

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**REGISTRO DE IMÁGENES PARA AGRICULTURA DE PRECISIÓN
USANDO LENGUAJE C**

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

Jorge Enrique Barba Escudero

ASESOR: Andrés Flores Espinoza

Lima, marzo del 2012

Resumen

La tesis expuesta abarca la elaboración de un sistema de registro para imágenes que fueron tomadas por un aeromodelo sobre un campo de cultivo de papa, el cual es parte de un ambicioso proyecto de agricultura de precisión que actualmente se está realizando el CIP (Centro Internacional de la Papa) y la PUCP.

Dicho sistema generará un mosaico con las imágenes obtenidas en los espectros infrarrojo y rojo visible que posteriormente serán procesadas para obtener mapas de productividad, los cuales brindan información espacial detallada acerca la zona de cultivo con el objetivo de reducir el consumo innecesario y el impacto ambiental.

En el primer capítulo se expondrá el estado del arte del proyecto a nivel global, posteriormente a nivel local y finalmente se analizarán las necesidades, las cuales enmarcan el objetivo del presente estudio.

En el segundo capítulo se exponen las tecnologías vigentes para la solución del problema, considerando ventajas y desventajas se seleccionará la más eficiente que cumpla con las especificaciones y los objetivos para solucionar el problema propuesto.

En el tercer capítulo se dará fundamento teórico a la solución propuesta y se detallará la implementación.

Finalmente en el cuarto capítulo se someterá al algoritmo a diversas pruebas usando imágenes con cambios en el punto de vista, cambios de escala, desenfoque, compresión JPEG e iluminación.

Índice general

Resumen	1
Introducción	3
Capítulo 1. Aspectos generales de la problemática	4
1.1. Entorno General	4
1.2. Entorno Específico	5
1.3. Entorno Organizacional	5
1.4. Proceso: Análisis del problema	5
Capítulo 2. Sensado remoto y tecnologías en el registro de imágenes	8
2.1. Presentación del asunto previo	8
2.2. Agricultura de precisión	10
2.3. OpenCV	11
2.4. Modelo Teórico	12
Capítulo 3. Planteamientos para el diseño del registro de imágenes	13
3.1. Objetivos de la investigación	13
3.2. Modelo del sistema	13
3.3. Descripción del funcionamiento del sistema	13
Capítulo 4. Pruebas y análisis de resultados	22
4.1. Muestra de resultados	23
4.2. Análisis de resultados	25
4.3. Análisis de resultados globales	26
4.4. Análisis de resultados por imágenes de prueba	28
4.5. Discusión de resultados	34
4.6. Análisis de resultados para imágenes de campo de cultivo usando transformaciones geométricas	35
Conclusiones	39
Recomendaciones	40
Bibliografía	41

Introducción

La actividad productiva en el país ha sufrido grandes modificaciones durante el siglo XX. Hasta la década de los años 50 la economía del país se basó en actividades agrícolas, una minería en crecimiento y manufactura deficiente. Durante la segunda mitad del siglo XX ocurrió una expansión de centros urbanos y se desarrollaron otras importantes actividades como la construcción, finanzas, comercio y otros servicios.

Actualmente las actividades agrícolas aportan solo el 7.6% del PBI nacional, que corresponde a la mitad de lo que era hace 4 décadas[10], contrariamente a otras actividades económicas primarias como la minería, pesca y actividades económicas terciarias.

A pesar de las deficiencias, de los cuatro productos agrícolas indispensables en la alimentación humana (papa, maíz, trigo, arroz), en el Perú se producen masivamente tres de ellos, como son la papa (costa y sierra), maíz (sierra y selva) y el arroz (costa y sierra). Sin embargo debido a políticas comerciales erróneas se ha arruinado a los productores, a tal punto que el Perú es calificado como un país importador neto de alimentos (PINA)[17].

En el caso de la papa, el cultivo más representativo del Perú, el rendimiento promedio ha aumentado de 11.6 a 12.0 t/ha entre 1999 y 2003 respectivamente, sin embargo aún estamos por debajo del rendimiento promedio de países de la región, por ejemplo Colombia (16 t/ha), Brasil (15 t/ha) y Chile (15 t/ha). Este bajo rendimiento se debe a problemas tecnológicos relacionados a la sanidad[2].

Debido a esta problemática, el Centro Internacional de la Papa (CIP) desarrolla proyectos enfocados en mejorar las técnicas de producción agrícolas, para así aumentar la productividad de una determinada área de cultivo y mejorar la calidad del producto.

Uno de los proyectos en desarrollo es la agricultura de precisión, la cual se refiere al monitoreo y control electrónico aplicado a la recolección de información y su procesamiento como soporte para la toma de decisión y para el uso espacial y temporal de insumos en la producción de cultivos y que actualmente es utilizado en varios países desarrollados.

Capítulo 1

Aspectos generales de la problemática

Actualmente debido al constante avance de la ciencia en el campo del procesamiento digital, surgen nuevas aplicaciones que aumentan la eficiencia de los procesos de producción.

Particularmente en agricultura, existe un proceso relativamente nuevo que involucra tecnologías vigentes y que es la base para la toma de decisiones en cuanto a necesidades de insumos en el área de cultivo, a este proceso se le conoce como agricultura de precisión.

La agricultura de precisión se basa en el sensado de ciertas características del suelo y del cultivo para así obtener una mejor cosecha, minimizando así costos en insumos[1]. La implementación del proceso de agricultura de precisión involucra procesamiento digital, fundamentalmente procesamiento de imágenes, cuyas definiciones y algoritmos se usaran en capítulos posteriores.

En el presente capítulo se abarcará la problemática del sensado remoto y procesamiento digital de imágenes, específicamente el registro de imágenes aplicado a imágenes de cultivos.

1.1. Entorno General

La agricultura de precisión se basa en una ciencia conocida como sensado remoto, la cual consiste en obtener información de algún objeto, superficie o algún fenómeno que sea de interés de investigación sin la necesidad de contacto físico. Esta ciencia aplicada a la agricultura consiste en la interpretación de imágenes digitales del área de cultivo que posteriormente serán analizadas, modificadas y almacenadas utilizando un sistema de información geográfica.

El proceso de sensado remoto para agricultura de precisión se realiza en países desarrollados usando satélites, los cuales debido al avance de la tecnología están cada vez mejor equipados y debido a ello generan imágenes de mayor calidad y resolución.

Existen diversos satélites comerciales, como por ejemplo IKONOS o SPOT, que ofrecen servicio de captura de imágenes, lo cual es una alternativa para las organizaciones que no cuentan con satélites propios. Sin embargo resulta muy costoso contratar sus

servicios, por ejemplo una imagen de 4 bandas (rojo, verde, azul e infrarrojo) que abarque una región 100 Km² cuesta más de US \$ 2,000.00 (dos mil dólares americanos). (Ver anexo 1)

1.2. Entorno Específico

Actualmente el Perú se encuentra en crecimiento económico, lo que contribuye al desarrollo de la industria. Además se encuentran vigentes tratados de libre comercio con varias de las mayores potencias mundiales (EE.UU., China, Unión Europea, etc), debido a esto se espera en los próximos años un aumento de la demanda en productos agrícolas, lo cual involucrará una constante mejora de la calidad de los productos exportados. Para mantener un crecimiento sostenido en la producción se requiere una constante inversión en tecnología (procesos y equipos usados) y capacitaciones constantes al personal, sin embargo debido a la precaria actividad agrícola en el Perú los productores tienen un presupuesto para inversión bastante limitado.

1.3. Entorno Organizacional

El Centro Internacional de la Papa (CIP) es un centro de investigación científica de papa, camote y otras raíces o tubérculos. El principal objetivo del CIP es reducir la pobreza y buscar una alimentación sostenida para los países en vías de desarrollo. Su sede principal se encuentra en el distrito de La Molina y además también cuenta con estaciones experimentales en Huancayo y San Ramón[12].

El CIP trabaja conjuntamente con la PUCP para implementar procesos de agricultura de precisión, sin embargo aun se encuentran en fase experimental. En dichos ensayos se han usado globos aerostáticos, aunque recientemente se ha optado por usar aeromodelos radio controlados. Para el análisis de las imágenes aéreas obtenidas el CIP cuenta con un software propio descrito en [15], sin embargo el software presenta principalmente dos dificultades en el manejo, que son el tiempo de procesamiento y la dificultad de interacción con el usuario. Ésto se puede corregir para facilitar los procedimientos usados y así aumentar la eficiencia y calidad de la producción.

1.4. Proceso: Análisis del problema

1.4.1. Descripción. El registro de imágenes es un proceso que consiste en alinear imágenes de una misma escena tomadas en distintos momentos, distintos puntos de vista y/o diferentes sensores[20].

El proceso de registro de imágenes es parte fundamental del procesamiento de éstas en agricultura de precisión, el cual se debe realizar en dos oportunidades con las imágenes obtenidas. En primer lugar se utiliza para generar un registro de imágenes en la misma banda (ya sea en espectro rojo cercano o infrarrojo). En segundo lugar se utiliza para generar un registro de imágenes multiespectrales (imágenes en espectro rojo cercano con infrarrojo), sin embargo nos centraremos en el procesamiento de las imágenes en la misma banda para generar un registro de éstas.

En el CIP se cuenta con software desarrollado en MATLAB para el registro de imágenes en la misma banda obtenidas de la toma aérea, el cual es un algoritmo muy completo capaz de crear un registro muy útil[15]. Sin embargo este software presenta tiempos de procesamiento prolongados, los cuales son incluso mayores a 5 minutos por cada par de imágenes de 640×480 , esto multiplicado por la cantidad de imágenes obtenidas en cada vuelo genera tiempos difíciles de manejar.

El personal del CIP se encuentra preparado para el análisis de imágenes y toma decisiones usando a la información proporcionada, sin embargo el software no cuenta con una interfaz gráfica, lo cual dificulta su uso al personal del CIP. Ésto debido a que los técnicos no necesariamente conocen detalles acerca de los métodos y procedimientos usados, entonces se requiere un entorno que abstraiga y facilite el procesamiento para que así el software diseñado pueda ser utilizado por personal sin conocimientos en esta área de la tecnología.

1.4.2. Análisis. El software diseñado en MATLAB consta de funciones, de las cuales podemos mencionar los siguientes:

- Lectura de imágenes
- Extracción de descriptores
- Búsqueda de descriptores comunes
- Detección de función de transformación
- Aplicación de función de transformación
- Muestra de resultados

En particular si bien el algoritmo funciona correctamente, como ya se mencionó el tiempo de procesamiento es largo.

En resumen, las funciones que presentan este problema son la extracción de descriptores, la búsqueda de descriptores comunes en ambas imágenes y la aplicación de la función de transformación.

Además se presentan dificultades al usarlo ya que el programa realizado carece de interfaz gráfica, siendo esto una dificultad para el usuario final, el cual no cuenta con los conocimientos técnicos acerca de los procedimientos realizados.

Para disminuir el tiempo de procesamiento se analizarán otras alternativas de algoritmos para generar un software más manejable, así como también se analizarán otros lenguajes de programación y librerías con funciones optimizadas.



Capítulo 2

Sensado remoto y tecnologías en el registro de imágenes

En este capítulo se expondrán conceptos teóricos implicados en el análisis de los problemas planteados en el capítulo 1. Además se compararán las soluciones propuestas y usando criterios debidamente fundamentados se elegirá la solución más conveniente del problema propuesto.

2.1. Presentación del asunto previo

2.1.1. Estado del arte. El sensado remoto actualmente se usa en multitud de aplicaciones [16], entre las más importantes podemos mencionar las siguientes:

- Monitoreo y evaluación ambiental (crecimiento urbano)
- Detección del cambio global (capa de ozono, deforestación, calentamiento global)
- Agricultura (condiciones de cultivo, erosión del suelo)
- Exploración de recursos no renovables (petróleo, minerales)
- Recursos naturales renovables (océanos, bosques, humedales)
- Meteorología (dinámica atmosférica, predicción del tiempo)
- Cartografía (topografía)
- Aplicaciones militares (tácticas y estrategias militares)

Para el caso de agricultura, los instrumentos empleados consisten en sensores ópticos (cámaras fotográficas digitales), los cuales dependiendo de la aplicación varían en la longitud de onda que pueden captar (ancho de banda).

El rango espectral usado para estas aplicaciones abarca las longitudes de onda de 620-670nm y 841-876nm, estos rangos corresponden a los espectros rojo visible e infrarrojo cercano respectivamente[16].

Las imágenes obtenidas en el rango de las longitudes de onda antes mencionadas se capturan desde el cielo a una cierta altura, la cual depende de la resolución espacial deseada. Las imágenes tomadas a mayor altura abarcarán mayor área, sin embargo la resolución espacial disminuirá y a menor altura se obtendrá mayor resolución espacial y menor área capturada.

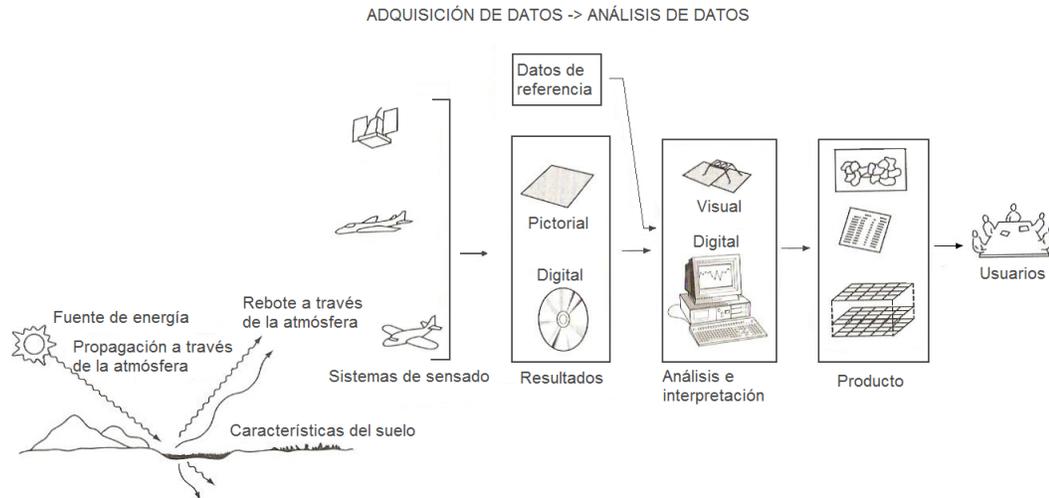


FIGURA 2.1.1. Sensores remotos de recursos. [24]

El resultado consiste en diversas imágenes que contienen información espacial acerca de la calidad del suelo de cultivo, minerales, escasez de agua, así como también predecir posibles plagas.

El proceso de sensores remotos se puede ver en la Figura 2.1.1.

2.1.2. Estado de la investigación. La representación digital de una imagen a color consiste en 3 matrices de datos, cuyas dimensiones son $M \times N$, en donde M y N representan la cantidad de píxeles en largo y ancho respectivamente. Cada matriz corresponde al contraste de uno de los 3 colores fundamentales rojo, verde y azul, esta representación se conoce como RGB.

Las imágenes se representan (y almacenan) usando un valor de intensidad para cada uno de los 3 colores fundamentales, el cual es un número comprendido entre 0 y 255. Estos datos de intensidad de colores se aplican para cada píxel de la imagen, donde 0 corresponde al nivel más bajo de intensidad y 255 corresponde al nivel más alto (saturación). Entonces al variar estos datos de las matrices es posible manipular la imagen. [5]

Existen numerosos lenguajes de programación que permiten procesar imágenes, entre los cuales podemos mencionar lenguajes enfocados a la implementación de algoritmos matemáticos como MATLAB, MATHEMATICA y MAPLE. Estos lenguajes contienen funciones implementadas para procesamiento de imágenes, además al ser de alto nivel proveen una sintaxis más familiar a la humana facilitando la programación.

Existen lenguajes de programación para software en general como por ejemplo C, C++, COBOL, Visual Basic, FORTRAN, PASCAL y el lenguaje ensamblador.

Los lenguajes antes mencionados son usados para distintas aplicaciones[8], algunas de las principales se mencionan a continuación:

- COBOL es usado para administración de negocios y procesa gran cantidad de contenido alfanumérico, no contiene funciones de cálculo complejo.
- Visual Basic es un lenguaje de programación diseñado específicamente para crear aplicaciones que se ejecutan en Windows.
- C es usado principalmente por programadores expertos para crear sistemas complejos para PCs.
- BASIC es un lenguaje fácil de usar, se utiliza preferentemente para la didáctica de programación.
- El lenguaje ensamblador es usado para computadores y microprocesadores específicos, pero son difíciles de escribir.
- De los lenguajes de programación antes mencionados, el más usado por los programadores ha sido el lenguaje C debido a que posee ventajas sobre los anteriores[7], algunas se mencionan continuación:
- Es un lenguaje flexible, que contiene órdenes, operaciones y funciones con las que se puede escribir cualquier clase de programa.
- Se puede usar C para aplicaciones complejas como sistemas operativos, compiladores o sistemas de tiempo real.
- Es un lenguaje de programación portable, es decir puede escribirse para un tipo particular de hardware y puede usarse en otros sin mayor modificación.

Con lenguaje C es posible la programación de la mayoría de procesadores modernos. Es actualizado continuamente, agregando nuevas bibliotecas, ahora C puede soportar aplicaciones de bases de datos, gráficos, procesamiento de texto, comunicaciones, etc.

2.2. Agricultura de precisión

La definición más simple de agricultura de precisión establece que ésta corresponde a un grupo de tecnologías que permiten la aplicación de insumos agrícolas, tales como fertilizantes, semillas, plaguicidas, etc., en forma variable dentro de un potrero, de acuerdo a los requerimientos y/o potencial productivo de varios sectores homogéneos, pre-definidos dentro del mismo[22]. Para esta tecnología se emplean:

- Satélite
- Aeronaves radio controladas

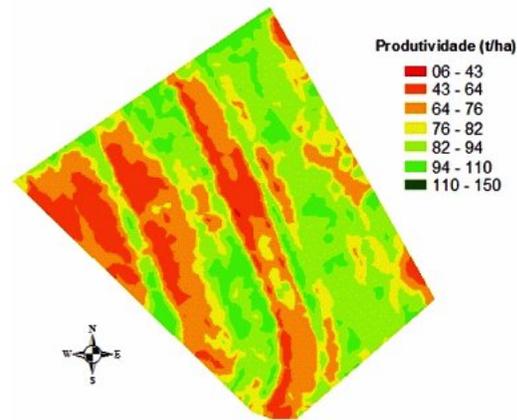


FIGURA 2.2.1. Mapa del rendimiento de la caña de azúcar en un área de 43 ha.[23]

- Aeronaves autónomas (UAV)

Los dispositivos utilizados para estos propósitos son:

- Sensores ópticos
- Altímetros
- Unidad de posicionamiento global (GPS)

Los objetivos de la agricultura de precisión son:

- Mejora de la calidad del producto
- Detección de déficit de insumos (agua, fertilizante)
- Prevención de posibles plagas
- Ahorro económico debido al consumo exacto de insumos

El resultado es un mapa de rendimiento de la superficie a analizar como se aprecia en la Figura 2.2.1. Esta imagen muestra áreas con deficiencia de insumos. La interpretación permite maximizar la calidad del producto y minimizar las pérdidas de insumos químicos agrícolas, reduciendo así también el impacto ambiental.[3]

2.3. OpenCV

OpenCV es un recurso libre de visión por computadora bajo licencia BSD escrito en C y C++ que puede ser ejecutada en Windows, Linux y Mac OS X diseñado para realizar aplicaciones en tiempo real sobre visión por computadora optimizando los recursos de ésta. Fue desarrollada por Intel y comprende más de 500 funciones. Uno de sus principales objetivos es proporcionar una forma rápida y fácil de crear complejas aplicaciones de visión por computadora.[18]

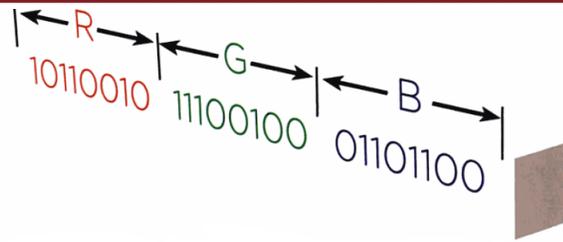


FIGURA 2.4.1. Los 24 bits que hay “detrás” de un píxel de una imagen a color[14]

2.4. Modelo Teórico

Una imagen es representada mediante una matriz de puntos, conocidos como píxeles (acrónimo de picture element), los cuales son representados mediante notación binaria, con “unos” y “ceros”[8]. Cada píxel es almacenado como un conjunto de bits, el tipo de imagen más elemental consiste en un solo bit por píxel (que representan el blanco y negro), tradicionalmente a esto se le llama lith en fotografía química.

Si hubiese más de un bit por píxel (por ejemplo 8 bits o 1 byte) se podrían almacenar más valores, con lo que se podría representar diversas tonalidades intermedias entre blanco y negro, lo cual se conoce como “escala de grises”.

Para el caso de las imágenes a color, estas son representadas y definidas mediante tonalidades de 3 colores fundamentales, los cuales son rojo, verde y azul (modelo RGB), que en diversas intensidades representan cada píxel de la imagen a color. Para el modelo RGB se utiliza 3 bytes por cada píxel (24 bits) y cada uno representa la intensidad de un color fundamental. La Figura 2.4.1 muestra la representación de un píxel.

Debido al aumento de la capacidad de procesamiento de las computadoras se permite la manipulación de la imagen con mayor eficiencia. Además la continua mejora de algoritmos usados para el procesamiento de las imágenes permite extraer más eficientemente información que no se podrían obtener con la información en crudo.

Capítulo 3

Planteamientos para el diseño del registro de imágenes

3.1. Objetivos de la investigación

3.1.1. Objetivo principal. Desarrollar un sistema de registro de imágenes de tomas aéreas para agricultura de precisión.

3.1.2. Objetivos específicos.

- Diseñar e implementar un programa en lenguaje C que permita hacer un registro de imágenes para su uso en agricultura de precisión.
- Diseñar una interfaz sencilla para la manipulación del programa.
- Minimizar los pasos de ejecución del programa para un fácil uso por el usuario.

3.2. Modelo del sistema

El diagrama de la Figura 3.2.1 explica el sistema de registro de imágenes que se implementará.

El proceso de registro de imágenes se dividirá principalmente en los siguientes algoritmos:

- Selección de puntos de control: Fast-Hessian
- Extracción de Descriptores: SURF
- Análisis de correspondencia: Nearest Neighbor
- Estimación de la Función de transformación: RANSAC

3.3. Descripción del funcionamiento del sistema

3.3.1. Selección de puntos de control. La detección de puntos de control está basada en el determinante de la matriz Hessiana, que está definido de la siguiente manera para un punto $X = (x, y)$.

$$(3.3.1) \quad H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}$$

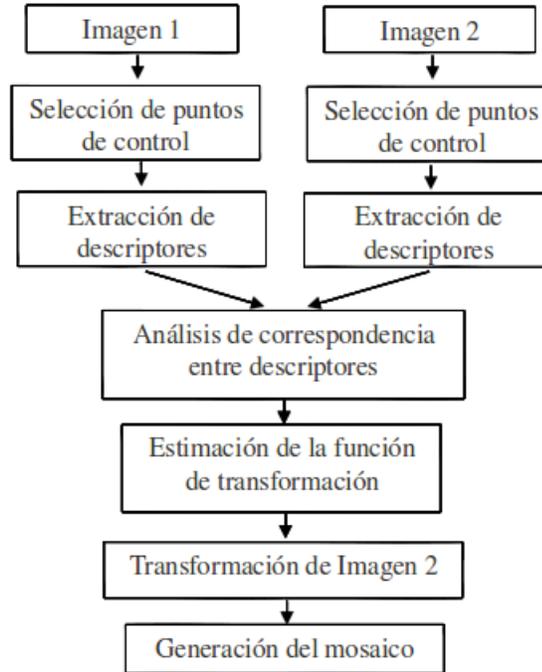


FIGURA 3.2.1. Diagrama de bloques que muestra el desarrollo del registro de imágenes

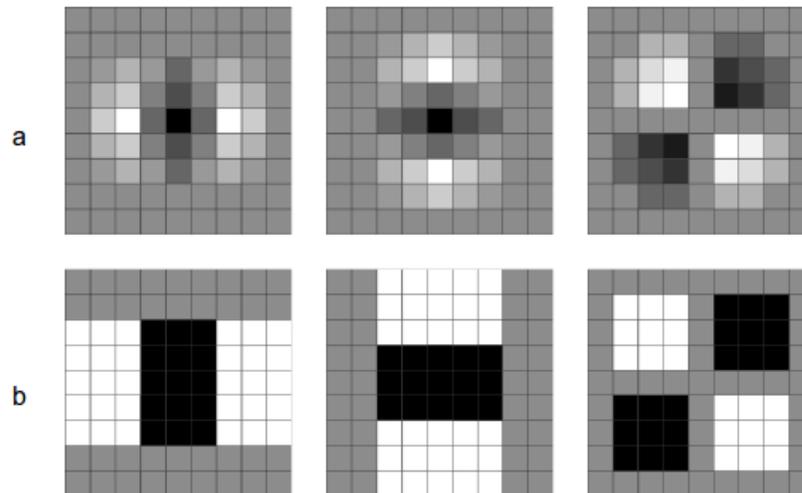


FIGURA 3.3.1. Aproximaciones de los Laplacianos del Gaussiano. Fila a) segunda derivada gaussiana, usada para obtener L_{xx} , L_{xy} , L_{yy} . Fila b) aproximación promediada usada para calcular D_{xx} , D_{xy} , D_{yy} .

Dado un punto $X=(x, y)$ en la imagen I, la matriz Hessiana H de dicho punto es obtenida mediante la convolución de la segunda derivada gaussiana de desviación estandar σ con la imagen I, de donde se obtiene $L_{xx}(x, y, \sigma)$, similarmente para $L_{xy}(x, y, \sigma)$ y $L_{yy}(x, y, \sigma)$. Los kernels utilizados se muestran en la Figura 3.3.1

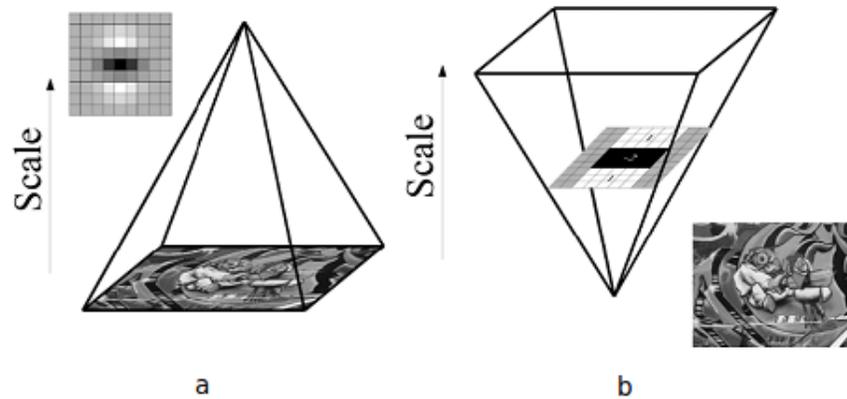


FIGURA 3.3.2. a) representación habitual de la representación Espacio-Escala. b) Enfoque del SURF en la cual varía el tamaño del filtro.[4]

En la 3.3.1 se observa la aproximación de la derivada de segundo orden con $\sigma = 1,2$ y representa la escala más baja (más alta resolución espacial) de la representación Espacio-Escala. Para mayor eficiencia en el computo se utilizan las aproximaciones D_{xx} , D_{xy} , D_{yy} . Luego se obtiene el determinante de la matriz Hessiana multiplicando por un factor de 0.9 al valor obtenido de las derivadas cruzadas:

$$(3.3.2) \quad \det(H_{aprox}) = D_{xx} \cdot D_{yy} - (0,9 D_{xy})^2$$

La convolución de la imagen con los kernels mostrados en la Figura 3.3.1 puede ser calculada de una forma computacionalmente eficiente empleando integral de imagen, definida en [4]. Este nuevo enfoque se conoce como “Fast Hessian”.

La representación espacio-escala es implementada computacionalmente usando pirámide de imágenes. Según Lowe [9] conforme se avance en los niveles de la pirámide se debe también submuestrear la imagen. Sin embargo el nuevo enfoque de la pirámide de imágenes propuesto por Herbert Bay en [6] considera a la imagen de tamaño fijo y va aumentando de tamaño los filtros usados (véase imagen 3.3.1). La pirámide de imágenes es dividida en octavas, las cuales a su vez contienen escalas. Conforme aumenta la escala en una misma octava, el tamaño del filtro debe ir aumentando de tamaño, sin embargo al aumentar de octava los incrementos de tamaño del filtro de deben duplicar con respecto a la anterior octava.

Para la primera escala se considera un filtro de 9x9, que corresponde a $\sigma = 1,2$. En la segunda escala se considera un filtro de 15x15 y su desviación estandar es calculada usando relación 3.3.3.

$$(3.3.3) \quad \sigma_{aprox} = \text{tamaño actual del filtro} \cdot \frac{\text{escala base del filtro}}{\text{tamaño base del filtro}}$$

Luego para localizar los puntos de interés, primero se debe umbralizar, y todos los puntos por debajo de ese valor son descartados. Luego se eligen los máximos valores de la comparación de cada píxel con sus 26 vecinos, 8 de la misma capa, 9 de la superior y 9 de la inferior. Por último los puntos localizados se deben interpolar en espacio y escala para determinar su precisión. Para hacer esto debemos expresar la matriz Hessiana como la expansión de Taylor centrada en el punto de interés.

$$(3.3.4) \quad H(X) = H + \frac{\partial H^T}{\partial X} \cdot X + \frac{1}{2} X^T \frac{\partial^2 H}{\partial X^2}$$

En la ecuación 3.3.4 H es la matriz Hessiana aproximada y $X = (x, y, \sigma)$. Derivando la función e igualandola a 0 se llega a la ecuacion 3.3.5.

$$(3.3.5) \quad \hat{x} = -\frac{\partial^2 H^{-1}}{\partial H^2} \cdot \frac{\partial H}{\partial X}$$

Si \hat{x} es mayor a 0.5 en cualquiera de sus 3 coordenadas (x, y o σ) entonces se debe ajustar la localización y ejecutar nuevamente la interpolación. El proceso se repite hasta que las 3 coordenadas sean menores a 0.5 o que se ejecute el número máximo de interpolaciones predeterminadas.

3.3.2. Extracción de descriptores.

3.3.2.1. Asignación de la orientación del descriptor. Con el objetivo de ser invariante a la rotación se asigna una orientación a cada punto de control hallado. Primero se debe calcular la respuesta wavelet Haar en la dirección x e y (Figura 3.3.3) en una vecindad circular de radio 6σ de la imagen I (donde σ es la escala alrededor del punto de interés), en estos casos también se utiliza la integral de imagen. El tamaño del wavelet es 4σ . La respuesta obtenida es promediada con un Gaussiano centrado en el punto de interés con $\sigma = 2,5$. Después de esto se obtiene las componentes paralelas al eje x e y. La orientación dominante se obtiene calculando la suma de un sector circular con ángulo $\pi/3$ alrededor del punto de interés (Figura 3.3.4), el sector donde se encuentre el mayor vector será la orientación del descriptor. [4]

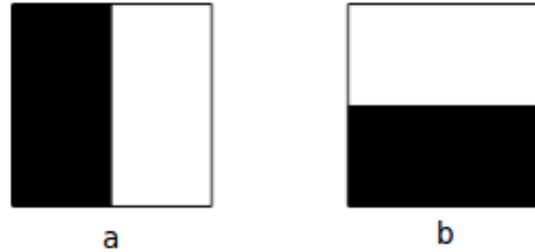


FIGURA 3.3.3. Haar wavelet. a) es usada para calcular la respuesta en el eje x, b) para el eje y.[4]

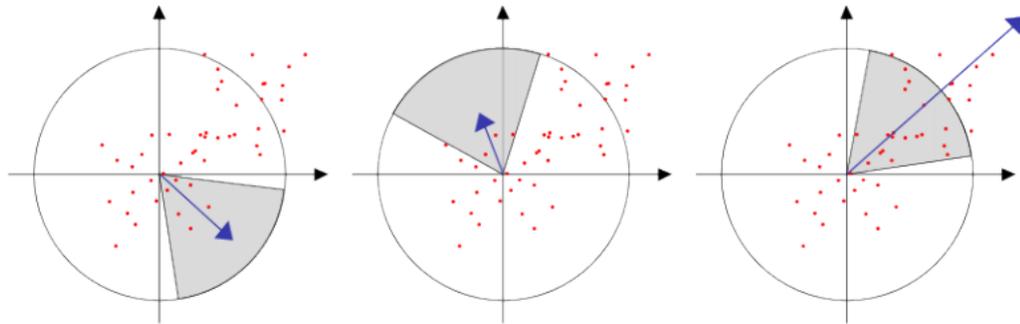


FIGURA 3.3.4. Asignación de la orientación. Los vectores dentro de la ventana de ángulo $\pi/3$ se suman. La orientación corresponde al mayor vector obtenido.[4]

3.3.2.2. Extracción de componentes del descriptor. Para extraer las componentes del descriptor, se debe generar un cuadrado de tamaño 20σ centrado en el punto de control orientado a lo largo de la orientación del descriptor. Luego esta ventana es dividida en 4×4 sub-regiones, de las cuales en cada una se calcula la respuesta wavelet Haar a 25 puntos regularmente espaciados, se suman las respuestas paralelas a los ejes x e y obteniéndose el vector de la ecuación 3.3.6.

$$(3.3.6) \quad v_{\text{subregión}} = \left[\sum dx \quad \sum dy \quad \sum |dx| \quad \sum |dy| \right]$$

Entonces cada sub-región tiene 4 valores, éstos multiplicados por las 16 sub-regiones, se obtiene un vector de longitud 64. El descriptor resultante es invariante a la rotación, escala y cambios de brillo.

3.3.3. Estableciendo correspondencia entre puntos de control. Los puntos de control tendrán correspondencia si la distancia entre los descriptores es menor a cierto umbral, a esto se le conoce como vecino más cercano (Nearest Neighbor).

Entonces D_A (Descriptor del punto A), es el vecino más cercano de D_B (descriptor punto B) si la distancia entre ellos es menor a cierto umbral.

Otro método parecido al anterior es usando también la distancia del segundo vecino más cercano, si la distancia entre D_A y D_B es menor a la distancia entre D_A y D_C , donde D_C es el descriptor segundo vecino más cercano, entonces existe una correspondencia entre ambos descriptores [19].

$$(3.3.7) \quad \frac{\|D_A - D_B\|}{\|D_A - D_C\|} \leq NNDR$$

donde NNDR (Nearest Neighbor Distance Ratio) es el valor umbral.

3.3.4. Estimación de la función de transformación. Para esto se utiliza el algoritmo RANSAC (RANDOM SAMPLE CONSENSUS), el cual es un método robusto para estimar los parámetros de la función de transformación. Consiste en elegir grupos de cuatro puntos y calcular la matriz de homografía H usando DLT (Direct Linear Transformación), luego se elige la solución con la mayor cantidad de puntos cercanos (inliers) con una tolerancia de píxeles ε píxeles.

3.3.5. Transformación de imágenes. En álgebra lineal, existen diversas transformaciones lineales (geométricas) [21].

3.3.5.1. Translación. La única diferencia entre las imágenes es la translación a lo largo de los ejes x e y.

$$\begin{aligned} u &= x + a_{00} \\ v &= y + b_{00} \end{aligned}$$

donde x e y son la posición de cada píxel de la imagen, a_{00} y b_{00} corresponden a la cantidad de píxeles desplazados en los ejes x e y respectivamente, por último u y v corresponden a las coordenadas de la imagen transformada.

3.3.5.2. Translación + dilatación. Considera además la dilatación en los ejes x e y.

$$\begin{aligned} u &= cx + a_{00} \\ v &= cy + b_{00} \end{aligned}$$

donde c corresponde al factor de dilatación.

3.3.5.3. *Translación + rotación.*

$$u = x \cos\theta + y \sin\theta + a_{00}$$

$$v = -x \sin\theta + y \cos\theta + b_{00}$$

donde θ corresponde al ángulo de rotación de la imagen.

3.3.5.4. *Transformación de afinidad.* La transformación de afinidad emplea seis parámetros. Ésto permite obtener en un eje un efecto de alargamiento y en el otro eje un efecto de encogimiento. Es la forma más general de transformación lineal.

$$u = a_{10}x + a_{01}y + a_{00}$$

$$v = b_{10}x + b_{01}y + b_{00}$$

donde a_{10} , a_{01} , b_{10} y b_{01} corresponden a los factores de alargamiento y encogimiento de los ejes.

3.3.5.5. *Transformación de perspectiva.* Es una transformación no geométrica que requiere ocho parámetros y es usada cuando es visto un objeto plano desde un punto de vista fijo.

$$u = \frac{a_{10}x + a_{01}y + a_{00}}{c_{10}x + c_{01}y + 1}$$

$$v = \frac{b_{10}x + b_{01}y + b_{00}}{c_{10}x + c_{01}y + 1}$$

La transformación de perspectiva es la que se obtendrá luego de emplear el algoritmo RANSAC y se usará para proyectar las imágenes tomadas a otro plano y así poder efectuar el mosaico de las imágenes.

Las imágenes obtenidas sufren cambios de perspectiva debido al movimiento del aeromodelo durante el recorrido de la toma de imágenes (Figura 3.3.5.5). Sin embargo la transformación más resaltante es la de translación.

Una vez obtenida la función de transformación, esta se le aplica a la segunda imagen, la cual quedará alineada con respecto a la primera. Luego se superponen las dos imágenes y los píxeles resultantes en el traslape será el promedio de ambos píxeles en cada una de las imágenes.

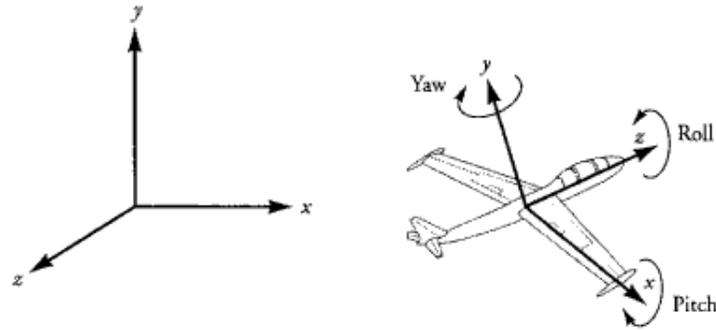


FIGURA 3.3.5. Ejes coordenados de un aeromodelo[13], los cambios en estos ejes generan cambios de perspectiva en las imágenes adquiridas

Para realizar el proceso usando varias imágenes, se deberá generar el mosaico de dos primeras imágenes y luego repetir el procedimiento entre la imagen resultante y la siguiente imagen, un ejemplo del resultado de este proceso se observa en la Figura 3.3.6.



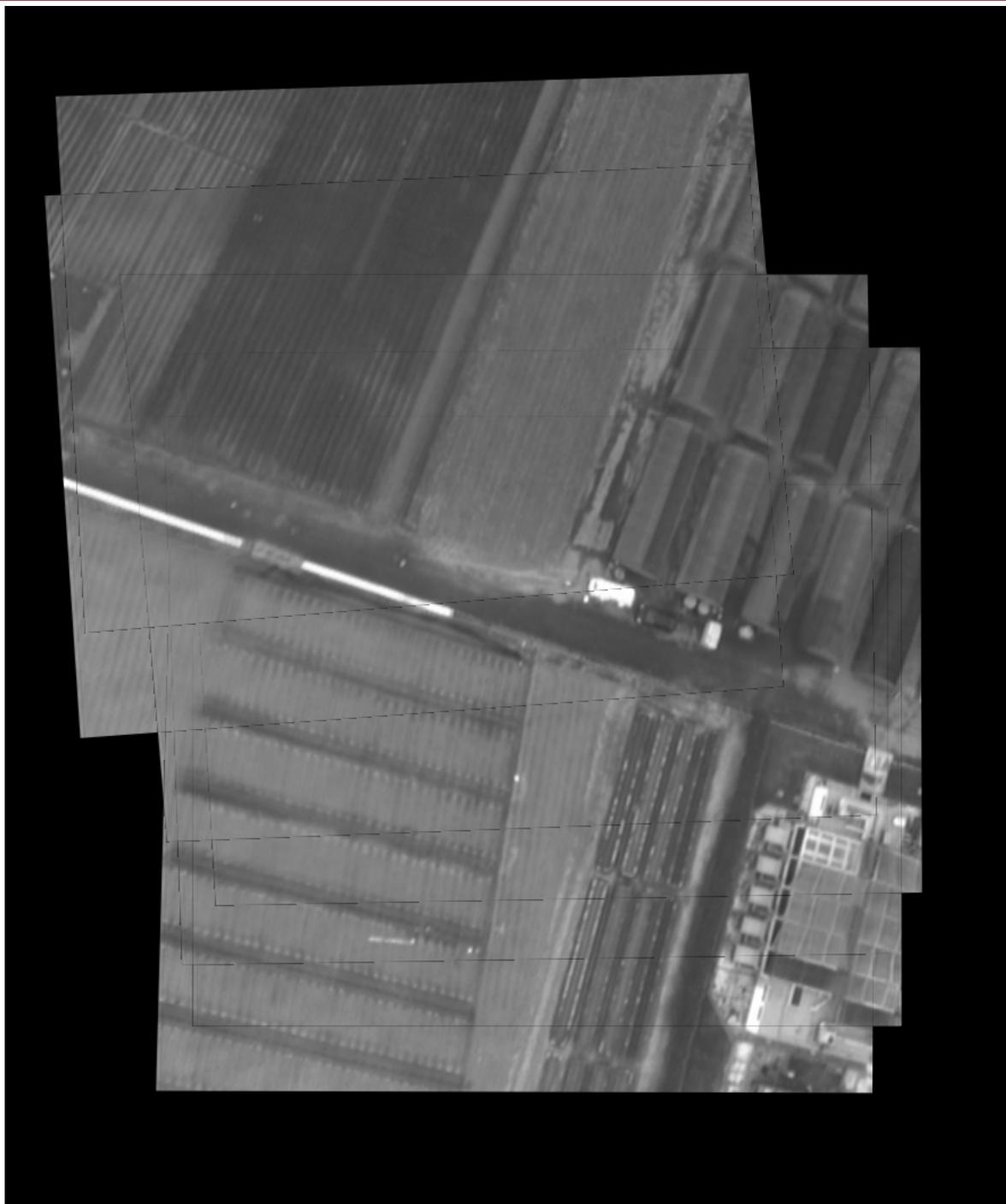


FIGURA 3.3.6. Mosaico generado a partir de 7 imágenes

Capítulo 4

Pruebas y análisis de resultados

En el presente capítulo se analizará de forma cuantitativa el desempeño del conjunto de algoritmos utilizado en el registro de imágenes, se expondrán algunas fórmulas que permiten cuantificar de forma objetiva, y de esta manera comparar, según diversos parámetros, el desempeño del algoritmo.

Se analizaron registros de 2 imágenes consecutivas de vuelo tomadas de una base de datos de 400 imágenes de los campos de cultivo del CIP.

De estas 400 imágenes la mitad corresponde al espectro infrarrojo cercano, el resto corresponde a espectro rojo visible.

Estas imágenes tienen resolución VGA, representadas en escala de grises y fueron tomadas por un aeromodelo que se desplazaba a una velocidad aproximada de 60 Km/h y capturaba imágenes cada 10 ms.

Para el análisis se cuenta con las imágenes de campo de cultivo en formato JPG y BMP, las cuales serán analizadas bajo diversos parámetros como son el número de correspondencias. Según [25] el error se puede cuantificar con la siguiente fórmula.

$$e = \sum_{i,j} [I_i(f_i(X)) - I_j(f_j(X))]^2$$

Donde I_i e I_j son un par de imágenes con traslape, f_i y f_j son las respectivas transformaciones globales de las imágenes. Para nuestro análisis se ha transformado una imagen y se ha mantenido la segunda como referencia, entonces:

$$f_j(X) = X$$

Y si dividimos el error entre la cantidad de píxeles traslapados se obtiene un error promedio:

$$\bar{e} = \frac{e}{\text{cantidad píxeles traslapados}}$$

Los resultados se comparan para distintos valores de NNDR (definido en 3.3.3).

4.1. MUESTRA DE RESULTADOS

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	774	774	774
Cant. de descriptores SURF imagen 2	1051	1051	1051
Descriptores en común	93	181	362
Error	6882345	6947278.5	7121660
Píxeles en común	268899	268899	268899
Error promedio	25.59	25.85	25.53
Tiempo de procesamiento (seg.)	2.6	2.62	2.6

CUADRO 1. Análisis en formato BMP con 87% de área en común

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	789	789	789
Cant. de descriptores SURF imagen 2	1060	1060	1060
Descriptores en común	90	180	361
Error	8790698	7679498.5	13046396
Píxeles en común	270575	269712	268099
Error promedio	32.48	28.47	48.66
Tiempo de procesamiento (seg.)	2.59	2.64	2.62

CUADRO 2. Análisis en formato JPEG con 87% de área en común

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	438	438	438
Cant. de descriptores SURF imagen 2	1051	1051	1051
Descriptores en común	20	56	130
Error	5650645	4854210.5	4391170
Píxeles en común	123481	123059	122960
Error promedio	45.76	39.44	35.71
Tiempo de procesamiento (seg.)	2.37	2.39	2.43

CUADRO 3. Análisis en formato BMP con 40% de área en común

4.1. Muestra de resultados

Los algoritmos fueron implementados en lenguaje C con la ayuda de la librería OpenCV bajo en sistema operativo GNU/LINUX con un procesador Intel Dual Core de 1.73 GHz.

En la tabla 1 se analizó imágenes con un alto porcentaje de área en común mayor al 50% para imágenes en formato BMP.

En la tabla 2 tabla se realiza el análisis con las imágenes en formato de compresión JPEG.

La tabla 3 y 4 muestran las pruebas realizadas en imagenes con 40% de área en común en formato BMP y JPEG respectivamente

4.1. MUESTRA DE RESULTADOS

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	442	442	442
Cant. de descriptores SURF imagen 2	1060	1060	1060
Descriptores en común	18	50	145
Error	7192502	6554147	7287975
Píxeles en común	123154	123035	122410
Error promedio	58.40	53.27	59.53
Tiempo de procesamiento (seg.)	2.32	2.46	2.53

CUADRO 4. Análisis en formato JPG con 40 % de área en común

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	183	183	183
Cant. de descriptores SURF imagen 2	389	389	389
Descriptores en común	21	54	102
Error	25943180	28059494	28365414
Píxeles en común	269434	271175	269473
Error promedio	96.28	103.47	105.26
Tiempo de procesamiento (seg.)	2.27	2.29	2.31

CUADRO 5. Análisis en formato BMP con 87 % de área en común

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	180	180	180
Cant. de descriptores SURF imagen 2	391	391	391
Descriptores en común	26	53	93
Error	27339486	25513176	25533180
Píxeles en común	267964	268098	267937
Error promedio	102.02	95.16	95.29
Tiempo de procesamiento (seg.)	2.28	2.3	2.31

CUADRO 6. Análisis en formato BMP con 87 % de área en común

Las siguientes pruebas muestran los resultados obtenidos aplicando el algoritmo propuesto a imágenes en el espectro rojo visible que han sido tomadas consecutivamente y por lo tanto presentan un alto porcentaje de área en común.

La tabla 5 muestra los resultados de la prueba para imágenes en formato BMP con 87 % de área en común.

La tabla 6 muestra los resultados de la misma prueba pero ahora aplicada a imágenes con formato JPEG.

La tabla 7 y 8 muestran los resultados de las pruebas aplicadas a imágenes con área en común de 37 % en formato BMP y JPEG respectivamente.

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	553	553	553
Cant. de descriptores SURF imagen 2	389	389	389
Descriptores en común	19	48	ND
Error	16730696	17129294	ND
Píxeles en común	115028	115780	ND
Error promedio	145.44	147.94	ND
Tiempo de procesamiento (seg.)	2.34	2.35	

CUADRO 7. Análisis en formato BMP con 37 % de área en común

NNDR	0.4	0.6	0.8
Cant. de descriptores SURF imagen 1	554	554	554
Cant. de descriptores SURF imagen 2	391	391	391
Descriptores en común	12	ND	ND
Error	18012120	ND	ND
Píxeles en común	116062	ND	ND
Error promedio	155.19	ND	ND
Tiempo de procesamiento (seg.)	2.32		

CUADRO 8. Análisis en formato JPEG con 37 % de área en común

4.2. Análisis de resultados

En primer lugar se puede resaltar que el factor NNDR es un componente crítico en el sistema de registro de imágenes y la determinación de la función de transformación depende de ciertos valores de NNDR, como se puede observar en el último caso, ya que no se pudo encontrar función de transformación.

En los últimos casos se intentó modificar el algoritmo disminuyendo los puntos de control para encontrar una función de transferencia que registre las imágenes, sin embargo el algoritmo halló una función de transformación que en la mayoría de casos no realizaba un correcto registro desde un punto de vista subjetivo, y además esto se podía apreciar en el dato del error, ya que salía considerablemente alto.

Como ya se pudo observar en las tablas, se ha hallado las diferencias a partir de 3 valores de NNDR que son 0.4, 0.6 y 0.8, dichos valores son números intermedios y se realizó el análisis en ellos debido a que fuera del rango medido se ha encontrado mayor incidencia de imágenes mal registradas.

Se puede observar que el valor 0.6 es el de mejor respuesta en la mayoría de los casos presentados, ya que se obtiene menor error promedio como en el caso del análisis mostrado en la tablas 2, 4 y 6, sin embargo en otros se mantiene entre los valores obtenidos usando 0.8 y 0.4, como se puede observar en las tablas 1, 3 y 5.

En los datos obtenidos, la diferencia entre el tiempo de procesamiento es mínima, sólo de algunas centésimas de segundo. Además en todos los casos se ha mejorado considerablemente con respecto al algoritmo utilizado en [15].

Se puede observar que en los casos que se trabajó con imágenes JPEG se ha obtenido un error mayor al que se obtuvo en los casos que se trabajó con imágenes BMP ya que debido a la compresión el descriptor disminuye sus características invariantes.

En los casos de imágenes en el rango rojo visible se ha tenido mayores dificultades para formar el registro, ya que el algoritmo SURF detecta una menor cantidad que cuando se trabaja con imágenes en el rango infrarrojo.

Además de esto en el último caso presentado, el algoritmo no pudo encontrar una matriz de transformación que cumpla con las necesidades, es por ello que se requiere mayor precisión al momento de analizar éstas imágenes.

4.3. Análisis de resultados globales

Para esta prueba se realizó un programa que registre dos imágenes consecutivas de la base de datos ya mencionada para los valores de NNDR igual a 0.8, 0.6 y 0.4 respectivamente y se obtuvieron los siguientes resultados:

Para el espectro infrarrojo cercano:

- Con NNDR = 0.8 se obtuvo el registro de las 200 imágenes, de las cuales 23 obtuvieron un error mayor a 50.
- Con NNDR = 0.6 se obtuvo el registro de las 200, de las cuales 7 imágenes obtuvieron un error mayor a 50.
- Con NNDR = 0.4 se obtuvo el registro de las 200, de las cuales 8 imágenes obtuvieron un error mayor a 50.

Para el espectro rojo visible:

- Con NNDR = 0.8 se obtuvo el registro de las 178 imágenes, de las cuales 41 obtuvieron un error mayor a 50.
- Con NNDR = 0.6 se obtuvo el registro de las 173, de las cuales 23 imágenes obtuvieron un error mayor a 50.
- Con NNDR = 0.4 se obtuvo el registro de las 153, de las cuales 26 imágenes obtuvieron un error mayor a 50.

En las pruebas se ha analizado si el error es mayor a 50, ya que según las imágenes registradas, se ha obtenido un error entre 10 y 30, un error mayor a 50 lo más

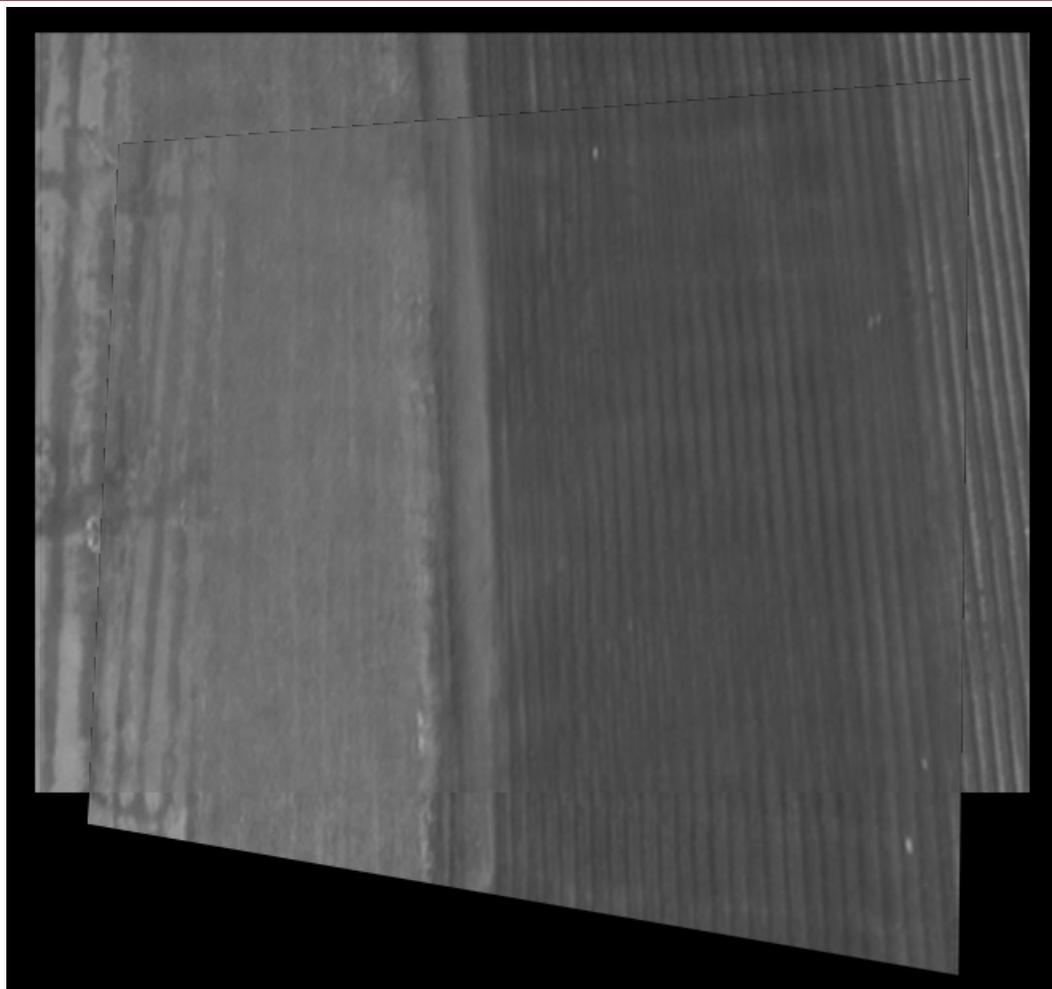


FIGURA 4.3.1. Ejemplo de imagen incorrectamente registrada (error promedio 64.63)

probable es que no halla un correcto registro, un ejemplo se puede observar en las Figuras 4.3.1 y 4.3.2.

Cabe resaltar que en imágenes en espectro infrarrojo se ha tenido varios registros con error mayor a 50 debido a que en varias imágenes de los campos de cultivo han sido de escenas con variaciones mínimas en intensidad, lo cual no permitía encontrar descriptores para hallar la correspondencia, para que no ocurran casos como este el CIP instala pequeñas “marcas” que reflejen la luz en espectros IR y rojo cercano para poder encontrar mayor número de descriptores.

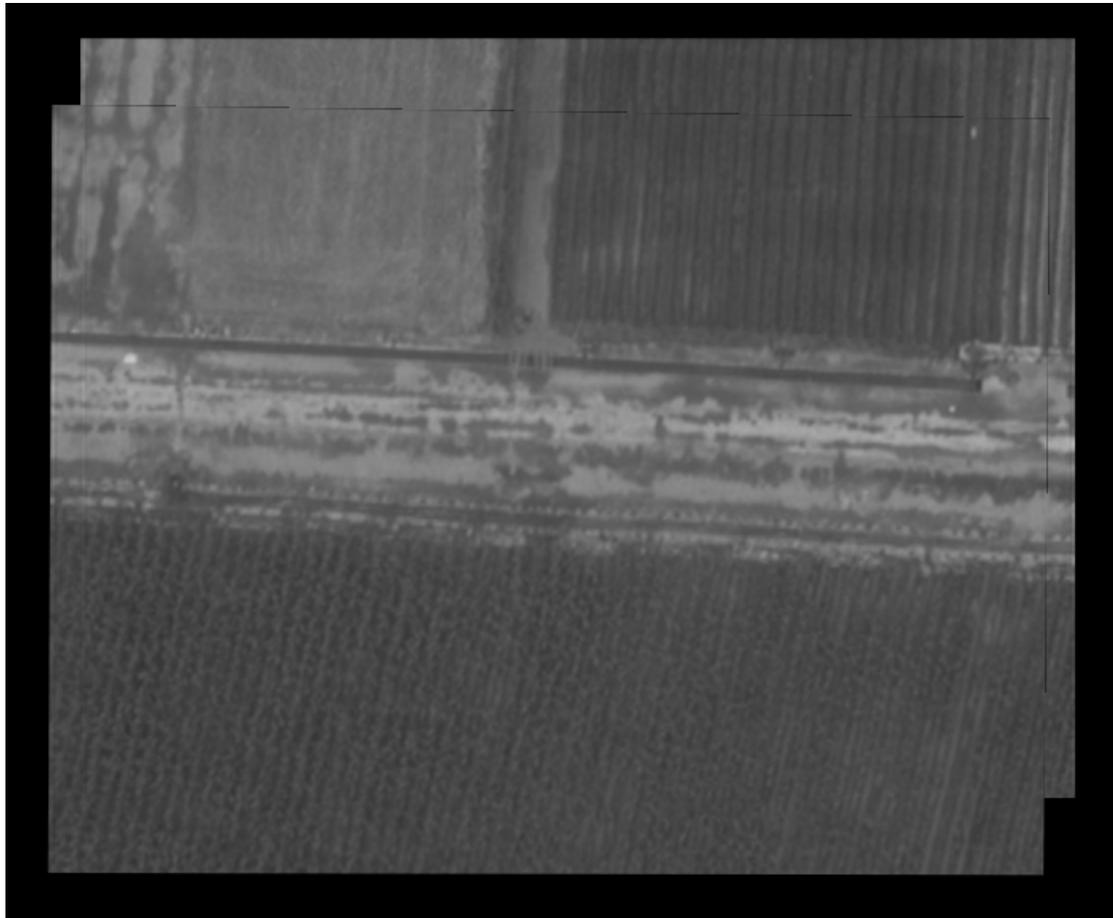


FIGURA 4.3.2. Ejemplo de imagen correctamente registrada (error promedio 10.15)

4.4. Análisis de resultados por imágenes de prueba

El siguiente análisis se desarrolló con imágenes de pruebas obtenidas de [11], las cuales son utilizadas para medir el desempeño de algoritmos que trabajen con regiones afines, dicho algoritmo consiste en hallar la función de transformación en un par de imágenes de 5 variaciones en la intensidad del cambio respectivo.

Luego se compara la verdadera función de transformación con la obtenida con el algoritmo.

En las siguientes tablas se analiza para cada par de imágenes, se trabajó con 8 grupos de 6 imágenes cada una, donde la primera imagen de cada grupo se transformó con respecto a las otras 5 de su grupo, obteniéndose 5 funciones de transformación por cada grupo, de las cuales se muestra el error (en porcentaje) de cada uno de los elementos, que son representados de la siguiente forma:

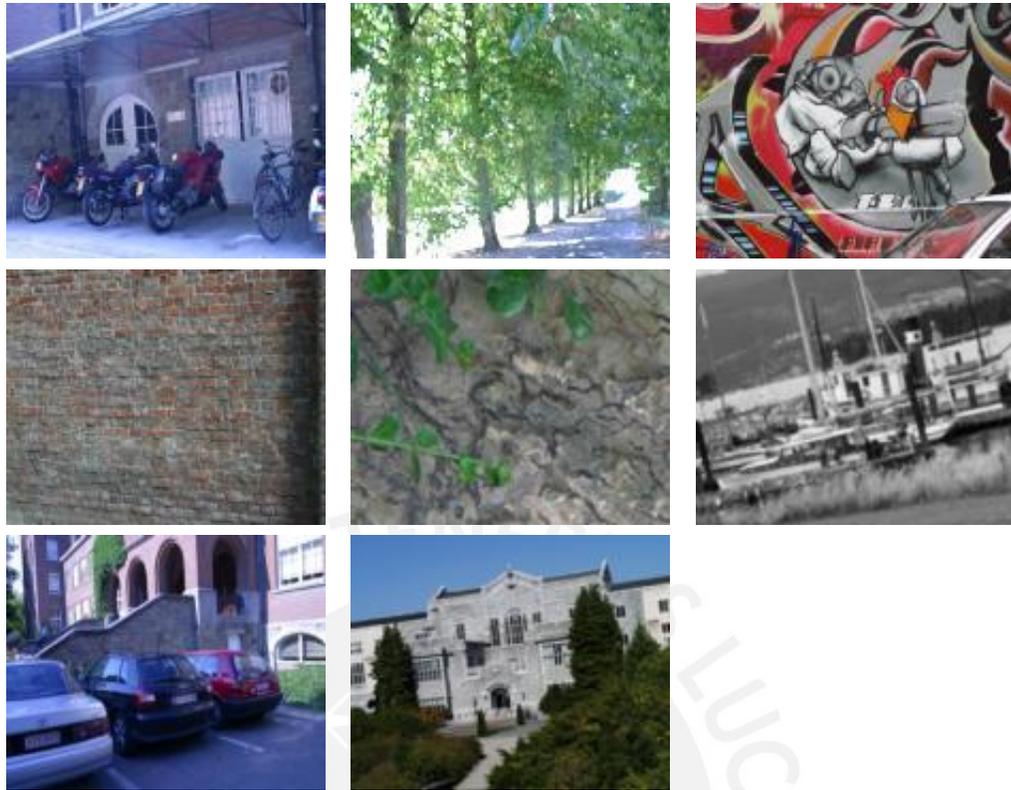


FIGURA 4.4.1. Imágenes: a b c d e f g h . (a), (b) Imágenes de prueba para cambios ante desenfoco. (c), (d) Imágenes de prueba para cambios ante cambios de punto de vista (perspectiva). (e), (f) Imágenes de prueba para cambios ante cambios en acercamiento y rotación. (g) Imagen de prueba para cambios ante iluminación. (h) Imagen de prueba ante cambios en compresión JPEG.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

La notación $i1-i2$ corresponde al análisis entre imágenes 1 y 2, $i1-i3$ a imágenes 1 y 3, etc. Todos los tiempos están en segundos.

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	1908	1351	648	3.32
i1-i3	1908	1175	573	2.86
i1-i4	1908	872	413	2.33
i1-i5	1908	767	643	2.12
i1-i6	1908	469	299	1.92

CUADRO 9. Desempeño del algoritmo ante el desenfoque

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.0015	0.0060	0.0032	0.0059	0.0205
b	0.1364	0.4515	0.3583	0.5274	0.4225
c	0.0429	0.7328	0.1929	0.0354	0.4146
d	0.1449	0.3304	0.0651	0.1528	1.1859
e	0.0014	0.0040	0.0056	0.0002	0.0095
f	0.0249	0.0362	0.0188	0.0043	0.0635
g	0.4782	1.3421	0.1098	86.7185	1.4991
h	0.1038	0.4371	0.6036	0.6550	0.6383
i	0	0	0	0	0

CUADRO 10. Error entre la matriz hallada y la real

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	5681	6027	648	94.18
i1-i3	5681	5508	294	92.91
i1-i4	5681	4830	262	93.95
i1-i5	5681	4416	193	93.29
i1-i6	5681	3037	181	94.11

CUADRO 11. Desempeño del algoritmo ante el desenfoque

4.4.1. Desenfoque. Resultados para las pruebas de desenfoque a la imagen 4.4.1 (a) se observan en las tablas 9 y 10.

Resultados para las pruebas de desenfoque a la imagen 4.4.1 (b) se observan en las tablas 11 y 12.

4.4.2. Cambio en el punto de vista. Resultados para las pruebas de cambios en el punto de vista a la imagen 4.4.1 (c) se observan en las tablas 13 y 14.

Resultados para las pruebas de cambios en el punto de vista a la imagen 4.4.1 (d) se observan en las tablas 15 y 16.

4.4.3. Cambios en acercamiento (Zoom) y rotación. Resultados para las pruebas de cambios en acercamiento (Zoom) y rotación a la imagen 4.4.1 (e) se observan en las tablas 17 y 18.

4.4. ANÁLISIS DE RESULTADOS POR IMÁGENES DE PRUEBA

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.0029	0.0003	0.0374	0.0130	0.0909
b	0.0145	0.0035	0.3084	0.1637	0.6126
c	0.0743	0.1274	0.8597	0.1413	1.7416
d	0.0029	0.0097	0.1724	0.2362	0.6812
e	0.0009	0.0003	0.0275	0.0118	0.0665
f	0.0100	-0.0783	9.3870	0.1441	3.7494
g	0.1442	0.0585	3.3787	5.6227	5.8464
h	1.0000	0.3089	4.2742	6.8443	3.2150
i	0	0	0	0	0

CUADRO 12. Error entre la matriz hallada y la real

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	2358	2598	396	4.75
i1-i3	2358	2761	235	5.25
i1-i4	2358	2596	221	4.67
i1-i5	2358	2942	189	6.01
i1-i6	2358	2563	182	5.32

CUADRO 13. Desempeño del algoritmo ante cambio en el punto de vista

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.0027	0.0095	0.9579	1.0563	6.7189
b	0.0035	0.0398	1.9801	9.0082	0.4039
c	0.0158	0.0089	5.4785	0.3018	0.4052
d	0.0095	0.0814	1.6956	1.3646	5.3623
e	0.0034	0.0234	2.4802	1.7885	1.3234
f	0.0008	-0.0005	1.1056	13.2387	5.7498
g	0.0132	0.2240	0.4520	1.6035	19.6325
h	0.3112	4.6200	314.2826	75.8969	17.0207
i	0	0	0	0	0

CUADRO 14. Error entre la matriz hallada y la real

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	4877	4087	448	56.82
i1-i3	4877	3995	260	54.21
i1-i4	4877	4028	180	55.9
i1-i5	4877	3996	110	56.29
i1-i6	4877	3811	148	56.26

CUADRO 15. Desempeño del algoritmo ante cambio en el punto de vista

Resultados para las pruebas de cambios en acercamiento (Zoom) y rotación a la imagen 4.4.1 (f) se observan en las tablas 19 y 20.

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.0025	0.0074	0.0011	3.0119	5.8778
b	0.0857	0.3371	0.3057	0.0249	0.5529
c	0.0596	0.0690	0.0020	9.4626	7.1802
d	0.0569	0.0178	0.0488	5.2614	0.3666
e	0.0017	0.0074	0.0065	1.0061	0.7180
f	0.0115	0.0125	0.0131	3.5261	1.0276
g	0.0211	0.0052	0.0044	2.8000	2.5320
h	0.1461	5.2826	0.8766	0.0687	794.4035
i	0	0	0	0	0

CUADRO 16. Error entre la matriz hallada y la real

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	1568	1476	123	4.21
i1-i3	1568	1451	97	4.55
i1-i4	1568	1967	107	6.22
i1-i5	1568	1808	80	5.6
i1-i6	1568	1907	67	6.17

CUADRO 17. Desempeño del algoritmo ante cambios en Zoom y rotación

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.0020	0.0029	0.0503	2.9447	11.8221
b	0.0088	-0.0426	0.0475	5.5353	33.9137
c	0.0105	0.0029	0.0028	0.2556	0.3303
d	0.0041	0.0147	0.0324	2.3324	18.2554
e	0.0081	0.0146	0.0703	0.7005	14.0168
f	0.0049	0.0040	0.0010	0.0691	0.2442
g	0.2244	2.6766	2.6523	810.5529	182.4383
h	-0.3929	0.1860	5.0028	370.4485	310.3177
i	0	0	0	0	0

CUADRO 18. Error entre la matriz hallada y la real

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	3957	4078	382	26.71
i1-i3	3957	3139	324	20.5
i1-i4	3957	2050	265	15.31
i1-i5	3957	1789	251	12.04
i1-i6	3957	1628	186	11.86

CUADRO 19. Desempeño del algoritmo ante cambios en Zoom y rotación

4.4.4. Cambio de brillo. Resultados para las pruebas de cambio de brillo a la imagen 4.4.1 (g) se observan en las tablas 21 y 22.¹

¹Para el caso de los valores de los elementos b, c, d, f, g y h de la Tabla 21, se ha calculado el error ejecutando solo la resta de los elementos, debido a que los valores de la matriz original en esos campos era 0, lo cual generaba una división por 0.

4.4. ANÁLISIS DE RESULTADOS POR IMÁGENES DE PRUEBA

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.0069	0.0018	0.0321	0.0841	1.8232
b	0.0075	0.0005	0.0118	0.0387	5.2113
c	0.0992	0.0041	0.0030	0.0155	0.2570
d	0.0043	0.0000	0.0063	0.2713	0.0335
e	0.0060	0.0003	0.0002	-0.0379	4.6704
f	0.0055	0.0001	0.0042	0.0267	0.2668
g	3.4152	0.3911	0.7908	2.9841	9.6408
h	3.3281	0.7088	1.2381	0.5824	56.0269
i	0	0	0	0	0

CUADRO 20. Error entre la matriz hallada y la real

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	2661	2636	1182	15.21
i1-i3	2661	2642	909	15.72
i1-i4	2661	2610	648	16.45
i1-i5	2661	2512	391	15.58
i1-i6	2661	2270	238	13.88

CUADRO 21. Desempeño del algoritmo ante cambios de luz

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.0006	0.0005	0.0007	0.0010	0.0037
b	0.0003	0.0002	0.0019	0.0032	0.0009
c	0.1750	0.0848	0.5582	0.9745	0.1987
d	0.0001	0.0002	0.0003	0.0004	0.0012
e	0.0007	0.0005	0.0015	0.0042	0.0063
f	0.1392	0.1471	0.2749	0.7841	1.3238
g	0	0	0	0	0
h	0	0	0	0	0
i	0	0	0	0	0

CUADRO 22. Error entre la matriz hallada y la real

	Descriptores		Descriptores en común	T. de procesamiento(s)
i1-i2	1961	1510	617	3.66
i1-i3	1961	1242	463	3.15
i1-i4	1961	977	366	2.75
i1-i5	1961	793	315	2.3
i1-i6	1961	633	261	2.01

CUADRO 23. Desempeño del algoritmo ante compresión JPEG

4.4.5. Compresión JPEG. Resultados para las pruebas de compresión JPEG a la imagen 4.4.1 (h) se observan en las tablas 23 y 24.

	i1-i2	i1-i3	i1-i4	i1-i5	i1-i6
a	0.7297	0.7421	0.7366	2.7470	0.7115
b	2.2857	0.8493	0.7715	2.8791	0.1911
c	0.6556	0.6096	0.6525	4.8394	0.9830
d	0.8097	0.6107	0.9120	1.4348	0.6545
e	0.7299	0.7400	0.7350	2.7428	0.7106
f	0.7824	0.7819	0.7349	2.8206	0.7185
g	0.2546	0.3894	0.4014	0.5128	0.8173
h	1.4111	0.8202	0.6092	3.4390	0.3675
i	0.7282	0.7394	0.7349	2.7360	-0.7125

CUADRO 24. Error entre la matriz hallada y la real

4.5. Discusión de resultados

4.5.1. Desenfoque. En el primer caso analizado, se ha tenido un tiempo de procesamiento bajo, además que para los 3 casos se ha tenido un error bajo, y una cantidad elevada de descriptores con correspondencia.

En el segundo caso se ha mantenido un error bajo para los 2 primeros casos, sin embargo en los restantes el error se ha incrementado de tal forma que ya no se ha tenido una correcta función de transformación.

4.5.2. Cambio en el punto de vista. En el primer caso, sólo se tuvo error bajo para las 2 primeras imágenes, para las siguientes se ha tenido un error elevado.

En el segundo caso se ha mantenido un bajo error para las 3 primeras imágenes, que equivalen a los 2 primeros registros.

4.5.3. Cambios en acercamiento (Zoom) y rotación. En caso de Zoom + rotación, se han obtenido pocos descriptores con correspondencia entre las imágenes. Sin embargo, para el primer grupo de imágenes se obtuvo un error bajo para las dos primeras funciones de transformación, luego el error aumenta, para el segundo grupo, se obtuvo error bajo para todas las transformaciones sin contar la última, que el error es alto.

4.5.4. Cambio de iluminación. Para este caso, conforme se oscurece la imagen, se obtienen menos descriptores y además en todos los casos se ha obtenido un error elevado.

4.5.5. Compresión JPEG. En el caso de la compresión JPEG se ha obtenido un error bajo en todas las imágenes analizadas, aun así cuando la calidad de la imagen debido a la compresión ha bajado considerablemente desde un punto de vista subjetivo, la cantidad de descriptores encontrados ha disminuido.

4.6. Análisis de resultados para imágenes de campo de cultivo usando transformaciones geométricas

En esta sección se analizará el desempeño del algoritmo propuesto ante transformaciones geométricas fijas (especialmente rotación y translación), del mismo modo que en la sección anterior. Sin embargo en este caso se usará imágenes propias del campo de cultivo modificadas para ser analizadas, es decir, se elegirá una imagen representativa y se modificará con el objetivo de medir los límites de cada transformación geométrica.

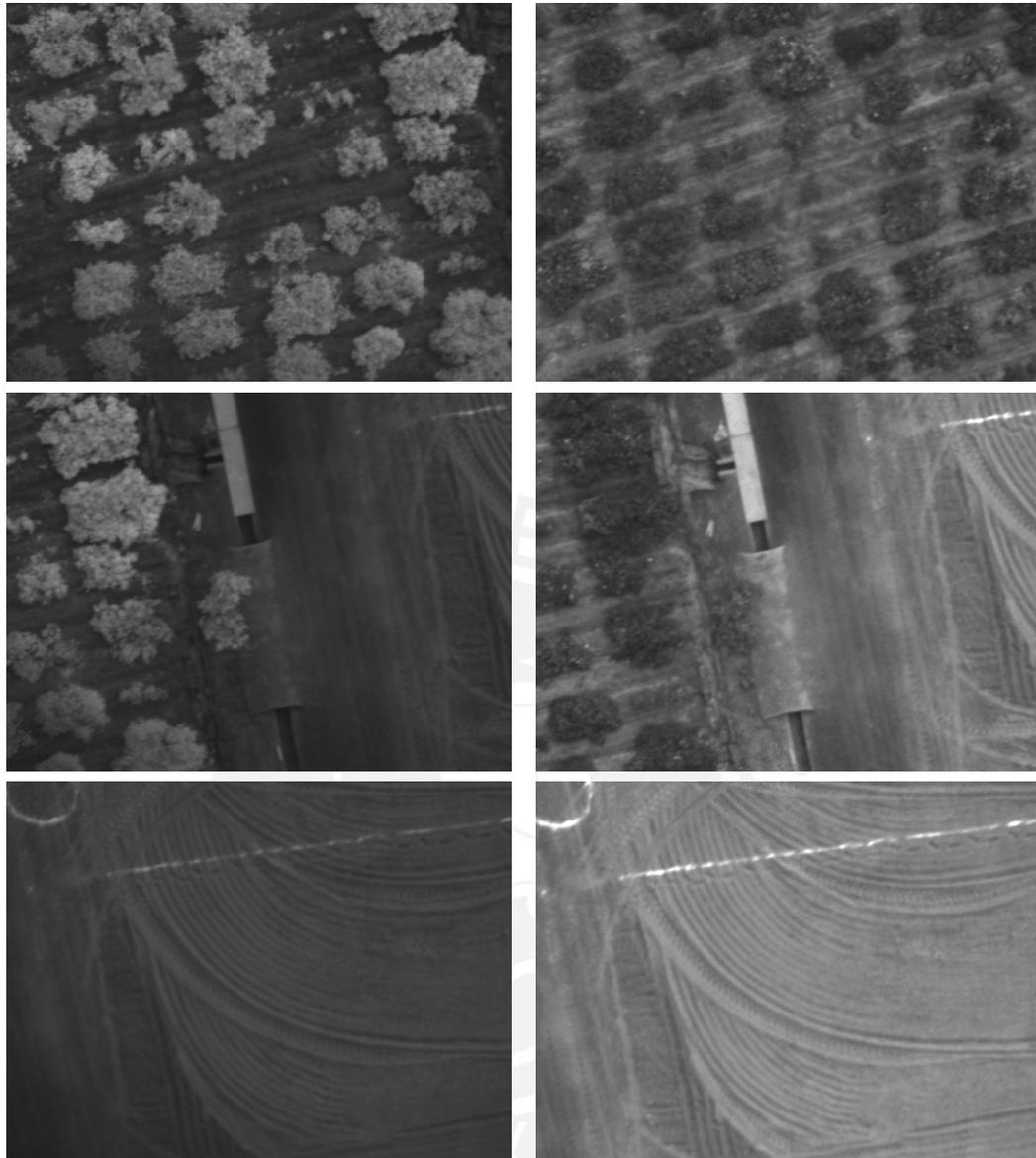
El procesamiento de las imágenes mostradas es el objetivo del desarrollo del software expuesto en el presente documento.

El desempeño del algoritmo fue probado para transformaciones geométricas de rotación en los 3 ejes (siendo eje z perpendicular al plano de la imagen) y además para translación en los ejes x e y.

El algoritmo no tuvo ningún problema al trabajar con imágenes que habían sido trasladadas en los ejes x e y. Tampoco tuvo problemas al probarse con imágenes rotadas en el mismo plano de la imagen. Sin embargo, el algoritmo mostró deficiencias al ser probado para imágenes rotadas en sus ejes paralelos (x e y).

En las figuras 4.6.2 y 4.6.3 se muestran las gráficas *error vs. ángulo* y *cantidad de descriptores vs. ángulo* de las imágenes de la figura 4.6.1 rotadas con respecto a los ejes x e y.

Como se puede observar, el algoritmo soporta transformaciones de rotación aceptable (mayores de 40 grados) para imágenes que muestran una amplia zona de vegetación. Sin embargo, el error aumenta fuera del área de vegetación (donde solo se observa el suelo). Esto se debe a que la vegetación presente en la imagen proporciona un alto contraste con el suelo, facilitando así la formación del registro de imágenes.



a
FIGURA 4.6.1. Filas *b* . Imágenes en infrarrojo y rojo visible que
c
muestran (a) zona de vegetación densa, (b) solo una porción de la
imagen con vegetación y (c) zona donde no hay vegetación.

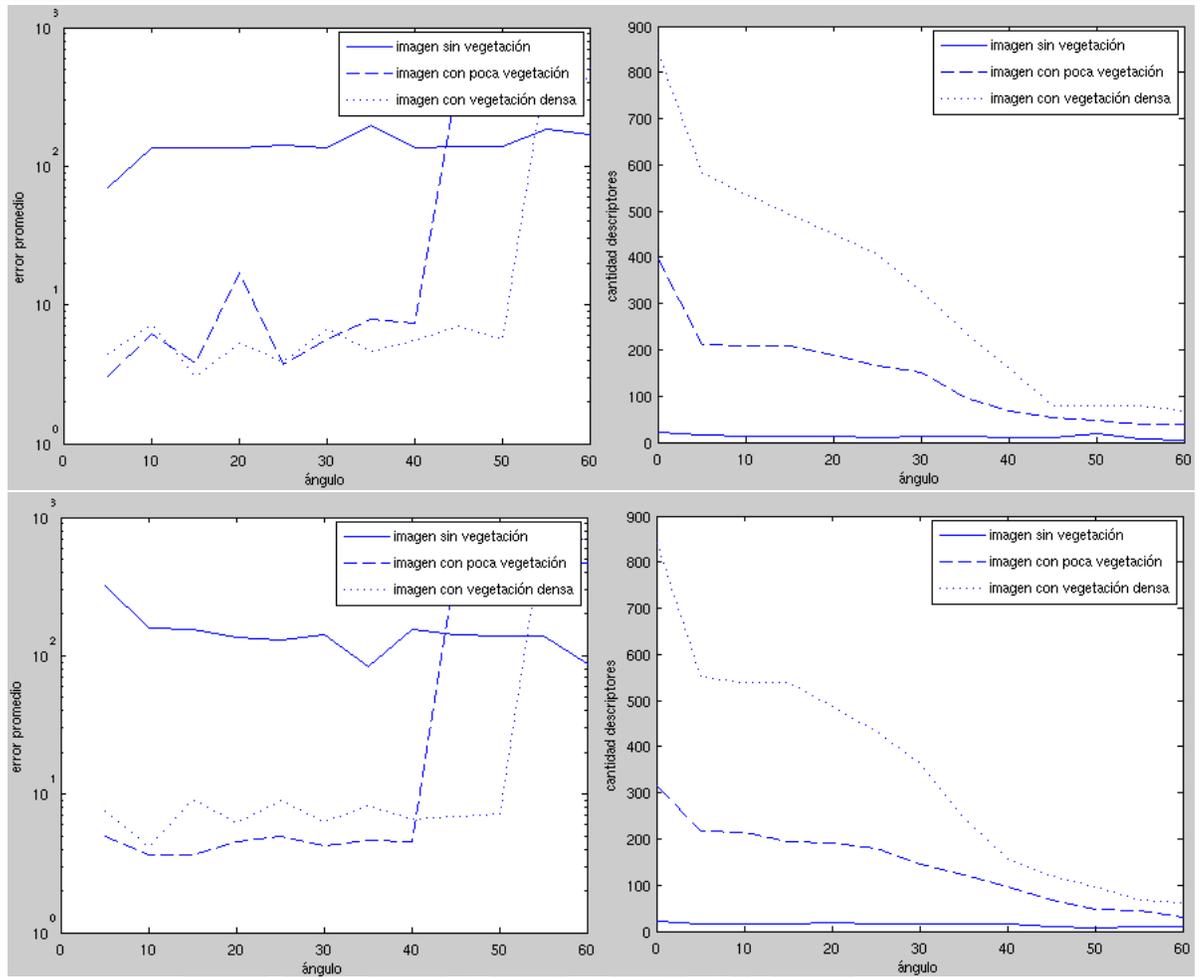


FIGURA 4.6.2. Gráficas de error vs. ángulo (izquierda) y cantidad de descriptores vs. ángulo (derecha) para las imágenes en infrarrojo. Arriba: rotación respecto al eje x. Abajo: rotación respecto al eje y.

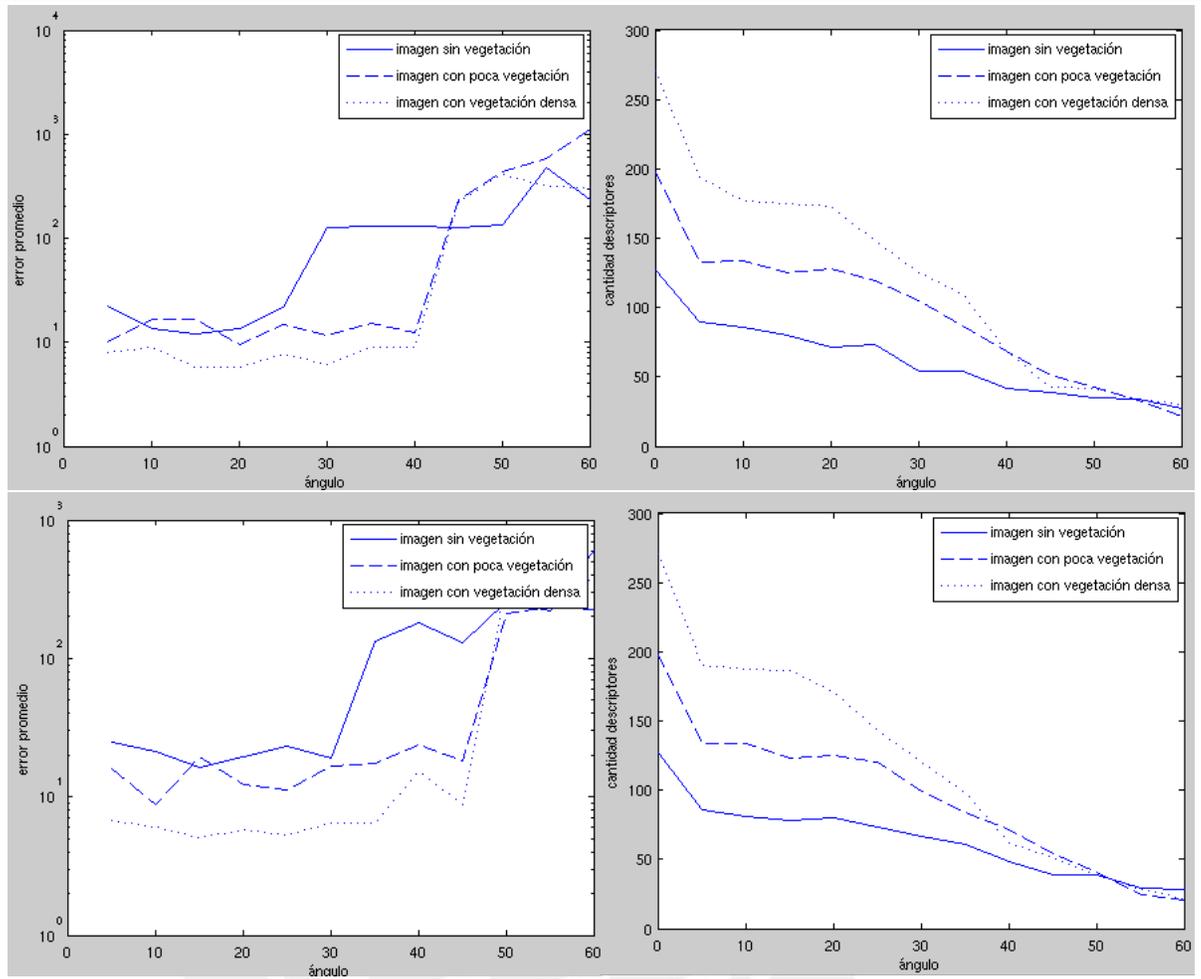


FIGURA 4.6.3. Gráficas de error vs. ángulo (izquierda) y cantidad de descriptores vs. ángulo (derecha) para las imágenes en rojo visible. Arriba: rotación respecto al eje x. Abajo: rotación respecto al eje y.

Conclusiones

El algoritmo SURF recientemente publicado, es una alternativa eficiente para esta clase de imágenes que han sido tomadas desde un aeromodelo y que presentan un área en común considerable (mayor al 50 %).

Se ha mejorado considerablemente el tiempo de procesamiento con respecto al diseño hecho en [15], el cual llegaba a ser tedioso su uso, e incluso ineficiente, esto debido al cambio del algoritmo por uno más veloz y además al estar programado por un lenguaje de programación de menor nivel, como es el caso del Lenguaje C.

El uso de la biblioteca de visión por computadora OpenCV ha sido sumamente útil ya que posee funciones para muchas aplicaciones.

En el caso de imágenes en espectro rojo visible, en el Capítulo 4 se observó que en varios casos falló ya que no se llegó a hallar una función de transformación por la poca cantidad de descriptores. Esto no ocurrió en el caso de las imágenes en espectro infrarrojo, que fueron registradas al 97 % (194 de 200 imágenes) ya que los campos de cultivo reflejaban la luz en esta banda del espectro.

El algoritmo utilizado ha demostrado ser robusto a transformaciones geométricas de translación, sin embargo soportó en menor medida otras transformaciones geométricas (proyectivas). Específicamente se analizó para transformaciones de rotación, obteniéndose ángulos de rotación máximos entre 45 y 55 grados en imágenes con alto contraste (vegetación densa). Este resultado disminuye cuando hay menos vegetación en la imagen (menor contraste).

La transformación geométrica dominante entre cada par de imágenes es la translación, ya que modela el desplazamiento de aeromodelo por el campo de cultivo. El resto de transformaciones aparecen en menor medida y pueden ser soportadas por el algoritmo, siempre y cuando existan objetos que generen contraste con el suelo.

El algoritmo fue implementado para generar registro de imágenes en campos de cultivo. Dichos campos de cultivo generarán el contraste necesario con el suelo, el cual será aprovechado por el algoritmo para la obtención de descriptores y con ellos la generación de registro de imágenes, debido a ésto el algoritmo cumple con los objetivos requeridos.

Recomendaciones

Como ya se mencionó antes, en los campos de cultivo, hay imágenes en que se obtiene una cantidad baja de descriptores, en estos casos es recomendable colocar “marcas”, las cuales son objetos que reflejan la luz en los espectros infrarrojo y rojo visible.

Conforme lo analizado en la sección 4.6, para las imágenes en espectro infrarrojo se ha obtenido una alta tasa de registros con error bajo, en el caso que se trabajó con NNDR de 0.6, la tasa fue de 97 % de acierto (con el supuesto que las imágenes con error mayor a 50 no están correctamente registradas), y para el caso de imágenes en espectro rojo visible, la tasa de aciertos fue menor (aproximadamente del 87 %) por lo que este valor es con el que se han obtenido mejores resultados experimentales.

Para un correcto desempeño del algoritmo se recomienda un traslape mayor a 50 %, aunque se ha visto que ha funcionado incluso hasta con 40 % de traslape. Sin embargo las imágenes obtenidas de forma nominal en el aeromodelo del CIP se obtuvo un traslape entre el 80 y 90 % debido al escaso tiempo entre imágenes (10 ms) por lo que el algoritmo en nuestro caso se ha desempeñado de forma correcta.

Según las pruebas hechas en la sección 4.6 el algoritmo funciona correctamente en zonas de vegetación densa, sin embargo presenta problemas en zonas de escasa vegetación debido a que no hay objetos que generen contraste con el suelo. Este problema puede ser resuelto colocando las “marcas” mencionadas anteriormente en zonas de poca vegetación, ya que generan contraste con el suelo necesario para el cálculo de descriptores.

Bibliografía

- [1] GODDARD SPACE FLIGHT CENTER.
- [2] Ministerio de Agricultura. Producción de papa.
- [3] J. Alvarez et al. *Alimentar al mundo: Un reto del siglo XXI*. Siglo XXI, 2003.
- [4] C. Evans. Notes on the open surflibrary. 2009.
- [5] R. Gonzales. *Digital Image Processing (2nd Edition)*. Addison - Wesley, 1992.
- [6] T. Tuytelaars y L. Van Gool H. Bay. Speed-up robust features. *In 9th European Conference on Computer Vision*, 2007.
- [7] L. Joyanes. *Programación en C*. McGraw Hill, 2001.
- [8] K. Laudon. *Sistemas de Información Gerencial*. Pearson Education, 2004.
- [9] D. Lowe. Object recognition from local scale-invariant features. *IEEE International Conference on In Computer Vision*, pages 1150–1157, 1999.
- [10] V. Torres Lozada. Agricultura peruana en los tiempos del tlc. *Documento de trabajo para el X congreso nacional*, 2004.
- [11] K. Mikolajczyk. Affine covariant features. 2007.
- [12] CENTRO INTERNACIONAL DE LA PAPA.
- [13] D. Parent. Computer animation: algorithms and techniques. 2007.
- [14] H. Rodriguez. *Imagen Digital: conceptos básicos*. Marcombo, 2005.
- [15] R. Rojas. *Diseño de un sistema de registro de imágenes orientado a la agricultura de precisión*. PUCP, 2009.
- [16] R. Schowengert. *Remote sensing and methods for image processing (2nd Edition)*. Academic Press, 1997.
- [17] O. Vargas. ¿qué es el alca? (globalización, estados unidos y américa latina). 2002.
- [18] G. Bradsky y A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.
- [19] K. Mikolajczyk y C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.
- [20] B. Zitova y J. Flusser. Image registration methods: a survey. *Image and Vision Computing 21 (2003) 977-1000*, 2003.
- [21] C. A. Glasbey y K. V. Mardia. A review of image methods. *Journal of applied statistics*, Vol.:25:155–177, 1998.
- [22] R. Ortega y L. Flores. Agricultura de precisión: Introducción al manejo sitio-específico.
- [23] D. Cerry y P. Graciano. Monitoreo del rendimiento de caña de azúcar. 2005.
- [24] T. Lilesand y R. Kiefer. *Remote sensing and image interpretation*. Wiley, 2008.
- [25] K. Duffin y W. Barrett. Globally optimal image mosaics. *Graphics Interface*, 2009.