarga de extraer características distintivas de las imágenes en escala de grises.

Las características son invariantes a escala de la imagen y la rotación, y parcialmente invariantes a cambios en la iluminación y el punto de vista de la cámara. Ellos están bien localizadas en los dominios espacial y de

frecuencia, la reducción de la probabilidad de interrupción por la oclusión, el desorden, o ruido. Un gran número de características se pueden extraer de las imágenes típicas con algoritmos eficientes. Además, las características son altamente distintivo, que permite que una sola función, que se ajustará correctamente con alta probabilidad en contra de una gran base de datos de características, proporcionando una base para el reconocimiento de objetos y la escena.

El coste de la extracción de estas características se reduce al mínimo mediante la adopción de un enfoque de filtrado de cascada, en la que se aplican las operaciones más costosas solamente en lugares que pasan una prueba inicial.

A continuación se presentan las principales etapas de cálculo utilizado para generar el conjunto de características de la imagen:

Según (Plaza Cordero & Zambrano Martínez, 2012) :

Detección de puntos extremos luego de obtener una transformación espacio-escala (Scale-space extrema detection)

La idea es encontrar dentro del espacio-escala de la imagen puntos invariantes a la traslación, el escalado y la rotación de la misma. Además esos puntos deben ser mínimamente afectados por el ruido y pequeñas distorsiones. Estos puntos serán los puntos extremos (máximos o mínimos) obtenidos de las diferencias Gaussianas aplicadas en el espacio-escala de la imagen.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

$$L(x, y, \sigma) = I(x, y) * G(x, y, \sigma)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Figura 4. Fuente (Lowe, 2004)

- Función scale-space: $L(x, y, \sigma)$

Se debe realizar una búsqueda de los keypoints en todas las localizaciones de todas las escalas, para lo cual se usa la función continua $L(x, y, \vartheta)$, convolucionando la imagen $I(x, y)$ y la gaussiana.

$$L(x, y, \vartheta) = G(x, y, \vartheta) * I(x, y)$$

- Función Difference-of-Gaussian: $D(x, y, \vartheta)$

En esta etapa se debe obtener keypoints más estables, por lo tanto, ahora se trabajará con difference-of-gaussian, la cual es simplemente la resta de las imágenes vecinas pero que deben ser de la misma octava.

$$D(x, y, \vartheta) = [G(x, y, k\vartheta) - G(x, y, \vartheta)] * I(x, y)$$

$$D(x, y, \vartheta) = L(x, y, k\vartheta) - L(x, y, \vartheta)$$

Localización exacta punto clave (Local extrema detection)

Ahora lo que se deberá realizar es encontrar los máximos y mínimos locales, para lo cual cada pixel de cada imagen de la pirámide obtenida en el punto anterior, serán comparados con los nueve pixeles de la escala anterior y posterior, además con los ocho pixeles de la misma imagen.

La búsqueda de extremos en el espacio-escala produce múltiples candidatos entre los que se encuentran puntos con poco contraste; los cuales no son estables a cambios de iluminación y al ruido. Para quitarlos se procede de la siguiente manera.

Primero se estima la función diferencia Gaussiana entorno a un punto (x_0, y_0, σ_0) con una serie de Taylor de grado 2:

$$\bar{\chi} = -\frac{\partial^2 D^{-1}}{\partial \chi^2} \frac{\partial D}{\partial \chi} \quad (2)$$

Reemplazando (2) en (1) obtenemos el valor del máximo local:

$$D(\bar{\chi}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \chi} \bar{\chi}$$

Figura 5. Fuente (Lowe, 2004)

Finalmente si $|D(\chi)| < 0,03$ el punto es eliminado de la lista de puntos clave suponemos que D toma valores entre 0 y 1.

Asignación de orientación

Mediante la asignación de una orientación a cada punto de la imagen basada en características locales de la misma, los puntos clave pueden ser descriptos relativos a estas orientaciones y de esta manera lograr características invariantes a las rotaciones.

Para cada punto de la imagen $L(x, y, \sigma)$, es posible determinar el módulo de su gradiente $m(x, y, \sigma)$ y la fase del mismo $\theta(x, y)$ utilizando diferencias entre pixeles:

$$m(x, y) = \sqrt{(\Delta L_x)^2 + (\Delta L_y)^2}$$

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2}$$

$$\theta(x, y) = \tan^{-1} \left(\frac{\Delta L_y}{\Delta L_x} \right)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)$$

Figura 6. Fuente (Lowe, 2004)

Para determinar de una forma fiel la orientación de cada punto clave, ésta es determinada tomando en cuenta las direcciones de todos los puntos de la imagen dentro de cierto entorno al mismo. Se genera entonces un histograma de direcciones con valores que varían de a 10 grados, ponderado por una ventana Gaussiana circular centrada en el punto clave, de desviación estándar igual a 1.5 el valor del nivel del en cuestión. Cada máximo en el histograma corresponde a la dirección dominante en el gradiente local y será

la asignada al punto clave. Si existen en el histograma otros máximos secundarios de valor mayor o igual al 80% del máximo principal, estos serán utilizados para generar nuevos puntos clave con esa dirección. Sólo al 15% de los puntos clave se les asigna más de una dirección.

Descriptor de puntos clave

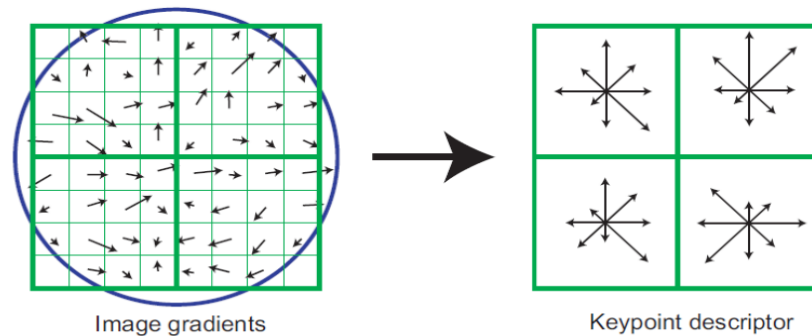


Figura 7. A la izquierda: la ventana Gaussiana pondera los valores de módulo y fase en la vecindad de los puntos de interés. A la derecha: los histogramas con 8 direcciones posibles realizados para cada subregión. (Lowe, 2004)

Hasta el momento, se le ha asignado a cada punto clave una escala, una locación y una orientación. El siguiente paso es determinar para cada punto clave un descriptor relativamente invariante a cambios de iluminación y afinidades, basados en el entorno del mismo.

Una vez determinadas la magnitud y fase del gradiente entorno a un punto clave, una ventana Gaussiana centrada en este pondera los valores de módulo y fase de 4×4 subregiones en la vecindad del mismo y nuevamente genera para cada subregión un histograma de 8 direcciones distintas. Se obtiene finalmente para cada punto clave un descriptor de $4 \times 4 \times 8 = 128$ valores.

En la Imagen 8 se ve cómo se computan los descriptores para cada punto clave. En el ejemplo se utilizan únicamente $2 \times 2 = 4$ subregiones en vez de $4 \times 4 = 16$.

Matching (cálculo de correspondencias)

Al ya tener un descriptor, que es un conjunto de elementos que tienen las principales orientaciones de un keypoint, se deberá determinar si en dos imágenes existen correspondencias, es decir similitud, para lo cual se usa la diferencia euclídea.

$$d_{if_i} = \sqrt{(a_i - b_i)^2}$$

$$n = \# \text{ histogramas} * \# \text{ rotaciones}$$

$$diftotal = \sum_{i=1}^n d_{if_i}$$

```

for all octaves
{
    List keypoint_list;
    for all scales
    {
        ConvolveImageGaussianParallel( );
        BuildDoGParallel( );
        //Detect Keypoint
        #pragma omp parallel for
        for all pixels p in Image
        {
            if( IsKeypoint(p) )
                #pragma omp critical
                keypoint_list.add(p);
        }
    }
    #pragma omp parallel for
    for all pixels kp in keypoint_list
    {
        ExtractFeature(kp);
    }
    DownSampleImageParallel( );
}
    
```

Figura 8. El pseudo código para implementar el algoritmo SIFT paralelo (Chen, 2013)



Figura 9. Proceso del algoritmo SIFT. (Zanchettin & Cenedese, 2012-2013)

2.3.6. Algoritmo RANSAC (Random Sample Consensus)

Concepto

(Rahul Raguram, Jan-Michael Frahm, & Marc Pollefeys, 2008) Es un algoritmo simple, pero potente, técnica que se aplica comúnmente a la tarea de estimar los parámetros de un modelo, utilizando los datos que puedan estar contaminados por los valores atípicos. RANSAC estima una relación global que se ajuste a los datos, mientras que al mismo tiempo la clasificación de los datos (puntos coherentes con la relación) y los valores extremos (puntos no es consistente con la relación).

Debido a su capacidad para tolerar una gran fracción de los valores atípicos, el algoritmo es una opción popular para una variedad de problemas de estimación robusta.

RANSAC opera en un “hypothesize and verify framework” marco hipótesis y verificar: un subconjunto mínimo de los puntos de datos de entrada se selecciona al azar y los parámetros del modelo se estimó a partir de este subconjunto.

Este bucle “hypothesize and verify (hipótesis-y-verificar)” se repite hasta que la probabilidad de encontrar un modelo con un mejor soporte de la mejor modelo actual cae por debajo de un umbral predefinido (típicamente 1% -5%). RANSAC a menudo puede encontrar la solución correcta, incluso para los altos niveles de contaminación; Sin embargo, el número de

muestras necesarias para hacerlo aumenta de manera exponencial, y el coste computacional asociado es sustancial.

Ha habido una serie de recientes esfuerzos encaminados a aumentar la eficiencia del algoritmo básico RANSAC. Algunas de estas estrategias por objeto optimizar el proceso de verificación del modelo, mientras que otros tratan de modificar el proceso de toma de muestras con el fin de generar preferentemente hipótesis más útiles. Si bien estos esfuerzos han demostrado una gran promesa, ninguno de ellos es directamente aplicable en situaciones en las que el rendimiento en tiempo real es esencial. Relativamente menos esfuerzos se han dirigido hacia el objetivo de la formulación de RANSAC de una manera que es adecuado para las implementaciones en tiempo real. En particular, se describe el marco RANSAC preventiva, donde se evaluó un número fijo de hipótesis en un entorno paralelo, de múltiples etapas. En este caso, el objetivo es encontrar, dentro de un presupuesto de tiempo fijo, la mejor solución a partir de un conjunto restringido de hipótesis. Si bien el marco RANSAC preventiva facilita la implementación en tiempo real, existen algunas limitaciones en el esquema. Una de las limitaciones primarias de preventivo RANSAC es su inherente capacidad de adaptación no a los datos. La selección de un número fijo de hipótesis implica implícitamente que una buena estimación previa de la relación está disponible; En la práctica, esto no suele ser el caso. Para los problemas de contaminación bajos, ya que evalúa muchas más hipótesis de lo necesario. Por otra parte, cuando la relación es demasiado baja, RANSAC preventiva es poco probable que encontrar una buena solución, ya que no prueba suficientes hipótesis.

En algunas aplicaciones del algoritmo SIFT pueden surgir algunos problemas debido a una evaluación de las características de los puntos significativos.

Esto significa que en algunos casos se pueden encontrar los resultados inconsistentes con la realidad, y por lo tanto es necesario analizar y procesar los datos de este algoritmo antes de poder usarlo.

Para filtrar estos valores, se solucionará mediante RANSAC.



Figura 10. Escenario problemático implementando SIFT.
(Zanchettin & Cenedese, 2012-2013)

De hecho, en una sola imagen, los puntos que identifican las correspondencias se distribuyen en un lado de la imagen, y luego teniendo en cuenta la posición horizontal de estos puntos se calculó el modelo de tendencia y eliminó las correspondencias en posiciones distantes del resto de los puntos. Imagen 12 Todos los puntos cuyas posiciones difieren mucho de lo general son eliminados como coincidencias encontradas.

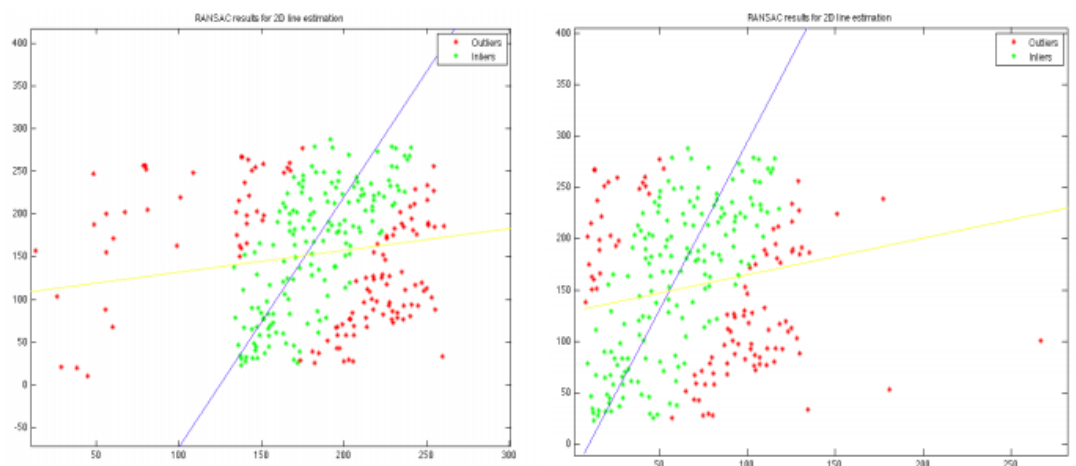


Figura 11. Análisis del Algoritmo RANSAC.
(Zanchettin & Cenedese, 2012-2013)

Seudocódigo:

```
points_fig1 = [ frames1(1,matches(1,:)) ; frames1(2,
    matches(1,:))];
points_fig2 = [ frames2(1,matches(2,:)) ; frames2
    (2,matches(2,:))];
ransac_dataInput(points_fig1,0.95,0.0001,25,struct('
    history',0,'compare',1));
ransac_dataInput(points_fig2,0.95,0.0001,25,struct('
    history',0,'compare',1));
```

Figura 12. Pseudocódigo Algoritmo RANSAC
(Zanchettin & Cenedese, 2012-2013)

Donde se aprecia pares de puntos que están en correspondencia con las bandas, pero no constituyen correspondencias reales asociados como dos puntos en el área de la derecha.

Se calcula la regresión lineal tomando puntos clave seleccionados por primera proyección, (match1, match2), que se representa mediante una fórmula que combina todas las parejas que no son coincidencias correctas.

```
match_temp = (intersect(match1',match2','rows'))';
X_fig = frames1(1,match_temp(1,:)) - frames2(1,
    match_temp(2,:));
Y_fig = frames1(2,match_temp(1,:)) - frames2(2,
    match_temp(2,:));
match_fig = [ X_fig ; Y_fig ];
ransac_dataInput(match_fig,0.95,0.0001,1,struct('
    history',0,'compare',1));
```

Figura 13. Pseudocódigo Algoritmo RANSAC.
(Zanchettin & Cenedese, 2012-2013)

La finalidad de este “mach_fig” es eliminar correctamente todos los datos anormales, dejando sólo los pares de datos correctos, que identifican las correspondencias entre las imágenes. Incluso las parejas que eran obviamente correctos, pero eso es parte del algoritmo que calculan las regiones de la tendencia general.

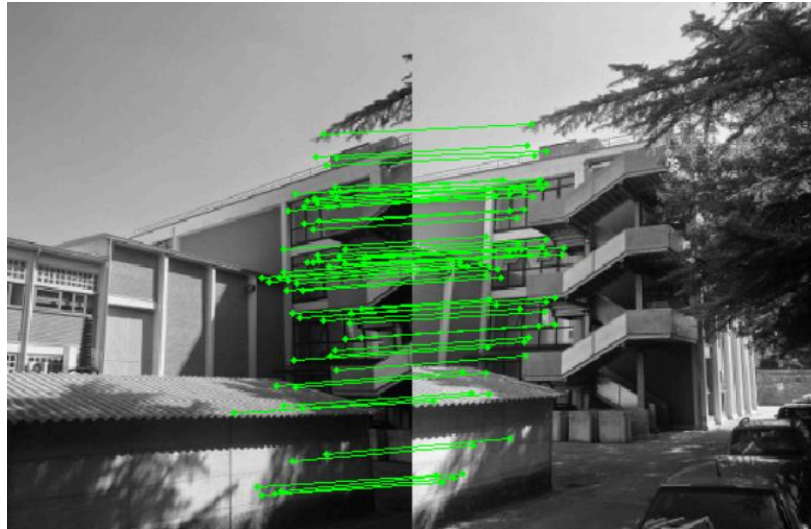


Figura 14. Después de realizar el procesamiento de RANSAC.
(Zanchettin & Cenedese, 2012-2013)

2.3.7. Algoritmo de Levenberg-Marquardt

Este algoritmo es una técnica iterativa que localiza el mínimo de una función multivariable que se expresa como la suma de los cuadrados de las funciones reales no lineales. Se ha convertido en una técnica estándar para problemas no lineales de mínimos cuadrados, ampliamente adoptado en un amplio espectro de disciplinas. LM se puede considerar como una combinación de descenso más agudo y el método de Gauss-Newton. Cuando la solución actual está lejos de ser el correcto, el algoritmo se comporta como un método del descenso más agudo: lento, pero garantiza a converger. Cuando la solución actual está cerca de la solución correcta, se convierte en un método de Gauss-Newton. A continuación, una breve descripción del algoritmo de LM basado en el material en se suministra.

El método de ajuste de curva de Levenberg-Marquardt es en realidad una combinación de dos métodos:

- El método de minimización de gradiente descendente: la suma de los errores cuadráticos se reduce mediante la actualización de los parámetros en la dirección más empinado-descenso.

$$\frac{\partial}{\partial p} x^2 = 2(y - \hat{y}(p))^T w \frac{\partial}{\partial p} (y - \hat{y}(p))$$

$$\begin{aligned} 2(y - \hat{y}(p))^T w \left[\frac{\partial \hat{y}(p)}{\partial p} \right] \\ = -2(y - \hat{y})^T WJ \end{aligned}$$

Donde la matriz $[\partial y / \partial p]$ representa la sensibilidad local de la función y a la variación en los parámetros p . Por simplicidad de notación J será utilizado para $[\partial y / \partial p]$.

El “h” es la actualización de parámetros que mueve en la dirección del descenso más agudo se da por:

$$h_{gd} = \alpha J^T W (y - \hat{y})$$

- El método de Gauss-Newton: la suma de los errores cuadráticos se reduce al asumir la función de mínimos cuadrados es localmente cuadrática, y encontrar el mínimo de la cuadrática

$$\hat{y}(p + h) \approx \hat{y}(p) + \left[\frac{\partial \hat{y}}{\partial p} \right] h = \hat{y} + Jh$$

El método de Levenberg-Marquardt actúa más como un método “gradient-descent” (descenso de gradiente) cuando los parámetros están lejos de su valor óptimo, y actúa más con el método de Gauss-Newton cuando los parámetros están cerca de su valor óptimo.

2.3.8. Herramientas

MATLAB

(Cuevas, Zaldivar, & Perez, 2010) Es un programa que permite realizar calculos tecnicos y cientificos. MatLAB con ya varios años en el mercado y una gran cantidad de funciones implementadas para diferentes disciplinas científicas se ha convertido practicamente en un estandar de programacion y desarrollo rapido de aplicaciones.

La unidad de procesamiento de MatLAB es la matriz, por lo que su utilizacion en procesamiento de imagenes (matrices) es una extension natural de su lenguaje de programacion.

TIPOS DE IMAGENES EN MATLAB

El toolbox de procesamiento de imágenes de MatLAB permite la manipulación de 4 tipos de imágenes:

- a) Imágenes de intensidad o escala de grises
- b) Imágenes binarias
- c) Imágenes indexadas
- d) Imágenes de color RGB

En esta sección solo se referencia el caso de las primeras por ser consideradas las más comunes y básicas en el procesamiento

IMAGENES DE INTENSIDAD O ESCALA DE GRISES

(Cuevas, Zaldivar, & Perez, 2010) Una imagen a escala de grises es una matriz cuyos valores han sido escalados para representar un determinado número de intervalos. Si la imagen es de tipo uint8 entonces los datos que la conforman se encuentran en el intervalo de [0,255]. Si la imagen es del tipo double, entonces los datos que la constituyen son de tipo flotante y se encuentran en el intervalo [0,1].

IMAGENES BINARIAS

Una imagen binaria es de tipo lógico se representa en MatLAB como arreglo que solo contiene unos y ceros. Estos ceros y unos son especiales, por que no implican valores numéricos, sino más bien banderas que indican el estado de falso (0) y verdadero (1).

PROGRAMACION EN MATLAB

(Cuevas, Zaldivar, & Perez, 2010) MatLAB no solamente puede ser usado como una potente calculadora en línea de comandos, sino que es posible realizar programas utilizando estructura y lineamientos propios de un lenguaje de programación. La ventaja de utilizar MatLAB para realizar programas creados para el procesamiento de imágenes o bien de otras funciones realizadas como herramientas para otras áreas afines como el procesamiento digital de señales, además puede de igual manera utilizarse

los poderosos modos de indexado de matrices.

A todas estas ventajas se añade que la forma de programar en MatLAB es prácticamente intuitiva y sencilla, ya que no es necesario declarar variables, ni preocuparse por el tipo de su contenido.

En la programación de MatLAB se pueden crear principalmente dos tipos de programas los llamados .m y las funciones. Este tipo de programas debe ser considerado como un script, el cual ejecuta una serie de instrucciones en el MatLAB.

LENGUAJE DE PROGRAMACION C++

(Zahonero, 2014) Existen muchas formas de analizar el código c++. Una es desde el punto de vista del programador de aplicaciones que se centra en los tipos y clases necesarias para crear un programa utilizando funciones y estructuras de control. Existe otra visión que es la realizada por el programador de sistemas que se centra en obtener los nuevos tipos y clases, que son, a su vez, abstracciones de conceptos.

En la práctica real y profesional, los programadores en el siglo xxi, normalmente adoptan ambos enfoques, debido a que el proceso de abstracción del problema conduce, normalmente, al uso frecuente y repetido de clases y tipos en los programas, teniendo presente las clases a nivel más alto, las relaciones entre ellas, etc.

Y nos describe también las herramientas básicas que utiliza el programador de las aplicaciones utilizando lenguaje de programación c++.

2.4. Definición de los términos básicos

- a) Adaptativa. Que está relacionado con la capacidad de adaptación.
- b) Adyacente. Es un adjetivo utilizado para nombrar a lo que se ubica en las cercanías de algún objeto o cosa.
- c) Anisótropo. Es la propiedad general de la materia según la cual cualidades como elasticidad, temperatura, conductividad, velocidad de propagación de la luz, etc. varían según la dirección en que son examinadas. Algo anisótropo podrá presentar diferentes características según la dirección. La anisotropía

de los materiales es más acusada en los sólidos cristalinos, debido a su estructura atómica y molecular regular. De la misma manera que los números pueden "medir" la magnitud escalar de una propiedad, los grupos matemáticos pueden "medir" la simetría de una propiedad. Así el tipo de anisotropía que presenta una propiedad puede caracterizarse mediante un subgrupo de un grupo de simetría de rotaciones.

- d) Automatización. Es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.
- e) Computacional. De la tecnología informática o relacionado con ella, o que utiliza los métodos o recursos de la informática.
- f) Distorsión. Deformación de un sonido, una imagen, una señal, etc., producida durante su transmisión o reproducción.
- g) Esquina. Es el lugar donde dos líneas o superficies se encuentran en una imagen.
- h) Estimar. Atribuir valor a una persona o cosa, o reconocer el mérito que tiene.
- i) Fotogrametría. Técnica para obtener mapas y planos de grandes extensiones de terreno por medio de la fotografía aérea.
- j) Imagen. Arreglo bidimensional de píxeles con diferente intensidad luminosa (escala de gris).
- k) Imagen panorámica. Es la que muestra un panorama, usualmente paisajístico o arquitectónico, y que se distingue por el amplio horizonte visual que cubre.
- l) Indexación. Ordenación de una serie de datos o informaciones de acuerdo con un criterio común a todos ellos, para facilitar su consulta y análisis.
- m) Matriz. Es un arreglo bidimensional de números.
- n) Método. Se refiere al medio utilizado para llegar a un fin. Su significado original señala el camino que conduce a un lugar.
- o) Modelo matemático. Es uno de los tipos de modelos científicos que emplea algún tipo de formulismo matemático para expresar relaciones, proposiciones sustantivas de hechos, variables, parámetros, entidades y relaciones entre variables.

- p) **Parámetro.** Es una variable que forma parte de los lenguajes de programación, metodos o algoritmos.
- q) **Procesamiento de Imágenes.** Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.
- r) **Proceso Automatizado.** Serie de pasos para desarrollar un trabajo, producto o cosa sin necesidad de la intervencion del hombre.
- s) **Registro de la imagen.** Es el proceso de transformación de diferentes conjuntos de datos a un sistema de coordenadas.
- t) **Restricción.** Limitación que se produce en alguna cosa, especialmente en el consumo de algo.
- u) **Robustez.** El primer paso que tenemos que dar, antes de proceder a analizar algo a fondo.
- v) **Superponer.** Poner una cosa encima de otra.
- w) **Superposición.** Acción de superponer.
- x) **Sutura de imagenes.** Coser una imagen o cerrarla mediante puntos de sutura.
- y) **Tramas.** Son un conjunto de puntos que al distribuirse más o menos separados, crean diferentes intensidades de color, cuanto más juntos están, el color es más oscuro y cuanto más separados más claro es el color.
- z) **Transformación.** Conjunto de transiciones que define la sucesion de los estados.
- aa) **Transición.** Estado intermedio entre uno más antiguo y otro a que se llega en un cambio.

3. CAPÍTULO III: MARCO METODOLÓGICO

3.1. Tipo de investigación

3.1.1. Tipo de estudio

Aplicada

Tiene por finalidad la búsqueda y consolidación del saber y la aplicación de los conocimientos para el enriquecimiento del acervo cultural y científico, así como la producción de tecnología al servicio del desarrollo integral de las naciones. La investigación aplicada puede ser Fundamental o Tecnológica.

La aplicada fundamental, se entiende como aquella investigación relacionada con la generación de conocimientos en forma de teoría o métodos que se estima que en un período mediano podrían desembocar en aplicaciones al sector productivo.

3.1.2. Diseño de estudio

Cuasi-experimental: Existen dos grupos de muestras manipulables que no están bajo un control estricto en tiempo y espacio, aplicando los algoritmos de stitching para verificar los efectos que pueda producir con la finalidad de resolver un problema el cual es reducir los errores de distorsión.

3.2. Población y Muestra

3.2.1. Población

La población para la investigación son los Algoritmos de Stitching.

3.2.2. Muestra

La muestra para la investigación son los 4 algoritmos de Stitching

3.3. Hipótesis

Si se aplican algoritmos SIFT, RANSAC y Levenberg-Marquardt se logrará un stitching de imágenes digitales de manera eficiente y con reducción de errores.

3.4. Variables

3.4.1. Variables independientes:

Algoritmos de SIFT, RANSAC y Levenberg-Marquardt

3.4.2. Variable dependiente:

Stitching eficiente de imágenes digitales.

3.5. Operacionalización

Variable	Dimensiones	Indicadores	Formulas	Técnicas o instrumentos de recolección de datos
Stitching de imágenes digitales	Performance	Error de distorsión	$distortion\ degree = Max\{Slope(i, j)\},$ $i, j \in centerSet.$	Las técnicas de recolección de la información será la observación directa del experimento haciendo uso de herramientas como fichas de observación y bitácoras. Las técnicas de recolección de la información será la observación directa del experimento haciendo uso de herramientas como fichas de observación y bitácoras.
		Tiempo de Stitching por escena	Se mide por tiempo de ciclo del reloj	
		Número de puntos "Matching"	E $= MAX(\sum_{i=1}^{k-1} \sum_{m \in \omega(i, i+1)} d_m, \sum_{i=k+1}^n \sum_{m \in \omega(i, i-1)} d_m)$	
		% de área de información útil	$info\ proportion = 1 - \frac{black\ pixels}{width \times height}$	

3.6. Abordaje metodológico, técnicas e instrumentos de recolección de datos

Ficha de observación:

La ficha de observación es considerada como una especie de procedimiento de investigación, el cual consiste básicamente en poder utilizar instrumentos adecuados para poder establecer una relación entre la hipótesis y los hechos reales, a través de la observación científica, también de la investigación sistematizada y ordenada.

Los instrumentos que utiliza la ficha de observación para poder registrar la descripción detallada de las cosas observadas e investigadas, además se considera también que este instrumento hace posible la recolección de datos, basado en un objetivo específico, en el cual se determinan variables específicas.

Cabe mencionar que la ficha de observación es aquel documento mediante el que es posible también tener toda la información posible de algún tema en particular, puede ser la información sobre alguien o sobre algo, esta obtención de datos son el resultado de la observación. Se considera que una ficha de observación puede durar gran o corta cantidad de tiempo.

Generalmente las características que posee una ficha de observación se llegan a determinar a través de la observación del área, el desempeño, el tiempo, las variables.

Bitácoras:

La bitácora es un cuaderno en el que se reportan los avances y resultados preliminares de un proyecto de investigación. En él, se incluyen con detalle, entre otras cosas, las observaciones, ideas, datos, de las acciones que se llevan a cabo para el desarrollo de un experimento o un trabajo de campo. Se puede ver como un instrumento cuya aplicación sigue un orden cronológico de acuerdo al avance del proyecto.

Según (Alva, 2011), “la bitácora es el diario de trabajo” y su elaboración es un paso imprescindible en el transcurso de un proyecto de investigación. Gracias a ello, los experimentos que se realicen posteriormente, pueden ser repetidos en el momento que se desee, obteniéndose los mismos resultados: la bitácora debe

guardar las condiciones exactas bajo las cuales se ha trabajado. Además, en ella se pueden escribir ideas e hipótesis derivadas del desarrollo empírico. Es importante mencionar que la bitácora es un texto que constantemente se está modificando; sin embargo, nunca se le deben arrancar hojas ni borrar información; si se comete algún error, se debe poner una línea en diagonal para indicarlo, de tal forma que el texto se siga apreciando, puesto que cualquier detalle, incluso un error, puede llegar a ser utilizado posteriormente. Función en el medio académico La bitácora de investigación es un género muy utilizado en el área científica porque permite documentar, de forma precisa, los diversos experimentos que se realizan, así como sus resultados. Además, este género escrito sirve para comunicar a otros miembros de la comunidad científica los detalles del proceso, para que éstos juzguen si se procedió de acuerdo a lo planeado, qué modificaciones serían pertinentes y qué criterios se utilizaron. Finalmente, la bitácora de investigación sirve como guía para futuros experimentos o modificaciones que el estudiante o investigador quiera realizar. Es un documento que guarda los pasos aprendidos.

3.7. Procedimiento para la recolección de datos

Esta parte se realizará en 3 pasos básicos:

a) Adquirir Imágenes

Se utilizara una bitacora para el registro de las caracteriztica de la imágenes.

b) Pre procesamiento de las Imágenes

Aquí emplearemos el uso de una ficha de observacion y bitacora para el registro de los cambios de las dimensiones de las imágenes.

c) Procesamiento de las imágenes

Se requerira una ficha de observacion para registrar el comportamiento de los algoritmos em las distintas fases del stitching de imágenes.

3.8. Análisis estadístico e interpretación de los datos

Se realizará haciendo uso de técnicas estadísticas para determinar el rendimiento de los algoritmos según los indicadores propuestos se usará un software estadístico especializado para el procesamiento de información.

Error de distorsión

La cantidad de área deformada que hay en la imagen panorámica producida durante el proceso de Stitching.

$$distortion\ degree = Max\{Slope(i, j)\}, i, j \in centerSet.$$

Número de puntos “Matching”

Son los puntos de referencia de la imagen adyacente y la imagen secuencial, por medio del cual se realiza el Stitching de imagen.

$$E = MAX(\sum_{i=1}^{k-1} \sum_{m \in \omega(i, i+1)} d_m, \sum_{i=k+1}^n \sum_{m \in \omega(i, i-1)} d_m)$$

Porcentaje de área de información útil (info porportion)

Al contrario del error de distorsión, estos son todas las partes visibles en la imagen panorámica.

$$info\ proportion = 1 - \frac{black\ pixels}{width \times height}$$

Tiempo de stitching por escena

Este dato es hallado por el tiempo de ciclos del reloj.

3.9. Principios éticos

El consentimiento informado: Un aspecto para tener en cuenta es cuánta información debe darse a los informantes, ya que existe el riesgo de que se les predisponga de manera positiva o negativa para los objetivos de la investigación. La responsabilidad moral del investigador será la que determine la implicación metodológica en este punto. Es importante reconocer que la información suministrada se convierte en un elemento que influye en la autenticidad del estudio debido a que, al tener el informante

conocimiento de la intencionalidad de la recolección de los datos, puede dar respuestas socialmente aceptables.

La confidencialidad: Los códigos de ética hacen énfasis en la seguridad y protección de la identidad de las personas que participan como informantes de la investigación. La confidencialidad se refiere tanto al anonimato en la identidad de las personas participantes en el estudio, como a la privacidad de la información que es revelada por los mismos, por tanto, para mantenerla se asigna un número o un pseudónimo a los entrevistados.

Manejo de riesgos: Dos aspectos se deben considerar para minimizar los riesgos a los participantes en la investigación. El primero se refiere a que el investigador deberá cumplir con cada una de las responsabilidades y obligaciones adquiridas con los informantes; el segundo está relacionado con el manejo posterior de los datos proporcionados. Una investigación cualitativa se justifica cuando cumple con los siguientes requisitos: “los riesgos potenciales para los sujetos individuales se minimizan; los beneficios potenciales para los sujetos individuales y para la sociedad se maximizan; los beneficios potenciales son proporcionales o exceden a los riesgos”.

3.10. Criterios de rigor científico

Consistencia

Los datos recolectados para este trabajo son de carácter científico y formal. El análisis realizado a los datos está hecho con total profesionalidad aplicando habilidades, técnicas y conocimientos de la ingeniería y de la investigación para mantener la consistencia de los datos y resulte en información consistente y útil.

Validez

Los datos obtenidos de la red ad hoc luego de su implementación será correctamente evaluadas y analizados para lograr dar un resultado válido que verifique la hipótesis.

Fiabilidad

Las técnicas e instrumentos de medición utilizados para obtener los valores de los

parámetros medibles de la red ad hoc inalámbrica, siempre dan como resultado valores aproximadamente iguales en las mismas circunstancias y situaciones.

Objetividad

Los datos que se obtendrán de la experimentación con los protocolos en la red ad hoc, así como los resultados del análisis de dichos datos están exentos de la influencia de la perspectiva de los investigadores.

4. CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

4.1. Resultados en tablas y gráficos

Número de puntos “Matching”

$$E = MAX \left(\sum_{i=1}^{k-1} \sum_{m \in \omega(i, i+1)} d_m, \sum_{i=k+1}^n \sum_{m \in \omega(i, i-1)} d_m \right)$$

Tabla 2

Número de puntos “Matching” en el “PRIMER CONJUNTO DE IMAGENES”

IDENTIFICADOR DE FIGURA	TECNICAS EMPLEADAS	
	SURF	SIFT
Figura 22	564	574
Figura 23	525	454
Figura 24	568	602

Fuente: (Autoría Propia)

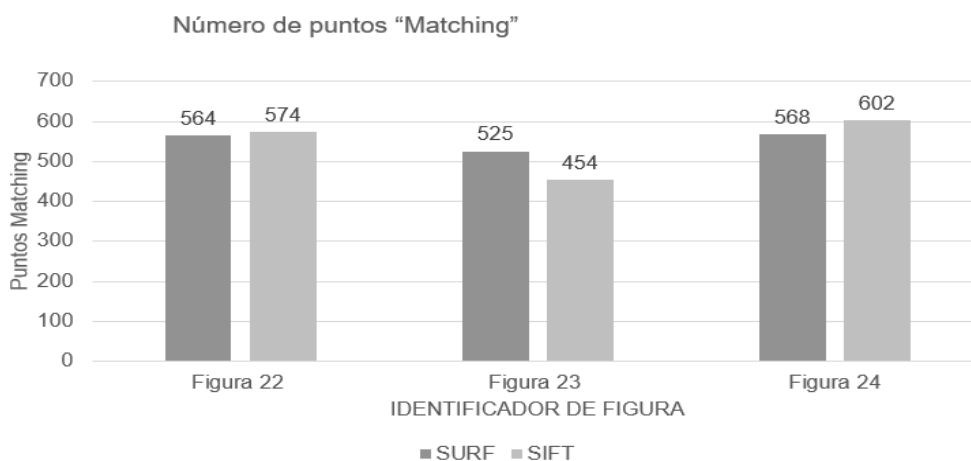


Figura 15. Es una imagen donde se grafica la Tabla 02. Fuente: Elaboración Propia

Tabla 3

Muestra medida por Número de puntos “Matching” en el “SEGUNDO CONJUNTO DE IMAGENES”

IDENTIFICADOR DE FIGURA	TECNICAS EMPLEADAS	
	SURF	SIFT
Figura 25	378	325
Figura 26	294	253
Figura 27	416	331
Figura 28	297	271

Fuente: (Autoría Propia)

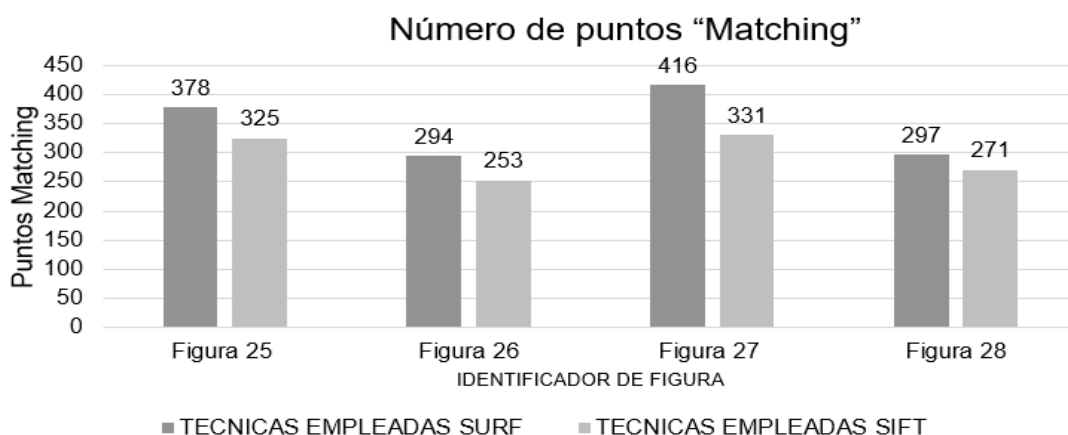


Figura 16. . Es una imagen donde se grafica la Tabla 03. Fuente: Elaboración Propia

Tabla 4

Muestra medida por Número de puntos “Matching” en el “PRIMER CONJUNTO DE IMAGENES”

IDENTIFICADOR DE FIGURA	TECNICAS EMPLEADAS	
	SURF-RANSAC	SIFT-RANSAC
Figura A - B	564	578
Figura B - C	525	454

Fuente: (Autoría Propia)

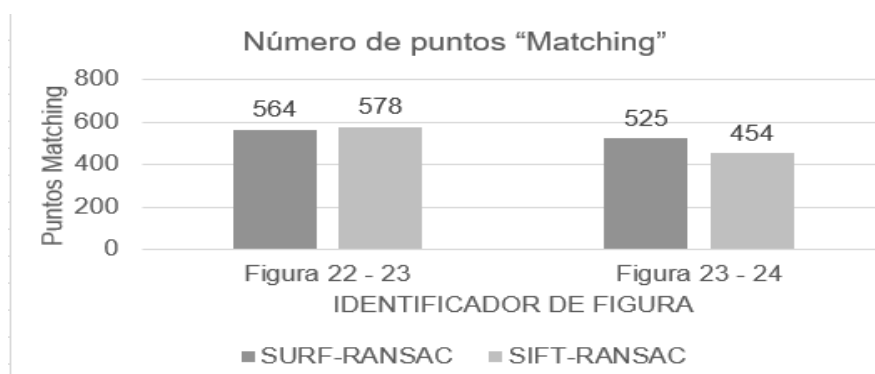


Figura 17. Es una imagen donde se grafica la Tabla 04. Fuente: Elaboración Propia

Tabla 5

Muestra medida por Número de puntos "Matching" en el "SEGUNDO CONJUNTO DE IMAGENES"

IDENTIFICADOR DE FIGURA	TECNICAS EMPLEADAS	
	SURF-RANSAC	SIFT-RANSAC
Figura 25 – 26	378	325
Figura 26 – 27	294	253
Figura 27 - 28	416	331

Fuente: (Autoría Propia)

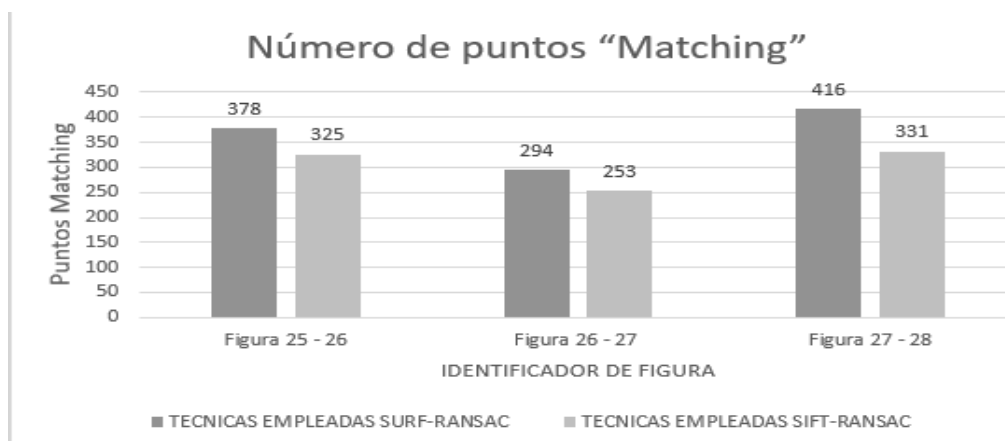


Figura 18. Es una imagen donde se grafica la Tabla 05. Fuente: Elaboración Propia

Porcentaje de área de información útil (info porportion)

Black pixels = Son los pixeles negros que contiene la imagen panorámica.

Width = Es ancho de la imagen panorámica.

Height = Es la altura de la imagen panorámica.

RESULTADO DEL STITCHING DEL PRIMER CONJUNTO DE IMAGENES

$$info\ proportion = 1 - \frac{black\ pixels}{width\ x\ height}$$

$$info\ proportion = 1 - \frac{21257}{278x\ 663}$$

$$info\ proportion = 0.8847\ \% \ area\ util$$

RESULTADO DEL STITCHING DEL SEGUNDO CONJUNTO DE IMAGENES

$$info\ proportion = 1 - \frac{black\ pixels}{width\ x\ height}$$

$$info\ proportion = 1 - \frac{24585}{281x\ 1023}$$

$$info\ proportion = 0.9145\ \% \ area\ util$$

Tiempo de stitching por escena

Se medira por tiempo de ejecucion del proceso en segundos.

Tabla 6

Muestra medida en Tiempo de Stitching

TECNICAS		
EMPLEADAS		
IDENTIFICADOR	SURF	SIFT
DE FIGURA		
Figura 25 - 26	0.65 sec	3.00 sec
Figura 26 - 27	0.51 sec	2.53 sec
Figura 27 - 28	0.52 sec	2.39 sec

Fuente: (Autoría Propia)

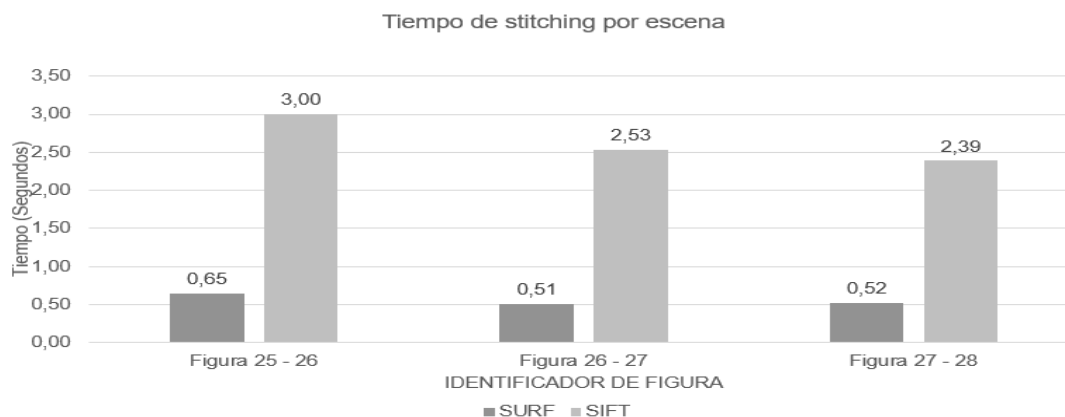


Figura 19. Es una imagen donde se grafica la Tabla 06. Fuente: Elaboración Propia

Tabla 7
Muestra medida en Tiempo de Stitching

TECNICAS EMPLEADAS		
IDENTIFICADOR DE FIGURA	SURF-RANSAC	SIFT-RANSAC
Figura 25 - 26	1.32433 sec	3.08583 sec
Figura 26 - 27	1.41584 sec	3.08887 sec
Figura 27 - 28	1.56422 sec	3.12688 sec

Fuente: (Autoría Propia)

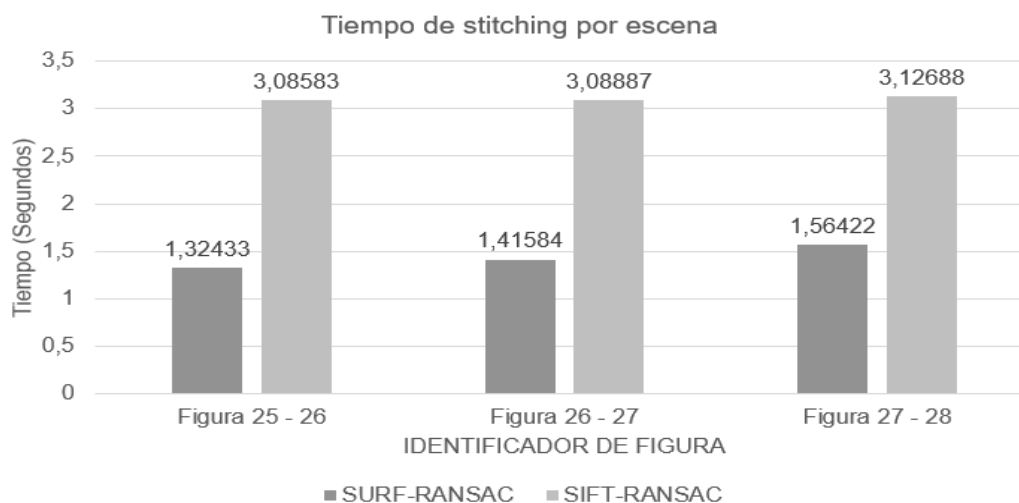


Figura 20. Es una imagen donde se grafica la Tabla 07. Fuente: Elaboración Propia

Tabla 8

Muestra medida en Tiempo de Stitching del Primer conjunto de Imágenes

TECNICAS EMPLEADAS

IDENTIFICADOR DE FIGURA	STITCHING
Figura 22 - 23 - 24	7.19 sec
Figura 22 - 23 - 24	18.83 sec

Fuente: (Autoría Propia)

Tabla 9

Muestra medida en Tiempo de Stitching del Segundo conjunto de imágenes

TECNICAS EMPLEADAS

IDENTIFICADOR DE FIGURA	STITCHING
Figura 25 – 26 – 27 - 28	8.85 sec
Figura 25 – 26 – 27 - 28	17.47 sec

Fuente: (Autoría Propia)

Realizado la técnica de Stitching con Visual Studio y OpenCV ejecutados en dos ordenadores con distintas características.

4.2. Discusión de resultados

La decisión de realizar Stitching de imágenes digitales, utilizando las técnicas anteriormente explicadas queda a criterio del autor de la investigación usada para replicar los algoritmos.

El porcentaje de efectividad según la investigación, varía a los resultados obtenidos por el autor, dado que las imágenes usadas no son las mismas que emplearon los autores originales de la investigación usada para replicar las técnicas.

Para la programación de las técnicas de Stitching de imágenes se usó el lenguaje C++ junto a la librería OpenCV 2.4.13 y el entorno de desarrollo se realizó en Visual Studio 2013. Al extraer los puntos “Matching” de cada imagen, se pudo observar una gran variación en los valores calculados matemáticamente, se concluyó que la razón era que aplicando el algoritmo Surf obtuvimos más puntos Matching como se explica en los gráficos anteriores.

Los descriptores de la investigación estaban siendo calculados manualmente,

utilizando una fórmula matemática y métodos propios del OpenCV contrariamente a la investigación de los autores originales aplican su lógica de programación.

Extraído de la investigación Titulada “The Improved Algorithm of Fast Panorama Stitching for Image Sequence and Reducing the Distortion Errors “. (Zhong Qu, Si-Peng Lin, Fang-Rong Ju, & Ling Liu, 2015)

Tecnica Utilizadas:

- Algoritmo Sift para realizar la conversión de imágenes a puntos
- Formula su programación mediante matrices para la unión de puntos.
- Optimiza la matriz mediante el algoritmo L-M.
- Como resultado obtienen lo siguiente

PRIMERA DISCUSION DE RESULTADOS

Tabla 10

The comparison data of distortion degree between the traditional method and the improved method

Scene	info proportion	
	The traditional method	The improved method
Figure 7(a)	81.4%	93.6%
Figure 8(a)	82.8%	94.7%
Figure 9(a)	85.4%	95.0%

Fuente: (Zhong Qu, Si-Peng Lin, Fang-Rong Ju, & Ling Liu, 2015)

Donde el porcentaje de área de información útil (info porportion) de imagen panorámica obtenido por nuestra investigación llega a un 83,20 % con se demuestra anteriormente y comparado con la investigación de referencia que llega a un porcentaje de 88,47 % en el primer conjunto de imágenes y 91, 45 % en el segundo conjunto de imágenes correctamente.

Nos lleva por un 10 % de diferencia.

SEGUNDA DISCUSION DE RESULTADOS

En este caso se evaluo por el Tiempo de Stitching de la imagen panoramica. Tomando como referencia la siguiente imagen donde nos muestra el tiempo de

duración del Stitching de las Imágenes y la cantidad de líneas trazadas por los puntos Ransac.

En esta parte solo nos referiremos al tiempo de stitching, lo demás se examinará más adelante.

Tabla 11
Experimental data in image registration process

Stitching processing	The inliers number after RANSAC	The time for stitching (s)
Figures 3(a) and 3(b)	79	10.26
Figures 3(b) and 3(c)	49	9.77
Figures 3(c) and 3(d)	56	11.82

Fuente: (Zhong Qu, Si-Peng Lin, Fang-Rong Ju, & Ling Liu, 2015)

En nuestra investigación realizamos el análisis de Tiempo de Stitching por cada fase del proceso es decir en los intervalos de los algoritmos implementados y al final el proceso completo del Stitching como se está explicado en los gráficos anteriores y teniendo como resultado (7,19 sec y 8,85 sec) dichos resultados se realizaron en distintos ordenadores, y en la investigación de referencia se llega a los resultados de 9,77 a 11,82 segundos.

Estos resultados dependerán mucho del ordenador donde se implementa la investigación y los programas para ejecutar dichas técnicas. Por lo cual existen cantidad de entornos y lenguajes de desarrollo para realizar el Stitching de Imágenes, considerando las diferentes formas de consumo de recursos del ordenador.

Otro punto importante sería la manera de utilizar los beneficios que trae los programas para Stitching de Imágenes para agilizar el tiempo de ejecución con el uso de métodos propios por el programa cada vez que actualiza su software.

TERCERA DISCUSION DE RESULTADOS

Se evaluó considerando los puntos Matching obtenidos en los diferentes procesos del Stitching de imágenes, los cuales son el medio de relación entre las imágenes. En nuestra investigación esto son obtenidos por cada técnica aplicada

para el proceso de Stitching como esta detallado en graficos anteriores.

En la investigación de referencia esto se obtiene mediante el proceso del algoritmo Ransac obteniendo un resultado de 49 a 79 líneas trazadas por los puntos Matching.

En nuestra investigación obtenemos 250 a 325 líneas trazadas por los puntos Matching de algoritmo Ransac, considerando que la técnica del algoritmo Surf y Ransac superan con casi el doble obteniendo de 275 a 400 puntos Matching de algoritmo Ransac.

Esto se debe principalmente a que en nuestra investigación se trabaja con imágenes distintas al de la investigación de referencia y la aplicación de técnicas de conversión de puntos distintas como es el caso del algoritmo surf.

5. CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

5.1. Adquisición imágenes

Estas imágenes son capturadas por un UAV con las siguientes características:

Marca: DJI

Modelo: Phantom 3 Estándar



Figura 21. UAV Utilizado para las capturas de las imágenes digitales.

Fuente:Elaboracion Propia.

(DJI, 2016) La cámara incorporada graba un vídeo de 2.7K, que es casi 3 veces más claro que la imagen en su televisor de alta definición, a 30 cuadros por segundo y un máximo de 40 Mbps. O grabar en HD 1080p completo para capturar cada fotograma y nunca se pierda un momento.

Phantom 3 Standard dispara archivos JPEG de 12 megapíxeles e incluso DNG RAW que le da el poder para crear fotos profesionales. Un sensor de 1 / 2,3 ", una lente rápida f / 2,8 prime y un enfoque predefinido están optimizados para las imágenes aéreas que proporcionan resultados impresionantes.

PRIMER CONJUNTO DE IMÁGENES:



Figura 22. Imagen “A” del primer conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia



Figura 23. Imagen “B” del primer conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.



Figura 24. Imagen “C” del primer conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.

SEGUNDO CONJUNTO DE IMÁGENES:



Figura 25. Imagen “W” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.



Figura 26. Imagen “X” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia



Figura 27. Imagen “Y” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia



Figura 28. Imagen “Z” del segundo conjunto de imágenes tomada por el UAV. Fuente: Elaboración Propia.

5.2. Determinar algoritmos para procesamiento de imágenes

ALGORITMOS	DESCRIPCION	NOTACION MATEMATICA
SURF	Se basa en propiedades similares, con una complejidad descubierta aún más. El primer paso consiste en fijar una orientación reproducible sobre la base de la imagen de una región circular alrededor del punto de interés. A continuación, se construye una región cuadrada alineada con la orientación seleccionada, y se extrae el descriptor SURF de ella. (Figura 29)	$v = (\sum d_x; \sum d_y; \sum d_x ; \sum d_y)$ <p>(Figura 30)</p>
SIFT	Se encarga de extraer características distintivas de las imágenes, las características son invariantes a escala de la imagen y la rotación, y parcialmente invariantes a cambios en la iluminación y el punto de vista de la cámara. Las características están bien localizadas en los dominios espaciales y de frecuencia, la reducción de la probabilidad de interrupción por la oclusión, el desorden, o ruido.	<p>Tiene dos Etapas:</p> <ul style="list-style-type: none"> • Función scale-space: $L(x, y, \sigma)$ $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$ <ul style="list-style-type: none"> • Función Difference-of-Gaussian: $D(x, y, \sigma)$ $D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y)$ $D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$ <p>(Plaza Cordero & Zambrano Martínez, 2012)</p>
Affine-SIFT	Simula un conjunto de puntos que se muestra en las	En este caso la imagen se va a transformar en todas las

(ASIFT)	<p>imágenes iniciales, dicha simulación se realiza en tres parámetros: el zoom, el ángulo de la cámara en latitud y el ángulo de la cámara en longitud. Que no son tratados por el algoritmo SIFT, y normaliza los otros 3 parámetros: la traslación, rotación y escala. Por lo tanto, ASIFT cubre efectivamente los seis parámetros de la transformación Affine. (Figura 31)</p>	<p>distorsiones posibles provocados por la orientación de la cámara (longitud ϕ, la latitud θ). Primero la imagen tendrá una rotación del ángulo (ϕ), luego a una inclinación de:</p> $t = \frac{1}{ \cos \theta }$ <p>Cuando tenemos imágenes digitales, la inclinación es por una t direccional, por lo tanto, se debe usar un filtro que le suavice usando una convolución de la desviación estándar y de la gaussiana. (Plaza Cordero & Zambrano Martínez, 2012) (Figura 32)</p>
Harris Detector (Mikolajczyk & Schmid, 2004)	<p>Se basa en la segunda matriz de momentos. La segunda matriz de momentos, también llamada la matriz de auto correlación, a menudo se utiliza para la detección de características o para la descripción de las estructuras locales de la imagen. Esta matriz debe adaptarse a cambios de escala para que sea independiente de la resolución de imagen.</p>	$u(x, \sigma_1, \sigma_D) = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$ $= \sigma_D^2 g(\sigma_1) * \begin{bmatrix} L_x^2(x, \sigma_D) & L_x L_y(x, \sigma_D) \\ L_x L_y(x, \sigma_D) & L_y^2(x, \sigma_D) \end{bmatrix}$
Harris-Laplace (Mikolajczyk & Schmid, 2004)	<p>Utiliza la función Harris escala adaptada (Harris Detector) para localizar puntos en la escala-espacio. A continuación, selecciona los puntos por los que el laplaciano Gaussianos alcanza un máximo a través de la</p>	<p>El algoritmo consta en: un punto de detección multi-escala y una selección iterativa de la escala y la ubicación.</p> <ol style="list-style-type: none"> 1. Encontrar el extremo local sobre la escala del registro del punto $x(k)$

	<p>escala. Se proponen dos algoritmos.</p> <p>El primero es un algoritmo iterativo que detecta simultáneamente la localización y la escala de regiones características. El segundo es un algoritmo simplificado, que es menos preciso, pero más eficiente.</p>	$\sigma_1^{(k+1)} = t\sigma_1^{(k)}$ <p>with $t \in [0.7, \dots, 1.4]$</p> <p>2. Detectar la ubicación espacial $x(k+1)$ de un máximo de Harris medir más cercana a $x(k)$ para el seleccionado.</p> <p>Vaya al paso 1 si: $\sigma_1^{(k+1)} \neq \sigma_1^{(k)}$ or $x^{(k+1)} \neq x^{(k)}$</p>
Harris-Affine (Mikolajczy & Schmid, 2004)	<p>El enfoque que se muestra cómo el “the Harris-Laplace” se comporta en el caso de transformaciones Affine de la imagen. Cada paso del algoritmo de detección se discute en detalle y se presenta un esquema del procedimiento iterativo.</p>	$\mu(x, \Sigma_1, \Sigma_D) = \det(\Sigma_D)g(\Sigma_1)$ $* ((\nabla L)(x, \Sigma_D)(\nabla L)(x, \Sigma_D)^T)$
MSER (Maximally Stable Extremal Regions) (J. Matas, O. Chum, M. Urban, & T. Pajdla, 2002)	<p>Imagínese todas las posibles umbralización de una imagen en escala de grises I. Los píxeles por debajo de un umbral como 'negro' y para aquellos por encima o igual como "blanco". Si nos mostraron con el cuadro t correspondiente al umbral t, veríamos por primera vez una imagen en blanco. Posteriormente puntos negros corresponden a los mínimos de intensidad locales van a aparecer y crecer.</p>	<p>$Q_1, \dots, Q(i-1), Q_i, \dots$ ser una secuencia de las regiones extrémale anidados, i.e. $Q_i \subset Q(i+1)$.</p> <p>Extremal regions (Q_i) es máximamente estable:</p> $q(i) = Q_{i+\Delta} \setminus Q_{i-\Delta} / Q_i $ <p>Tiene un mínimo local en i^* (\cdot Denota la cardinalidad). $\Delta \in S$ es un parámetro del método. (Imagen 29)</p>

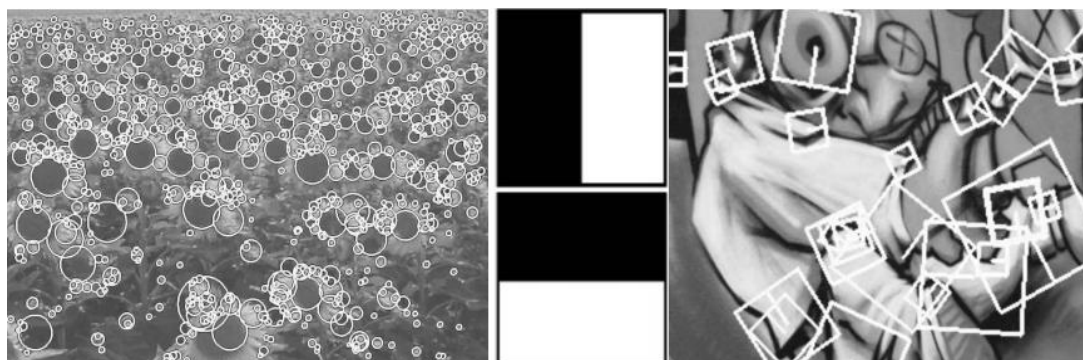


Figura 29. Descriptor SURF puntos de interés. (Bay, Tuytelaars, & Van Gool)



Figura 30. Las entradas de descripción de una subregión representan la naturaleza del patrón de intensidad subyacente. (Bay, Tuytelaars, & Van Gool)

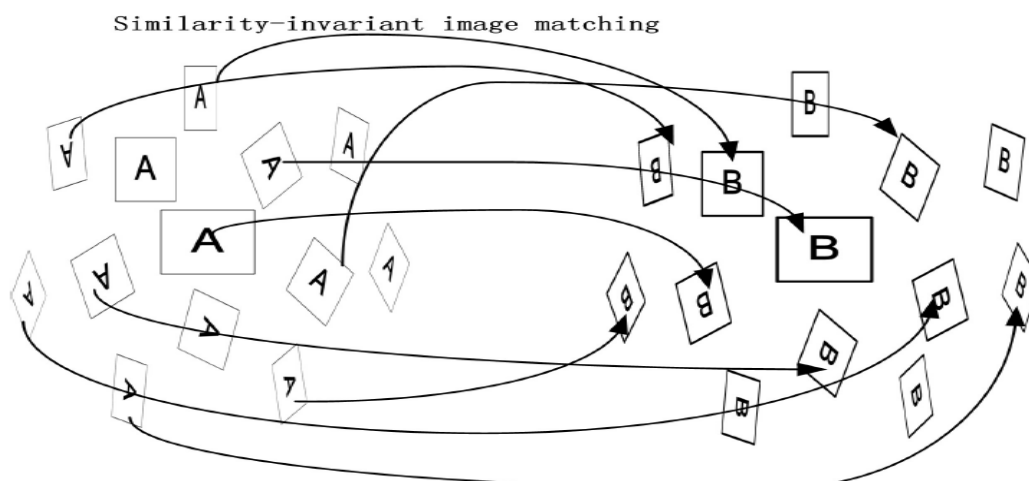
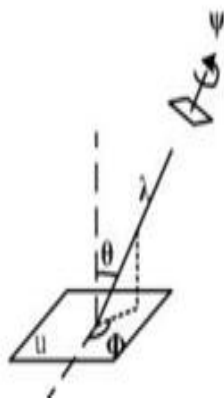


Figura 31. Visión general del Algoritmo ASIFT. (Morel & Yu, 2011)



$$A = \lambda \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}$$

Figura 32. Interpretación de la formula geométrica de ASIFT. (Morel & Yu, 2011)

Image I is a mapping $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$. Extremal regions are well defined on images if:

1. \mathcal{S} is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation \leq exists. In this paper only $\mathcal{S} = \{0, 1, \dots, 255\}$ is considered, but extremal regions can be defined on e.g. real-valued images ($\mathcal{S} = \mathbb{R}$).
2. An adjacency (neighbourhood) relation $A \subset \mathcal{D} \times \mathcal{D}$ is defined. In this paper 4-neighbourhoods are used, i.e. $p, q \in \mathcal{D}$ are adjacent (pAq) iff $\sum_{i=1}^d |p_i - q_i| \leq 1$.

Region \mathcal{Q} is a contiguous subset of \mathcal{D} , i.e. for each $p, q \in \mathcal{Q}$ there is a sequence $p, a_1, a_2, \dots, a_n, q$ and $pAa_1, a_iAa_{i+1}, a_nAq$.

(Outer) Region Boundary $\partial\mathcal{Q} = \{q \in \mathcal{D} \setminus \mathcal{Q} : \exists p \in \mathcal{Q} : pAq\}$, i.e. the boundary $\partial\mathcal{Q}$ of \mathcal{Q} is the set of pixels being adjacent to at least one pixel of \mathcal{Q} but not belonging to \mathcal{Q} .

Extremal Region $\mathcal{Q} \subset \mathcal{D}$ is a region such that for all $p \in \mathcal{Q}, q \in \partial\mathcal{Q} : I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region).

Maximally Stable Extremal Region (MSER). Let $\mathcal{Q}_1, \dots, \mathcal{Q}_{i-1}, \mathcal{Q}_i, \dots$ be a sequence of nested extremal regions, i.e. $\mathcal{Q}_i \subset \mathcal{Q}_{i+1}$. Extremal region \mathcal{Q}_{i^*} is maximally stable iff $q(i) = |\mathcal{Q}_{i+\Delta} \setminus \mathcal{Q}_{i-\Delta}| / |\mathcal{Q}_i|$ has a local minimum at i^* ($|\cdot|$ denotes cardinality). $\Delta \in \mathcal{S}$ is a parameter of the method.

Figura 33. Definiciones utilizadas matemáticamente. (J. Matas, O. Chum, M. Urban, & T. Pajdla, 2002)

	DESCRIPCION DEL ANALISIS	ALGORITMOS			
		SURF	SIFT	ASIFT	FAST
Según (Plaza Cordero & Zambrano Martínez, 2012)	<p>La evaluación consiste en los diversos keypoints que se obtuvieron con sus respectivos “match” y se realiza la comparación, haciendo uso de los algoritmos.</p> <p>En el último se podrá observar que se hace uso de TILTS, siendo este el número de simulaciones a realizarse.</p>	Un método inspirado en SIFT, pero que trae una nueva funcionalidad: su rapidez.	<p>Es una técnica bastante joven.</p> <p>Método SIFT que solo simula tres parámetros.</p>	Método ASIFT se simulan seis parámetros: el zoom, el ángulo de la cámara en latitud, el ángulo de la cámara en longitud y normaliza los parámetros de traslación, rotación y escala.	
Según (Pérez)	La evaluación consiste en contabilizar el tiempo y	Han demostrado ser	Era muy costosa en tiempo de ejecución,		Descartado este algoritmo debido a

Conte)	cerciorarse de obtener un mínimo de puntos después del RANSAC para diferentes pares de imágenes. La decisión de probarlo en la zona mixta es que su comportamiento supone un reflejo de las zonas de bosque y ciudad a la par.	capaces de encontrar suficientes puntos en ambos tipos de zonas.	así como el número de puntos detectados era demasiado grande y sin un método eficaz para reducir estos.		su ineficiencia en la detección de puntos en las zonas de bosque y el tiempo necesario para su finalización.
--------	--	--	---	--	--

Para la presente investigación se aplicó un algoritmo de detección de punto de interés en las imágenes.

Según (Duarte Villaseñor & Chang Fernandez) en su investigación “*Clasificación de objetos en imágenes usando SIFT*” muestran cuantitativamente que las características locales SIFT son apropiadas para representar clases de objetos. Además, los mismos también muestran la robustez del método obtenido ante variaciones de iluminación, escala, rotación y oclusión.

Según (Zanchettin & Cenedese, 2012-2013) “*Algoritmo di RANSAC: panoramica, confronti e applicazioni*” nos dice que el algoritmo SIFT detecta los puntos estables de un objeto, de tal manera que el mismo objeto se puede reconocer con la varianza de iluminación, escala, rotación y transformaciones afines. Debido a su invariancia al cambio de la iluminación, la escala y la rotación, el algoritmo SIFT se utiliza para la adquisición de imágenes en cualquier condición.

5.3. Aplicar los Algoritmos de Stitching

Conversión de imágenes a puntos.

Técnica Empleada

SURF (acelerarse características robustas) (Yuan's, 2013)

About Hessian

En Surf, simplemente usamos una aproximación al número entero el determinante de detector de burbuja de Hesse.

Dado un píxel, el grupo de acción de este píxel es algo así como:

$$H(f(x, y)) = \begin{bmatrix} \frac{\sigma^2 f}{\sigma x^2} & \frac{\sigma^2 f}{\sigma x \sigma y} \\ \frac{\sigma^2 f}{\sigma x \sigma y} & \frac{\sigma^2 f}{\sigma y^2} \end{bmatrix}$$

Para adaptarnos a cualquier escala, vamos a filtrar la imagen por un núcleo de Gauss, por lo que dado un punto $x = (x, y)$, la matriz de Hesse $H(x, \sigma) \times$

en el punto σ escala se define como:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

Donde $L_{xx}(x, \sigma)$ es la convolución de la derivada de segundo orden de Gauss con la imagen I en el punto x , y de manera similar para $L_{xy}(x, \sigma)$ y $L_{yy}(x, \sigma)$.

About Feature Descriptor

En Surf, utilizamos la suma de la respuesta “the Haar wavelet” alrededor del punto de interés.

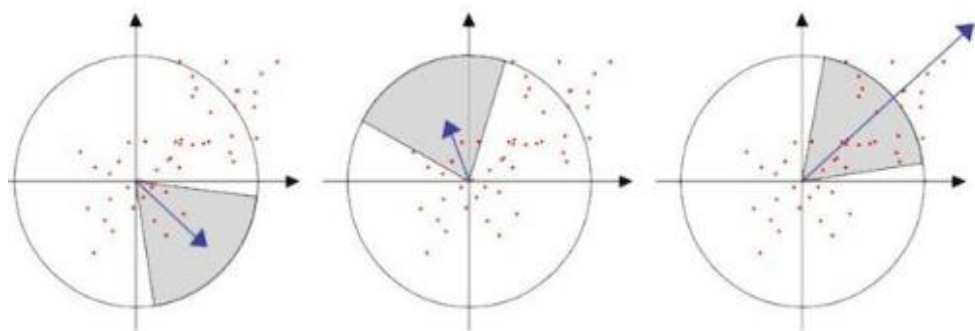


Figura 34. Descripción grafica "Feature Descriptor. (Yuan's, 2013)

Primero calculamos las respuestas “the Haar wavelet” en dirección X e Y dentro de un barrio circular de $6s$ radio alrededor del punto de interés, con s la escala en que se detectó el punto de interés. Calculamos la suma de las respuestas de ondas pequeñas verticales y horizontales en un área de exploración, a continuación, cambiar la orientación de barrido (añadir $\pi / 3$), y volver a calcular, hasta que encontremos la orientación con el mayor valor de la suma, esta orientación es la orientación principal de la función descriptor.

Ahora es el momento de extraer el descriptor. En primer lugar, construimos una región cuadrado centrado alrededor del punto de función, y orientado a lo largo de la orientación principal que ya se dieron anteriormente, el tamaño de esta ventana es de $20s$, s es la escala en que se detectó el punto de interés. En segundo lugar, nos dividimos esta región

con regularidad en pequeñas 4×4 subregiones cuadrados, para cada sub-región, calculamos las respuestas de ondas pequeñas “the Haar wavelet” en 5×5 puntos de muestra regularmente espaciados.

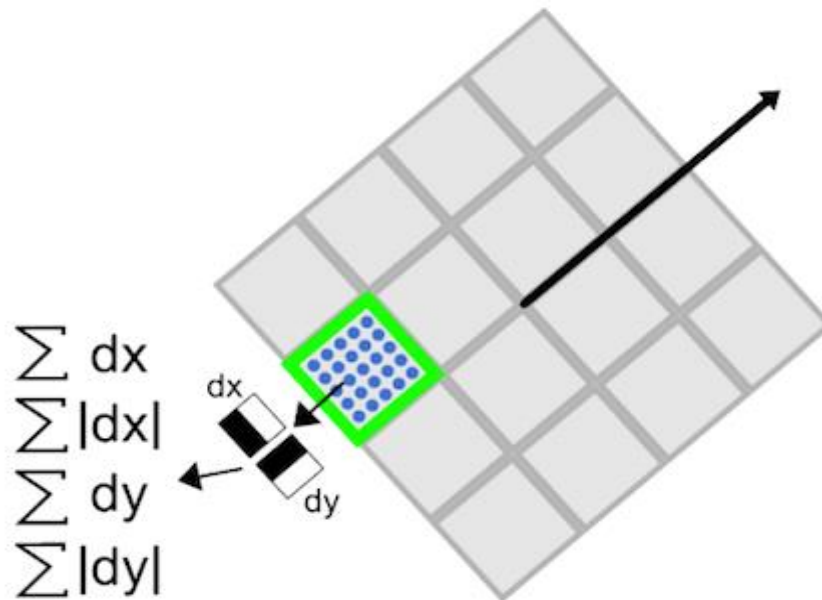
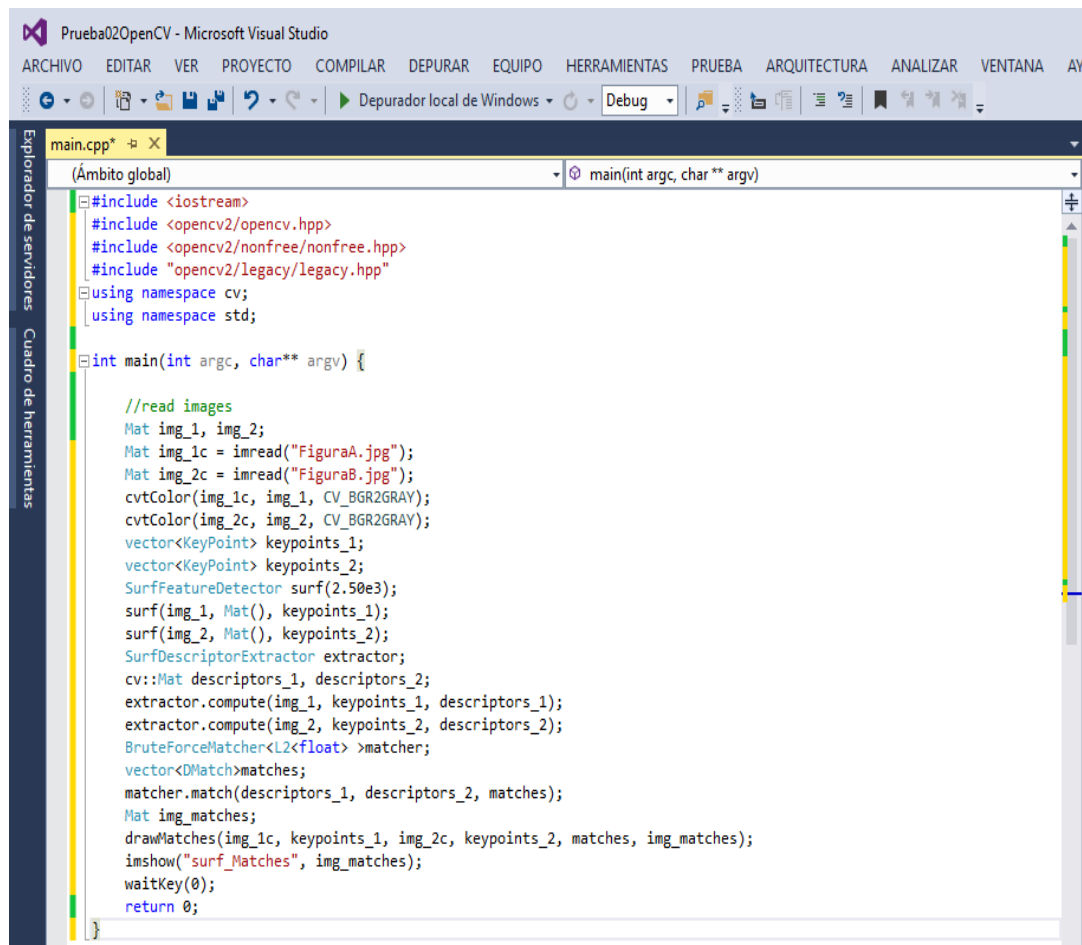


Figura 35. Matriz de puntos de Surf. (Yuan's, 2013)

Extraemos la suma de los valores de las respuestas en ambos x y la orientación y , por otra parte, extraemos la suma de los valores absolutos de las respuestas, por lo tanto, cada sub-región tiene un 4-D descriptor vector v . Concatenación de esto para todos 4×4 subregiones, nuestra descriptora final es un vector 64-D.



```

main.cpp* - Microsoft Visual Studio
ARCHIVO EDITAR VER PROYECTO COMPILAR DEPURAR EQUIPO HERRAMIENTAS PRUEBA ARQUITECTURA ANALIZAR VENTANA AY
Depurador local de Windows - Debug
main.cpp* - X
(Ámbito global) - main(int argc, char** argv)
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/nonfree/nonfree.hpp>
#include "opencv2/legacy/legacy.hpp"
using namespace cv;
using namespace std;

int main(int argc, char** argv) {
    //read images
    Mat img_1, img_2;
    Mat img_1c = imread("FiguraA.jpg");
    Mat img_2c = imread("FiguraB.jpg");
    cvtColor(img_1c, img_1, CV_BGR2GRAY);
    cvtColor(img_2c, img_2, CV_BGR2GRAY);
    vector<KeyPoint> keypoints_1;
    vector<KeyPoint> keypoints_2;
    SurfFeatureDetector surf(2.50e3);
    surf(img_1, Mat(), keypoints_1);
    surf(img_2, Mat(), keypoints_2);
    SurfDescriptorExtractor extractor;
    cv::Mat descriptors_1, descriptors_2;
    extractor.compute(img_1, keypoints_1, descriptors_1);
    extractor.compute(img_2, keypoints_2, descriptors_2);
    BruteForceMatcher<L2<float> >matcher;
    vector<DMatch>matches;
    matcher.match(descriptors_1, descriptors_2, matches);
    Mat img_matches;
    drawMatches(img_1c, keypoints_1, img_2c, keypoints_2, matches, img_matches);
    imshow("surf_Matches", img_matches);
    waitKey(0);
    return 0;
}

```

Figura 36. El algoritmo SURF implementado Visual Studio y OpenCV. Fuente: Elaboración Propia.

PRIMER CONJUNTO IMÁGENES:



Figura 37. Resultado del proceso del algoritmo SURF de Imagen A e Imagen B.
Fuente: Elaboración Propia.



Figura 38. Resultado del proceso del algoritmo SURF de Imagen B e Imagen C.
Fuente: Elaboración Propia

SEGUNDO CONJUNTO DE IMÁGENES:



Figura 39. Resultado del proceso del algoritmo SURF de Imagen W e Imagen X.
Fuente: Elaboración Propia.



Figura 40. Resultado del proceso del algoritmo SURF de Imagen X e Imagen Y.
Fuente: Elaboración Propia.



Figura 41. Resultado del proceso del algoritmo SURF de Imagen Y e Imagen Z.
Fuente: Elaboración Propia.

SIFT (Plaza Cordero & Zambrano Martínez, 2012)

Scale-space extrema detection (Detección de puntos extremos luego de obtener una transformación espacio-escala)

- Función scale-space: $L(x, y, \vartheta)$

Se debe realizar una búsqueda de los keypoints en todas las localizaciones de todas las escalas, para lo cual se usa la función continua $L(x, y, \vartheta)$, convolucionando la imagen $I(x, y)$ y la gaussiana.

$$L(x, y, \vartheta) = G(x, y, \vartheta) * I(x, y)$$

- Función Difference-of-Gaussian: $D(x, y, \vartheta)$

En esta etapa se debe obtener keypoints más estables, por lo tanto, ahora se trabajará con difference-of-gaussian, la cual es simplemente la resta de las imágenes vecinas pero que deben ser de la misma octava.

$$D(x, y, \vartheta) = [G(x, y, k\vartheta) - G(x, y, \vartheta)] * I(x, y)$$

$$D(x, y, \vartheta) = L(x, y, k\vartheta) - L(x, y, \vartheta)$$

Local extrema detection

Ahora lo que se deberá realizar es encontrar los máximos y mínimos locales, para lo cual cada pixel de cada imagen de la pirámide obtenida en el punto anterior, serán comparados con los nueve pixeles de la escala anterior y posterior, además con los ocho pixeles de la misma imagen.

Keypoint localization (Detección de puntos claves)

Se debe almacenar la información de los keypoints como es la octava y su escala, además de la fila y columna, cabe mencionar que se eliminarán los puntos que presenten las siguientes características:

- Supresión de puntos de bajo contraste
- Supresión de puntos situados a lo largo de bordes

Orientation assignment

Se toma un pixel como central y se va a determinar cuál es la orientación, para lo cual seleccionamos una región de 16x16 pixeles alrededor del pixel central, y a cada pixel se calculará su gradiente, usando el módulo e inclinación. Para cada keypoint se deberá agrupar la información en un histograma en 36 orientaciones, el cual posee los valores de orientación que son ponderados usando el valor del módulo y pasando por la ventana circular gaussiana de $\sigma = 1.5 \times \text{escala}$.

Keypoint Descriptor

Se debe realizar un descriptor, para lo cual cada gradiente de la región de 16x16 se deberá rotar como lo especifica en el punto anterior, luego seleccionamos una región de 4x4 pixeles alrededor del pixel central, obteniendo 16 regiones de 8 orientaciones, para no tener cambios bruscos se le pasará mediante una ventana circular gaussiana de $\sigma = 0,5$.

Matching (cálculo de correspondencias)

Al ya tener un descriptor, que es un conjunto de elementos que tienen las principales orientaciones de un keypoint, se deberá determinar si en dos imágenes existen correspondencias, es decir similitud, para lo cual se usa la diferencia euclídea.

$$d_{if_i} = \sqrt{(a_i - b_i)^2}$$

$$n = \# \text{ histogramas} * \# \text{ rotaciones}$$

$$d_{if_{total}} = \sum_{i=1}^n d_{if_i}$$

```

main.cpp*  X
(Ámbito global)  main(int argc, char ** argv)
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/nonfree/nonfree.hpp>
#include "opencv2/legacy/legacy.hpp"

using namespace cv;
using namespace std;

int main(int argc, char** argv){

    //Leer las imágenes
    Mat img_1c = imread("FiguraA.jpg");
    Mat img_2c = imread("FiguraB.jpg");
    Mat img_1, img_2;

    //Transformar las imágenes a escala de grises
    cvtColor(img_1c, img_1, CV_BGR2GRAY);
    cvtColor(img_2c, img_2, CV_BGR2GRAY);
    SIFT sift(50, 5);
    vector<KeyPoint> key_points_1, key_points_2;
    Mat detector;

    //Hacer Sift, encontrar puntos clave
    sift(img_1, Mat(), key_points_1, detector);
    sift(img_2, Mat(), key_points_2, detector);
    SiftDescriptorExtractor extractor;
    Mat descriptors_1, descriptors_2;
    extractor.compute(img_1, key_points_1, descriptors_1);
    extractor.compute(img_2, key_points_2, descriptors_2);

    BruteForceMatcher<L2<float> >matcher;
    vector<DMatch>matches;
    matcher.match(descriptors_1, descriptors_2, matches);

    //Resultado

    Mat img_matches;
    drawMatches(img_1c, key_points_1, img_2c, key_points_2, matches, img_matches);
    imshow("Sift", img_matches);
    waitKey(0);
    return 0;
}

```

Figura 42. El algoritmo SIFT implementado Visual Studio y OpenCV. Fuente: Elaboración Propia.

PRIMER CONJUNTO IMÁGENES:



Figura 43. Resultado del proceso del algoritmo SIFT de Imagen A e Imagen B.
Fuente: Elaboración Propia.



Figura 44. Resultado del proceso del algoritmo SIFT de Imagen B e Imagen C.
Fuente: Elaboración Propia.

SEGUNDO CONJUNTO IMÁGENES:

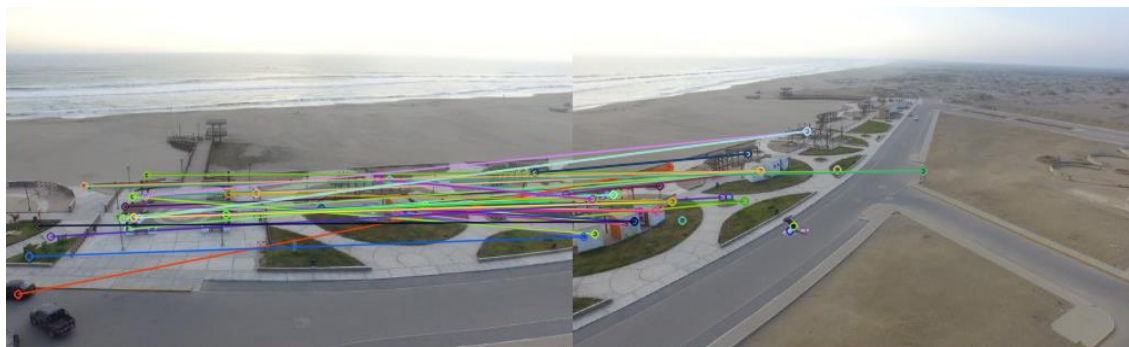


Figura 45. Resultado del proceso del algoritmo SIFT de Imagen W e Imagen X.
Fuente: Elaboración Propia.



Figura 46. Resultado del proceso del algoritmo SIFT de Imagen X e Imagen Y.
Fuente:Elaboracion Propia.

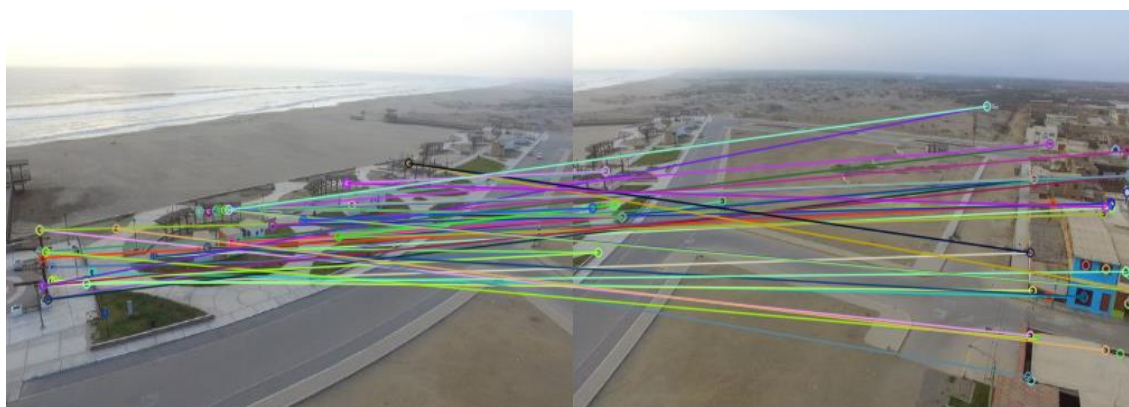


Figura 47. Resultado del proceso del algoritmo SIFT de Imagen Y e Imagen Z.
Fuente:Elaboracion Propia.

Proceso de Correspondencia de los puntos.

Técnica Empleada

RANSAC

En algunas aplicaciones del algoritmo SIFT pueden surgir algunos problemas debido a una evaluación de las características de los puntos significativos. Esto significa que en algunos casos se pueden encontrar los resultados inconsistentes con la realidad, y por lo tanto es necesario analizar y procesar los datos de este algoritmo antes de poder usarlo.

Para filtrar estos valores, se solucionará mediante RANSAC.



Figura 48. Escenario problemático implementando SIFT. Fuente: Elaboración Propia

Si siguiendo con el proceso tomaremos como ejemplo la **Figura 48** se prueba a continuación con RANSAC para identificar las correspondencias correctas. En primer lugar, se analizaron por separado los dos grupos de puntos significativos de las dos imágenes distintas I_1 e I_2 ; Fue aplicado al cálculo de la tendencia lineal de regresión entre sus puntos de vista y RANSAC para localizar los datos en una posición anómala, que no pertenecen a la parte que debe coincidir con la zona común entre las dos fotos.

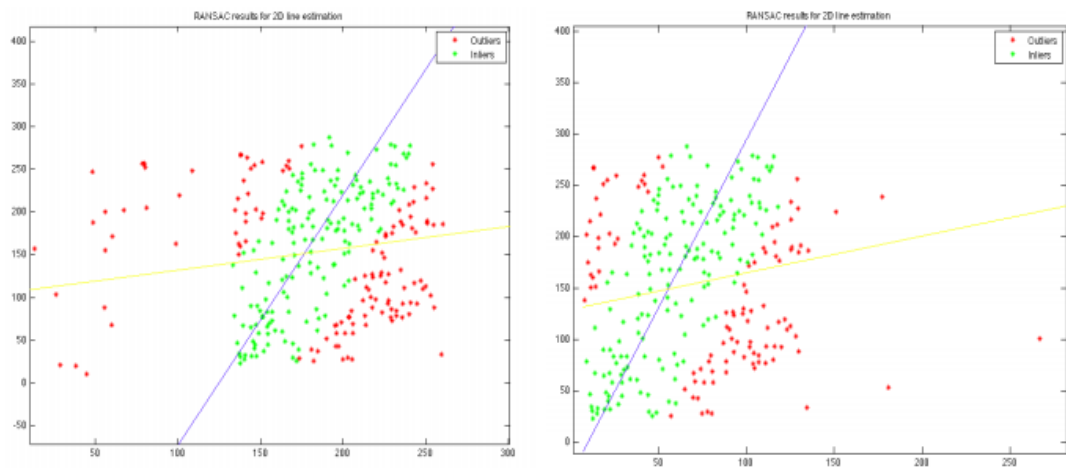


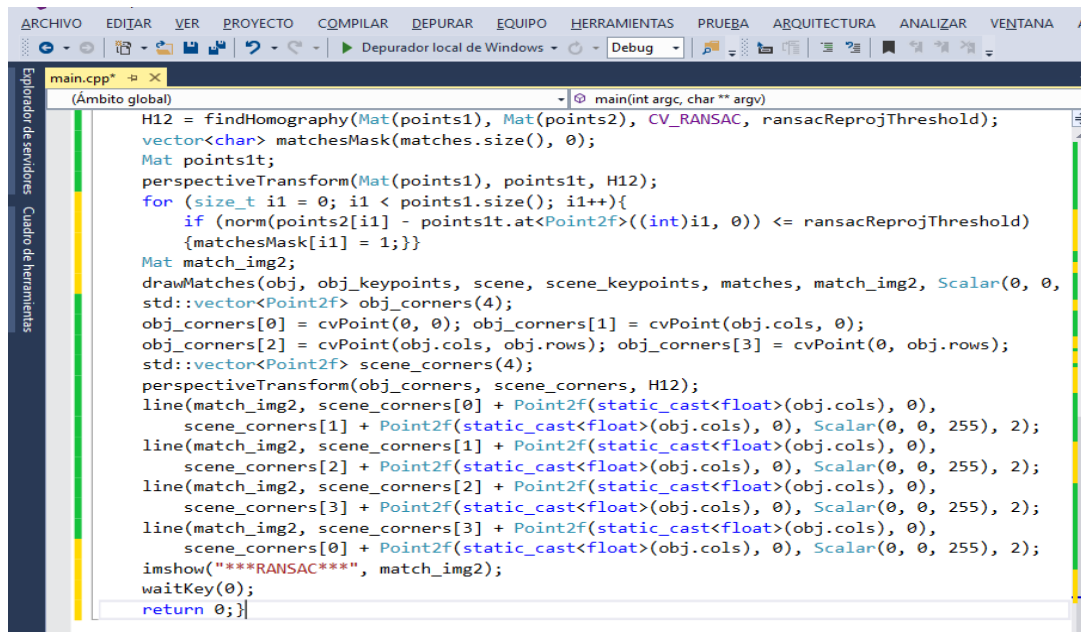
Figura 49. Análisis del Algoritmo RANSAC. (Zanchettin & Cenedese, 2012-2013)

```

main.cpp*  X
(Ámbito global)  main(int argc, char** argv)
#include <iostream>
#include "opencv2/opencv.hpp"
#include "opencv2/core/core.hpp"
#include "opencv2/features2d/features2d.hpp"
#include "opencv2/highgui/highgui.hpp"
using namespace cv;
using namespace std;
int main(int argc, char** argv)
{
    Mat obj = imread("FiguraA.jpg");
    Mat scene = imread("FiguraB.jpg");
    if (obj.empty() || scene.empty()){
        cout << "No se puede Abrir la Imagen!\n";
        return 0;
    }
    vector<KeyPoint> obj_keypoints, scene_keypoints;
    Mat obj_descriptors, scene_descriptors;
    ORB detector;
    detector.detect(obj, obj_keypoints);
    detector.detect(scene, scene_keypoints);
    detector.compute(obj, obj_keypoints, obj_descriptors);
    detector.compute(scene, scene_keypoints, scene_descriptors);
    BFMatcher matcher(NORM_HAMMING, true);
    vector<DMatch> matches;
    matcher.match(obj_descriptors, scene_descriptors, matches);
    Mat match_img;
    drawMatches(obj, obj_keypoints, scene, scene_keypoints, matches, match_img);
    imshow("RANSAC", match_img);
    vector<int> queryIdxs(matches.size()), trainIdxs(matches.size());
    for (size_t i = 0; i < matches.size(); i++){
        queryIdxs[i] = matches[i].queryIdx;
        trainIdxs[i] = matches[i].trainIdx;
    }
}

```

Figura 50. El algoritmo RANSAC implementado Visual Studio y OpenCV PARTE 1. Fuente: Elaboración Propia



```

main.cpp* - x
(Ambito global)
main(int argc, char** argv)
{
    H12 = findHomography(Mat(points1), Mat(points2), CV_RANSAC, ransacReprojThreshold);
    vector<char> matchesMask(matches.size(), 0);
    Mat points1t;
    perspectiveTransform(Mat(points1), points1t, H12);
    for (size_t i1 = 0; i1 < points1.size(); i1++){
        if (norm(points2[i1] - points1t.at<Point2f>((int)i1, 0)) <= ransacReprojThreshold)
            {matchesMask[i1] = 1;}}
    Mat match_img2;
    drawMatches(obj, obj_keypoints, scene, scene_keypoints, matches, match_img2, Scalar(0, 0, 255), 2);
    std::vector<Point2f> obj_corners(4);
    obj_corners[0] = cvPoint(0, 0); obj_corners[1] = cvPoint(obj.cols, 0);
    obj_corners[2] = cvPoint(obj.cols, obj.rows); obj_corners[3] = cvPoint(0, obj.rows);
    std::vector<Point2f> scene_corners(4);
    perspectiveTransform(obj_corners, scene_corners, H12);
    line(match_img2, scene_corners[0] + Point2f(static_cast<float>(obj.cols), 0),
        scene_corners[1] + Point2f(static_cast<float>(obj.cols), 0), Scalar(0, 0, 255), 2);
    line(match_img2, scene_corners[1] + Point2f(static_cast<float>(obj.cols), 0),
        scene_corners[2] + Point2f(static_cast<float>(obj.cols), 0), Scalar(0, 0, 255), 2);
    line(match_img2, scene_corners[2] + Point2f(static_cast<float>(obj.cols), 0),
        scene_corners[3] + Point2f(static_cast<float>(obj.cols), 0), Scalar(0, 0, 255), 2);
    line(match_img2, scene_corners[3] + Point2f(static_cast<float>(obj.cols), 0),
        scene_corners[0] + Point2f(static_cast<float>(obj.cols), 0), Scalar(0, 0, 255), 2);
    imshow("***RANSAC***", match_img2);
    waitKey(0);
    return 0;
}

```

Figura 51. El algoritmo RANSAC implementado Visual Studio y OpenCV PARTE 2.
Fuente: Elaboración Propia.

PRIMER CONJUNTO IMÁGENES:



Figura 52. Resultado del proceso del algoritmo RANSAC de Imagen A e Imagen B.
Fuente: Elaboración Propia.



Figura 53. Resultado del proceso del algoritmo RANSAC de Imagen B e Imagen C.
Fuente: Elaboración Propia.

SEGUNDO CONJUNTO IMÁGENES:

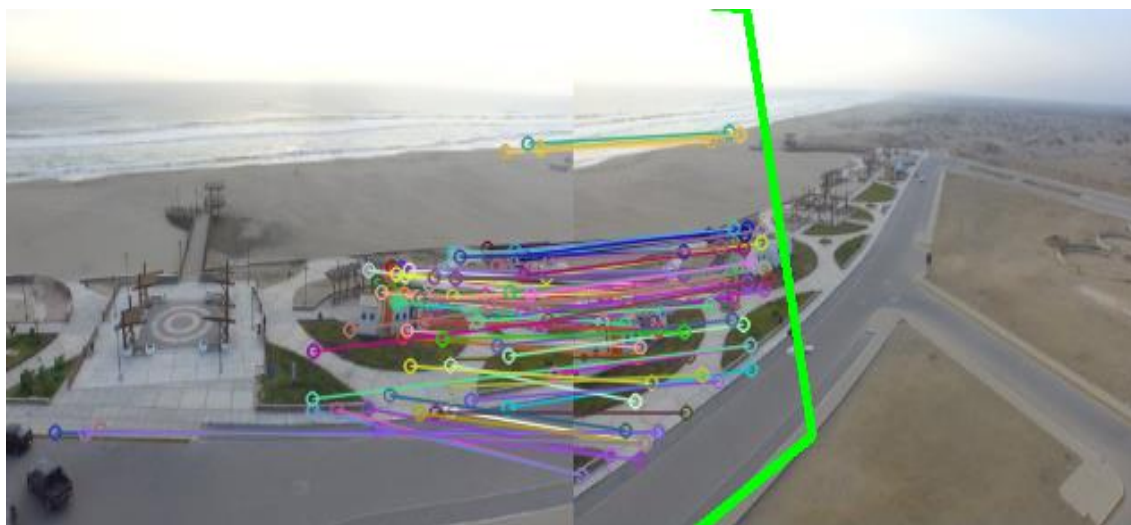


Figura 54. Resultado del proceso del algoritmo RANSAC de Imagen W e Imagen

X. Fuente: Elaboración Propia.



Figura 55. Resultado del proceso del algoritmo RANSAC de Imagen X e Imagen

Y. Fuente: Elaboración Propia.

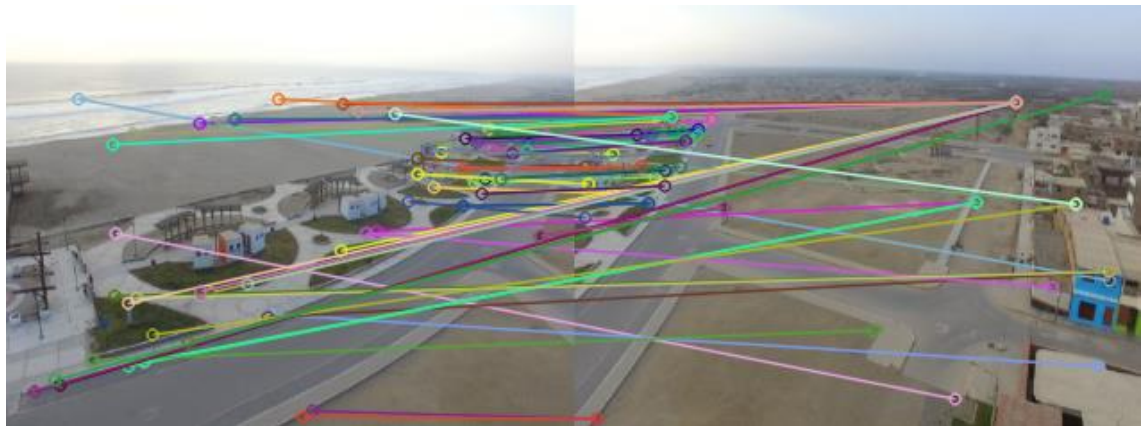


Figura 56. Resultado del proceso del algoritmo RANSAC de Imagen Y e Imagen

Z. Fuente: Elaboración Propia.

5.4. Optimizar la imagen panorámica

TECNICA EMPLEADA

Algoritmo Levenberg Marquardt (L-M)

(Lourakis, 2005) El algoritmo de Levenberg-Marquardt (L-M) es una técnica iterativa que localiza el mínimo de una función multivariable que se expresa como la suma de cuadrados de funciones no lineales de valor real. Se ha convertido en una técnica estándar para problemas cuadrados no lineal, ampliamente adoptado en un amplio espectro de disciplinas. L-M se puede pensar en una combinación de pendiente y el método de Gauss-Newton. Cuando la solución actual está lejos de la correcta, el algoritmo se comporta como un método de descenso más empinado: lento, pero garantizado para converger. Cuando la solución actual está cerca de la solución correcta, se convierte en un método de Gauss-Newton.

MODELO MATEMATICO (Chunyang Zhao & Huaici Zhao, 2016)

Sea:

$$M_{inlier} = \{(m_{1i}, m_{2i}) \mid i = 1, \dots, n\}$$

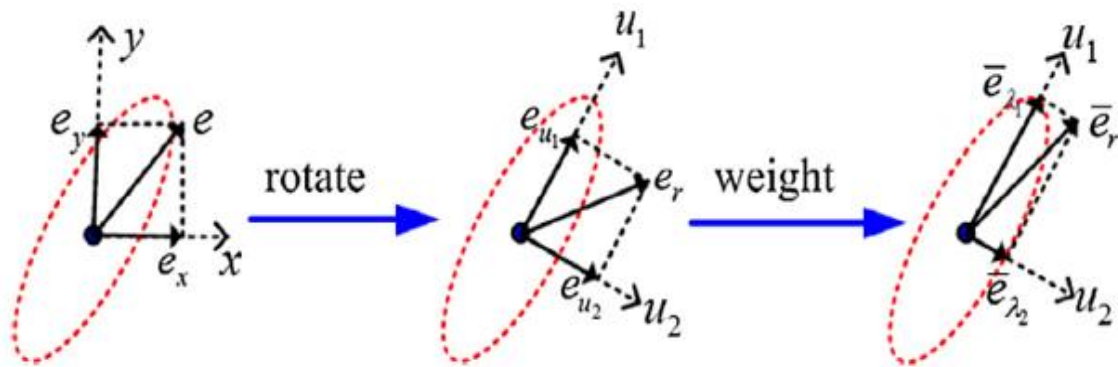
Representan las correspondencias “inliers”, y “m1i”, “m2i” representan la posición de píxel en I1 e I2 respectivamente. Entonces, la función objetiva E utilizada el algoritmo L-M se puede escribir como:

$$E = \arg \min_H \sum_{i=1}^n ||e||_2^2 = \arg \min_H \sum_{i=1}^n ||m_{2i} - H(m_{1i})||_2^2$$

Donde “E” es el vector de obtenido por la aplicación de RANSAC o un vector de imágenes en donde: m1i es la posición del píxel, y H () Es la transformación homográfica.

Sin embargo, la función objetivo tampoco considera las características anisotrópicas y no idénticas del error de localización de características. Por lo tanto, la homografía estimada no es la estimación óptima.

Para resolver el problema, proponemos un nuevo método de refinamiento de la matriz homográfica llamado CW L-M. El método CW L-M propuesto utiliza una nueva función objetiva “E”, considerando el error de localización de características, puede escribirse como:



$$\tilde{E} = \arg \min_H \sum_{i=1}^n \left[\frac{(u_{1i}^T \cdot (m_{2i} - H(m_{1i})))^2}{\lambda_{1i}} + \frac{(u_{2i}^T \cdot (m_{2i} - H(m_{1i})))^2}{\lambda_{2i}} \right]$$

Figura 57. Formula y diseño de funciones de L-M. (Chunyang Zhao & Huaici Zhao, 2016)

Donde $\lambda_{1i}, \lambda_{2i}$ son los valores propios de la matriz de covarianza de la correspondencia de punto clave “I” en las correspondencias M_{inlier} , y $U = [u_{1i} u_{2i}]$ son el auto vector correspondiente.

Como se muestra en la **Figura 57** la nueva función objetivo considera el anisótropo girando el vector de error re-proyectado a la dirección de los ejes de la matriz de covarianza. Mientras tanto, la nueva función objetivo se pondera utilizando valores propios de la matriz de covarianza, que refleja las diferentes contribuciones de diferentes puntos clave. El error proyectado ponderado puede explicarse indicando que cuantos más pequeños son los valores propios de la matriz de covarianza, mayores son las contribuciones a la función objetivo, y viceversa.

Además, los vectores de error re-proyectados son girados a la dirección del eje y son normalizados usando valores propios de la matriz de covarianza. Después de eso, el error de los diferentes puntos clave es isotrópico e idénticamente distribuido, y la suposición del método L-M se satisface, por lo que la solución es óptima.

```

main.cpp
(Ámbito global) main(int argc, char ** argv)

image_points.push_back(cv::Point2d(399, 561));
image_points.push_back(cv::Point2d(337, 297));
image_points.push_back(cv::Point2d(513, 301));
image_points.push_back(cv::Point2d(345, 465));
image_points.push_back(cv::Point2d(453, 469));
std::vector<cv::Point3d> model_points;
model_points.push_back(cv::Point3d(0.0f, 0.0f, 0.0f));
model_points.push_back(cv::Point3d(0.0f, -330.0f, -65.0f));
model_points.push_back(cv::Point3d(-225.0f, 170.0f, -135.0f));
model_points.push_back(cv::Point3d(225.0f, 170.0f, -135.0f));
model_points.push_back(cv::Point3d(-150.0f, -150.0f, -125.0f));
model_points.push_back(cv::Point3d(150.0f, -150.0f, -125.0f));

double focal_length = im.cols;
Point2d center = cv::Point2d(im.cols / 2, im.rows / 2);
cv::Mat camera_matrix = (cv::Mat_<double>(3, 3) << focal_length, 0, center.x, 0
cv::Mat dist_coeffs = cv::Mat::zeros(4, 1, cv::DataType<double>::type);
cout << "Camera Matrix " << endl << camera_matrix << endl;
cv::Mat rotation_vector;
cv::Mat translation_vector;
cv::solvePnP(model_points, image_points, camera_matrix, dist_coeffs, rotation_v
vector<Point3d> nose_end_point3D;
vector<Point2d> nose_end_point2D;
nose_end_point3D.push_back(Point3d(0, 0, 1000.0));
projectPoints(nose_end_point3D, rotation_vector, translation_vector, camera_mat
for (int i = 0; i < image_points.size(); i++)
{circle(im, image_points[i], 3, Scalar(0, 0, 255), -1);}
cv::line(im, image_points[0], nose_end_point2D[0], cv::Scalar(255, 0, 0), 2);
cout << "Rotation Vector " << endl << rotation_vector << endl;
cout << "Translation Vector" << endl << translation_vector << endl;
cout << nose_end_point2D << endl;
cv::imshow("Output", im);
cv::waitKey(0);
  
```

Figura 58. El algoritmo implementado en Visual Studio y OpenCV. Fuente: Elaboración Propia

PRIMER CONJUNTO IMÁGENES:



Figura 59. Imagen Panorámica optimizada con el algoritmo L-M.
 Fuente:Elaboracion Propia.

SEGUNDO CONJUNTO IMÁGENES:



Figura 60. Imagen Panorámica optimizada con el algoritmo L-M. Fuente:Elaboracion Propia

En la **Figura 60** se muestra como resultado una imagen panorámica con la implementación de los algoritmos de Conversión de Puntos (SIFT) y el algoritmo de Correspondencia de puntos (RANSAC), en la Imagen 49 se muestra como resultado una imagen panorámica optimizada implementando el algoritmo L-M.

6. CAPÍTULO VI: CONSIDERACIONES FINALES Y RECOMENDACIONES

6.1. Conclusiones

Los algoritmos determinados para el procesamiento de imágenes en nuestra investigación aumentaron el número de puntos Matching, estos algoritmos SURF, SIFT, RANSAC Y L-M mejoraron la imagen panorámica.

Además, al aplicar estos algoritmos de STITCHING para unir las imágenes digitales acelero la eficiencia del tiempo de Stitching de la imagen panorámica y obtenemos una imagen panorámica de un 83,197 % de información útil, en comparación con el método de la investigación de referencia en los paper, el cual nos gana por un 10 % más de información útil no pudimos superar dicho porcentaje, pero si el tiempo de Stitching el cual debió ser por el entorno de desarrollo, las funciones de la librería OpenCV y de tal manera se aprovechó más los recursos del ordenador.

Los resultados experimentales verificar que la mejora proceso de stitching se realizó mediante la implementación de algoritmos que implementan fórmulas matemáticas y métodos de selección de puntos de referencias en las imágenes poder reducir el error de distorsión de la imagen panorámica.

6.2. Recomendaciones

Una posible mejora a la investigación, es que deben realizar:

- Pruebas con imágenes obtenidas bajo situaciones controladas teniendo en consideración el clima (lluvias, vientos fuertes).
- La posición del lente de la cámara en referencia a la iluminación del sol.
- Las dimensiones de las imágenes que mantenga un similar ancho y alto.
- Una Resoluciones idénticas.

6.3. Referencias

- Alva, R. (2011). *Diseño de notas de laboratorio. La bitácora*.
- Bay, H., Tuytelaars, T., & Van Gool, L. (s.f.). SURF: Speeded Up Robust Features. *ETH Zurich (Katholieke Universiteit Leuven)*, 6-7.
- Candiago, S., Remondino, F., De la Giglio, M., Dubbini, M., & Gattelli, M. (2015). Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images. . *Remote Sensing*, 4026-4047.
- Chen, Y. (14 de mayo de 2013). *System and method for SIFT implementation and optimization* . Obtenido de <https://www.google.com/patents/US8441489>
- Chunyang Zhao , & Huaici Zhao. (2016). Accurate and robust feature-based homography estimation using HALF-SIFT and feature localization error weighting. *EL SEVIER*.
- Cuevas, E., Zaldivar, D., & Perez, M. (2010). *Procesamiento digital de imagenes con MATLAB y Simulink*. Mexico: Alfaomega Ra-Ma.
- DJI. (2016). *DJI The Future of possible*. Obtenido de <https://www.dji.com/es/phantom-3-standard>
- Duarte Villaseñor, M. M., & Chang Fernandez, L. (s.f.). Clasificación de objetos en imagenes usando SIFT. *INAOE*, 8-14.
- Figueiras, A. R. (2002). *Una Panorámica de las Telecomunicaciones*. Madrid: Prentica Hall.
- J. Matas, O. Chum, M. Urban, & T. Pajdla. (2002). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *BMVC* , 3.
- Jinwook Jeong, & Kyungkoo Jun. (2016). A Novel Seam Finding Method Using Downscaling and Cost for Image Stitching. *Hindawi*.

- Jung-Hee Suk, Chun-Gi Lyuh, Sanghoon Yoon, & Tae Moon Roh. (2015). Fixed Homography–Based Real-Time SW/HW Image Stitching Engine for Motor Vehicles. *ETRI Journal*, 1143-1153.
- Lourakis, M. I. (2005). A Brief Description of the Levenberg-Marquardt Algorithm Implemened. *Foundation for Research and Technology*.
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Vancouver, B.C., Canada*, 1-28.
- MathWorks. (1994-2016). *Feature Based Panoramic Image Stitching*. Obtenido de MathWorks: http://es.mathworks.com/help/vision/examples/feature-based-panoramic-image-stitching.html?s_tid=gn_loc_drop
- Mikolajczyk, K., & Schmid, C. (2004). Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 8.
- Minchen Zhu, Weizhi Wang, Bingham Liu, & Jingshan Huang. (2013). Efficient Video Panoramic Image Stitching Based on an Improved Selection of Harris Corners and a Multiple Constraint Corner Matching.
- Montabone, S. (2010). *Beginning Digital Image Processing: Using Free Tools for Photographers*. USA: Apress.
- Morel, J.-M., & Yu, G. (2011). *Image Processing on line (IPOL)*, 5-6.
- Pérez Conte, R. (s.f.). Mapas mediante imágenes tomadas. *TFG EN ENGINYERIA INFORMÀTICA, ESCOLA D'ENGINYERIA (EE), UNIVERSITAT AUTÒNOMA DE BARCELONA (UAB)*, 4-5.
- Plaza Cordero, A. M., & Zambrano Martínez, J. L. (2012). Estudio y Selección de las Técnicas SIFT, SURF y ASIFT de Reconocimiento de Imágenes para el Diseño de un Prototipo en Dispositivos Móviles. *EST 2012*, 4-5.

- Rahul Raguram, Jan-Michael Frahm, & Marc Pollefeys. (2008). A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. *Springer-Verlag Berlin*, 500–513.
- Tao Yang, Jing Li, Jingyi Yu, Sibing Wang, & Yanning Zhang. (2015). Diverse Scene Stitching from a Large-Scale Aerial Video Dataset.
- Yuan's, E. (11 de 2013). *Eric Yuan's Blog*. Obtenido de <http://eric-yuan.me/surf/>
- Y.Xiong, & K.Pulli. (2010). “Fast panorama stitching for high-quality panoramic images on mobile phones. *IEEE Transactions on Consumer Electronics*, 298–306.
- Zahonero, J. (2014). *Programacion en C,C++,Java y UML*. Mexico,D.F: Mc Graw Hill Education.
- Zanchettin, A., & Cenedese, A. (2012-2013). Algoritmo di RANSAC: panoramica, confronti e applicazioni. *Tesi di Laurea Triennale in Ingegneria dell'Informazione*, 32-44.
- Zhong Qu, Si-Peng Lin, Fang-Rong Ju, & Ling Liu. (2015). The Improved Algorithm of Fast Panorama Stitching for Image Sequence and Reducing the Distortion Errors. *Mathematical Problems in Engineering*, 1-12.