

Filtros y técnicas de clasificación en imágenes de teledetección.

Por Mariano Della Vedova

Trabajo Especial de Licenciatura en Ciencias de la Computación



Facultad de Matemática, Astronomía y Física
Universidad Nacional de Córdoba
Argentina

Septiembre, 2013

DIRECTOR
Dr. Oscar H. Bustos
FAMAF/UNC. Córdoba, Argentina.

Una de las tareas más importantes de la teledetección, si no es la más importante, es la clasificación de imágenes satelitales. Teniendo en cuenta que los ruidos siempre están presentes en estas imágenes, es esencial analizar las mejores formas de preparar una imagen para obtener como resultado la mejor clasificación posible.

En este trabajo se generó un ambiente de trabajo que facilita la generación de distintos ruidos y el uso de filtros, teniendo en cuenta la necesidad de realizar estudios que involucren estas funcionalidades. También desarrolló un método para comparar clasificaciones, con el objetivo de poder decidir el grado de satisfacción al aplicar cada uno de estos métodos.

Para el desarrollo de este trabajo se tomaron como base los Estándares de Ingeniería del Software de la Agencia Espacial Europea con el fin de aplicarlos en la implementación de las herramientas mencionadas. Se llevaron a cabo las distintas fases definidas por los estándares mencionados, generando la documentación requerida para cada una de ellas. Lo cual permite tener un grado de confianza mayor sobre el software desarrollado.

Palabras claves: teledetección, filtros, ruido, clasificación, posclasificación, ingeniería del software, ESA, requerimientos, arquitectura de software, IDL, ENVI.

AGRADECIMIENTOS

Muchos hechos, sensaciones y personas aparecieron y desaparecieron en mi vida desde que tendría que haber empezado a realizar este trabajo y que empecé realmente. Entre esas sensaciones necesitaba una motivación y ésta solo surgió gracias a una persona. Si no fuera por Gise, este trabajo y muchas otras cosas nunca las hubiese hecho. Por eso el agradecimiento a ella es especial.

A mis padres, Quique y Maggi, que siempre confiaron, me apoyaron y esperan tanto de mi. A mis suegros, Mari y José, que tuvieron que aceptar que invadiera un espacio para utilizarlo para mi estudio y lo tomaron siempre con respeto y sobre todo paciencia.

También quiero agradecer a una persona que ya no veo tanto, pero que le guardo un cariño enorme. A quien ha sido mi guía espiritual mucho tiempo y que me ha dejado enseñanzas que me formaron como persona. Gracias madrina.

Agradezco también a quienes han hecho que todos estos años sean tan agradables. Mis amigos que siempre han estado al lado mío a pesar de mis errores y mis asiertos, mis dos “bandas”, “la banda del R12” y “la banda del tío”. Y a las personas que hacen que cada día en mi trabajo sea especial y placentero, los integrantes del “Call”.

A mi director, Oscar, mi más sincero agradecimiento por guiarme en esta etapa de mi carrera. También a Marcelo, Daniel y Sergio, que han invertido parte de su tiempo en el trabajo especial que había empezado, y yo no pude aprovecharlo como hubiese querido.

TABLA DE CONTENIDOS

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura del trabajo	3
2. Conocimientos básicos sobre la teledetección espacial	4
2.1. Fundamentos de la teledetección	4
2.1.1. El espectro electromagnético	6
2.2. Representación de imágenes digitales	7
2.2.1. Vecinos de un píxel	9
2.3. Sensores	10
2.3.1. Resolución de un sistema sensor	10
2.3.2. Sensores pasivos	12
3. Correcciones de ruido en imágenes	15
3.1. Ruido en la imagen	15
3.1.1. Fuentes de error en una imagen espacial	16
3.1.2. Modelado del ruido	18
3.2. Filtros espaciales	19
3.2.1. Bases del filtrado espacial	20
3.2.2. Filtros espaciales lineales	22

3.2.3. Filtros espaciales de orden	24
3.2.4. Filtros adaptativos	26
4. Clasificación	29
4.1. Clasificación supervisada	30
4.1.1. Fase de entrenamiento	31
4.1.2. Fase de asignación	31
4.2. Clasificación no supervisada	36
4.2.1. Clasificación ISODATA	36
4.3. Verificación de resultados	38
5. Resultados	41
5.1. Ruidos	41
5.2. Filtros lineales	44
5.3. Filtros de orden	50
5.4. Filtros adaptativos	57
5.5. Posclasificación	59
6. Conclusiones	60
6.1. Trabajo a futuro	60
A. Documento de Requerimientos de Usuario	62
B. Documento de Especificación de Software	69
C. Documento de Diseño Detallado	87
D. Manual de Usuario del Software	98
E. Plan de Verificación y Validación del Software	110
Bibliografía	129

ÍNDICE DE FIGURAS

2.1. Esquema de una onda electromagnética	5
2.2. Espectro electromagnético	6
2.3. Representación de una imagen multiespectral.	9
2.4. Representación de un explorador de barrido.	13
2.5. Representación de un explorador de empuje.	14
3.1. Un modelo del proceso de degradación/restauración de una figura	16
3.2. Una ventana de un filtro espacial. El primer píxel de salida es calculado (izquierda), el siguiente píxel del salida es calculado para la misma fila (centro), y luego de que la fila es completada el proceso se repite para la siguiente fila (derecha). El píxel de salida es ubicado en la coordenada del píxel sombreado en la imagen de salida.	20
3.3. El problema de la región del borde de una imagen para un filtro.	21
4.1. Clasificador de paralelepípedos. Un ejemplo de un conjunto de paralelepípedos bidimensionales.	35
4.2. Clasificador de paralelepípedos. Posibles problemas.	35
5.1. Comparación entre imagen original e imagen con ruido gaussiano	42
5.2. Comparación entre imagen original e imagen con ruido uniforme	43
5.3. Comparación entre imagen original e imagen con ruido uniforme	44
5.4. Comparación entre imagen original, imagen con ruido uniforme e imagen con ruido uniforme con filtro de media aritmética aplicado	45
5.5. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de media geométrica aplicado	46

5.6. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de media armónica aplicado	47
5.7. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de media contrarmónica aplicado	48
5.8. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro gaussiano aplicado	49
5.9. Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de mediana aplicado	50
5.10. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de mínima aplicado	51
5.11. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de máxima aplicado	52
5.12. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro olímpico aplicado	53
5.13. Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de media alfa recortada aplicado	54
5.14. Comparación entre imagen original, imagen con ruido uniforme e imagen con ruido uniforme con filtro de media de los k valores más cercanos aplicado	55
5.15. Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de mediana de los k valores más cercanos aplicado	56
5.16. Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de reducción de ruido local aplicado	57
5.17. Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de mediana adaptativo aplicado	58
5.18. Matriz de confusión y coeficiente de Kappa entre dos clasificaciones	59

LISTA DE ACRÓNIMOS

CCD Charge Couple Devices

CONAE Comisión Nacional de Actividades Espaciales

ESA Agencia Espacial Europea

ν frecuencia

Hz Herz

IFOV Instantaneous Field Of View

IRC Infrarrojo Cercano

IRM Infrarrojo Medio

IRT Infrarrojo lejano o Térmico

Js julios

κ coeficiente kappa

λ longitud de onda

lm lumens

M Microondas

MWIR Infrarrojo de onda media

ND Nivel Digital

NV Nivel Visual

SWIR Infrarrojo de onda corta

eV electron-volts

W watts

1.1. Motivación

La teledetección espacial ha tenido un gran avance en las últimas décadas. Tanto es así que las imágenes y datos adquiridos mediante la teledetección son una fuente de información cada vez más importante en el estudio de la Tierra. Su utilización nos permite realizar el seguimiento de eventos que ocurren en nuestro planeta, eventos que se llevan a cabo a lo largo de los años tales como desertificación, deforestación, crecimiento de manchas urbanas; como así también eventos que ocurren en un corto período de tiempo como incendios forestales, desbordes de ríos, etc. Esto obliga a una actualización dinámica del software que permite el tratamiento de los datos obtenidos de este modo.

Si se tiene en cuenta concretamente las imágenes de teledetección, se utilizan procesamientos orientados a transformarlas por medio de una ecuación o una serie de ecuaciones (con la computadora) con la idea obtener una nueva imagen digital y sacar el mayor provecho para obtener la información contenida en la misma.

Toda imagen de teledetección presenta perturbaciones no deseadas, que son debidas a diversas causas provenientes de deficiencias en los procesos de percepción, digitalización de señales y/o registro de datos. El ruido puede o bien degradar o bien ocultar totalmente el contenido de información radiométrica de una imagen digital. Con el fin de remover el ruido en la mayor medida posible se usan procedimientos matemáticos llamados “filtros” que se aplican sobre la imagen con ruido. En general, consisten en definir una ventana de tamaño 3x3, 5x5 ó 7x7 de píxeles que se van desplazando sobre la imagen y sobre la cual se realizan ciertas operaciones o transformaciones cuyas resultantes se usan para remplazar el valor del píxel central. Debe tenerse en cuenta que esta transformación no sólo puede reducir el nivel de “ruido” presente en la imagen, sino que altera (o puede alterar) los valores legítimos de la misma. Existen algunos filtros, relativamente recientes, que tienen en cuenta esta situación y tratan de evitar distorsiones significativas en tales valores legítimos.

Por otro lado, una de las tareas más importantes en el procesamiento y análisis de imágenes es clasificar cada píxel como perteneciente a una cierta categoría de terrenos también llamada “tema”; por ejemplo clasificar un píxel como perteneciente a la clase “agua”, “trigo”, “ciudad”, “camino”, etc. La clasificación digital usa información espectral representada por los valores digitales en una o más bandas, para clasificar cada píxel individual basado en esta información espectral. El objetivo es asignar todos los píxeles de la imagen a clases o temas particulares. La imagen clasificada que resulta, es un mosaico de píxeles cada uno de los cuales pertenece a un tema particular y es esencialmente un

mapa temático de la imagen original.

Por todo lo expuesto anteriormente, la remoción del ruido habitualmente debe preceder a cualquier técnica de mejoramiento o clasificación de una imagen. El objetivo en la remoción de ruidos es restaurar una imagen para obtener una lo más aproximada posible a la escena real focalizada en la imagen. Una vez aplicado un filtro a la imagen observada se procede a clasificar los distintos píxeles que integran la imagen.

Aunque también surge la necesidad de analizar el rendimiento de las diversas técnicas de clasificación que se usan habitualmente en el análisis estadístico de imágenes de teledetección, frente a la aplicación de diversos filtros de ruido. Dado que todo filtro involucra el riesgo de perder información legítima, es relevante preguntarse también qué sucede con la clasificación si se prescinde del filtrado previo.

La importancia que presentan los filtros y las técnicas de clasificación en la tarea de aprovechar al máximo la información que proveen las imágenes de teledetección motivó la realización de este trabajo.

1.2. Objetivos

Este trabajo tiene como objetivo contribuir al análisis de filtros y técnicas de clasificación en imágenes de teledetección. Por lo que se propone generar un ambiente para la realización de una comparación entre diversos métodos de clasificación de imágenes en combinación con filtros aplicados previamente para remover distorsiones en las imágenes de teledetección tal como llegan a los usuarios. Así como también se propone realizar el análisis apelando a simulación de modelos ya consagrados en la literatura sobre imágenes de teledetección.

En este trabajo se plantea disponer de las herramientas para realizar una comparación entre los métodos habituales de clasificación con y sin filtrado previo. Los filtros que se proponen usar en este trabajo serán también los de uso habitual. Se quiere así realizar una primera aproximación al tema de diseño de nuevos filtros y/o técnicas de clasificación que sean más confiables (lo que está fuera del alcance de este proyecto).

El resultado de este trabajo tiene como objetivo motivar la realización de un estudio de simulación por medio del cual se obtenga un criterio de qué procedimientos son adecuados para obtener resultados de clasificación en presencia de imágenes con ruido, que sean confiables respecto de los criterios usados para analizar los resultados de una clasificación: análisis de la matriz de confusión y coeficiente Kappa.

Así mismo, en este trabajo se propone respetar buenas prácticas definidas por la ingeniería del software para obtener un software de calidad, teniendo como objetivo atributos como: correctitud, fiabilidad, robustez, performance, usabilidad, verificabilidad, facilidad para el mantenimiento, adaptación a cambios, reusabilidad y portabilidad. Para lograr sistemas de software que cumplan con estos atributos es necesario pasar por una serie de etapas bien definidas que determinan su ciclo de vida. Cada una de estas etapas o fases tiene definido su comienzo y final de una manera precisa, con entregables hacia la siguiente etapa. Las etapas de un modelo de ciclo de vida comprenden:

- análisis y especificación de los requerimientos
- diseño y especificación del sistema
- codificación y testeo de los módulos

- integración y testeo del sistema
- entrega del producto de software y mantenimiento del mismo

En este trabajo se utilizan los estándares de ingeniería del software definidos y utilizados por la Agencia Espacial Europea (ESA). Esto permitirá desarrollar herramientas con los atributos antes mencionados, utilizando estándares internacionales sobre los que están basados los desarrollos de la Comisión Nacional de Actividades Espaciales (CONAE).

1.3. Estructura del trabajo

Los temas desarrollados en los próximos capítulos incluirán:

Capítulo 2. Se ven conceptos básicos de la teledetección espacial y se describen distintos tipos de sensores. También se explica cómo se representan las imágenes digitales. En resumen, este capítulo contiene una descripción general de los conceptos que se utilizarán a lo largo de este trabajo.

Capítulo 3. En este capítulo se muestran los diferentes tipos de ruidos que se pueden encontrar en imágenes de teledetección espacial y la fuente que puede producirlos. También se desarrolla la definición de distintos filtros que pueden utilizarse para eliminar estos ruidos.

Capítulo 4. Descripción sobre diferentes tipos de clasificación. Este capítulo contempla todas las etapas de la clasificación tanto supervisada como no supervisada. Por último, contiene una propuesta para la comparación entre clasificaciones.

Capítulo 5. Se muestran los resultados obtenidos luego de cumplir todas las etapas de desarrollo.

Capítulo 6. Se presentan las conclusiones de este trabajo, así como también un breve análisis del posible trabajo a futuro.

CAPÍTULO 2

CONOCIMIENTOS BÁSICOS SOBRE LA TELEDETECCIÓN ESPACIAL

La *teleobservación* es el conjunto de técnicas que permite obtener información acerca de un objeto, área o fenómeno a través del análisis de datos adquiridos por un dispositivo que no está en contacto con el objeto, área o fenómeno bajo estudio [Rabolli y Gulich, 2005]. La *teledetección* se refiere en un sentido general a la instrumentación, técnicas y métodos usados para observar la superficie de la Tierra, usualmente mediante la formación de una imagen en una posición, estática o móvil, a una cierta distancia remota de esa superficie [Richards y Jia, 2006]. Dicho de otra forma, la teledetección es la medición de propiedades de objetos sobre la superficie terrestre usando datos adquiridos desde un avión o un satélite [Schowengerdt, 2006].

En este capítulo se muestran diferentes conceptos que se relacionan fuertemente con la teledetección y que serán utilizados a lo largo de este trabajo. La sección 2.1 presenta algunos aspectos físicos que se deben tener en cuenta en teledetección. Por otro lado, la sección 2.2 define como son representadas las imágenes digitales. Por último, en la sección 2.3 se describen nociones básicas sobre algunos tipos de sensores que se utilizan en teledetección.

2.1. Fundamentos de la teledetección

La *teledetección* es la técnica que nos permite obtener información a distancia de los objetos situados sobre la superficie terrestre. Para que esta observación remota sea posible, es preciso que entre los objetos y el sensor exista algún tipo de interacción. Los tres principales elementos de cualquier sistema de teledetección son: sensor, objeto observado y flujo energético que los pone en relación y permite, en última instancia, detectar el objeto. Las tres formas de adquirir información a partir de un sensor remoto son: por reflexión, por emisión y por emisión-reflexión [Chuvieco, 2010]. Este trabajo está orientado a imágenes adquiridas por reflexión o por emisión, por lo que se pondrá especial atención a estas dos formas de obtener información.

El *flujo energético* entre la cubierta terrestre y el sensor constituye una forma de *radiación* conteniendo dos campos de fuerza ortogonales entre sí; *eléctrico* y *magnético* (como muestra la figura 2.1). Las características de este flujo energético pueden describirse por dos elementos: *longitud de onda* (λ) y *frecuencia* (ν). La primera hace referencia a la distancia entre dos picos sucesivos de una onda, mientras que la frecuencia designa el número de ciclos pasando por un punto fijo en una unidad

de tiempo. Ambos elementos están inversamente relacionados:

$$\lambda = \frac{c}{\nu} \quad (2.1)$$

donde c indica la velocidad de la luz ($2,998 \times 10^8 \text{ m s}^{-1}$). En definitiva, a mayor longitud de onda, menor frecuencia y viceversa.

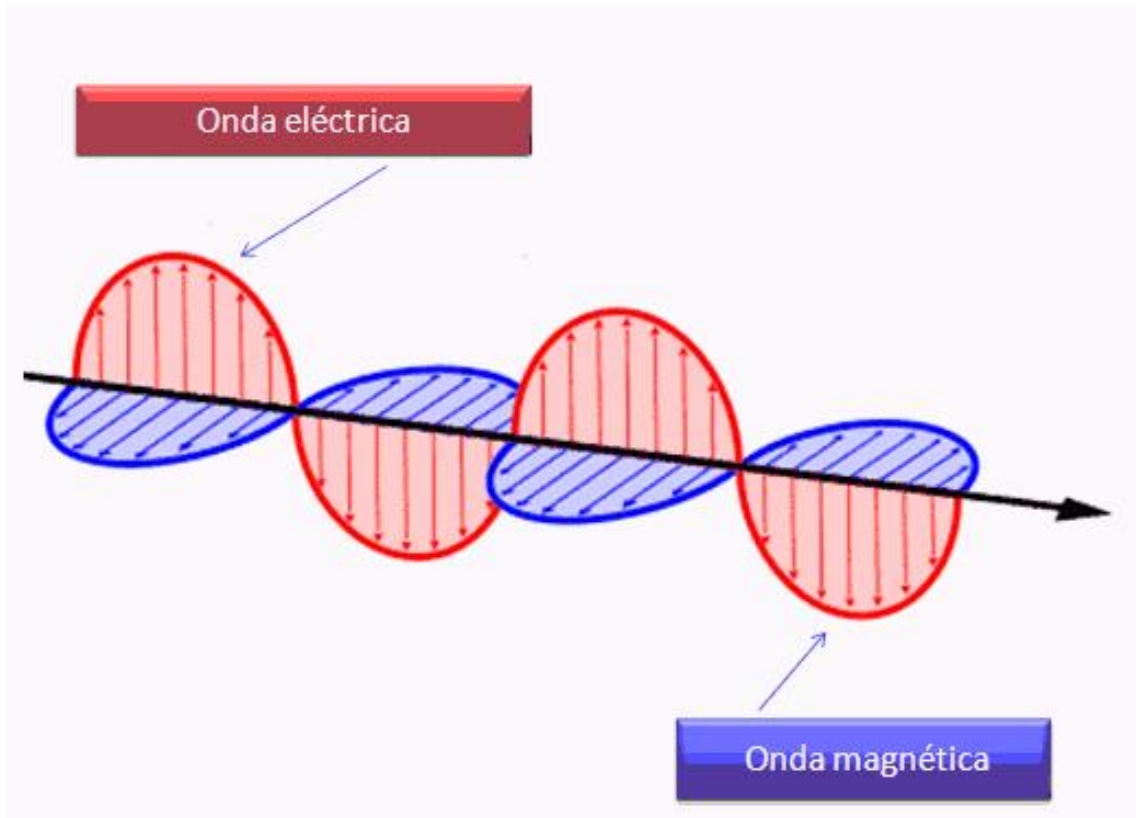


Figura 2.1: Esquema de una onda electromagnética

Las *ondas electromagnéticas* pueden ser visualizadas como propagaciones de ondas sinusoidales con longitud de onda λ (figura 2.1), o pueden ser pensadas como una cadena de partículas sin masa, cada una viajando con un patrón en forma de onda y moviéndose a la velocidad de la luz. Cada partícula sin masa contiene una cierta cantidad de energía. Cada porción de energía es llamado *fotón*. Podemos calcular la cantidad de energía transportada por un fotón, siempre que se conozca su frecuencia:

$$Q = h\nu \quad (2.2)$$

donde Q es la energía radiante de un fotón y h la constante de Planck ($6,626 \times 10^{-34} \text{ Js}$).

Sustituyendo la ecuación 2.1, podemos expresar:

$$Q = h(c/\lambda) \quad (2.3)$$

lo que significa en definitiva que a mayor longitud de onda (o menor frecuencia) el contenido energético será menor y viceversa.

Las unidades de longitud de onda es el metro, con los términos micrómetros o micrones (denotado μm e igual a 10^{-6} m) y nanómetros (10^{-9} m) siendo los más usados [Chuvienco, 2010; Gonzalez y Woods, 2002; Rabolli y Gulich, 2005]. La frecuencia se mide en Herz (Hz) [Gonzalez y Woods, 2002], con un Herz siendo igual a un ciclo de una onda sinusoidal por segundo. Una unidad comunmente

usada para la energía son los electron-volts (eV) [Gonzalez y Woods, 2002] o los julios (Js) [Chuvieco, 2010].

La luz que no tiene color es llamada luz acromática o monocromática. El único atributo de esta luz es su intensidad, o cantidad. El término *nivel de gris* generalmente es usado para describir intensidad monocromática porque está en un rango desde el negro al blanco. Tres cantidades básicas son usadas para describir la calidad de una fuente de luz cromática: *radiancia*; *luminosidad*; y *brillo*. Radiancia es la cantidad total de energía que fluye desde la fuente de luz, y es usualmente medida en watts (W). La luminosidad, medida en lumens (lm), mide la cantidad de energía que un observador percibe de una fuente de luz. Finalmente, el brillo es un descriptor subjetivo de percepción de luz que es prácticamente imposible de medir [Gonzalez y Woods, 2002].

2.1.1. El espectro electromagnético

De las fórmulas anteriores, se deduce que podemos definir cualquier tipo de energía radiante en función de su longitud de onda o frecuencia. Aunque la sucesión de valores de longitud de onda es continua, suelen establecerse una serie de bandas en donde la radiación electromagnética manifiesta un comportamiento similar. La organización de estas bandas de longitud de onda o frecuencia se denomina *espectro electromagnético* y se muestran en la figura 2.2. Comprende desde las longitudes de onda más cortas (rayos gamma, rayos X) hasta las kilométricas (telecomunicaciones -ondas de radio-).

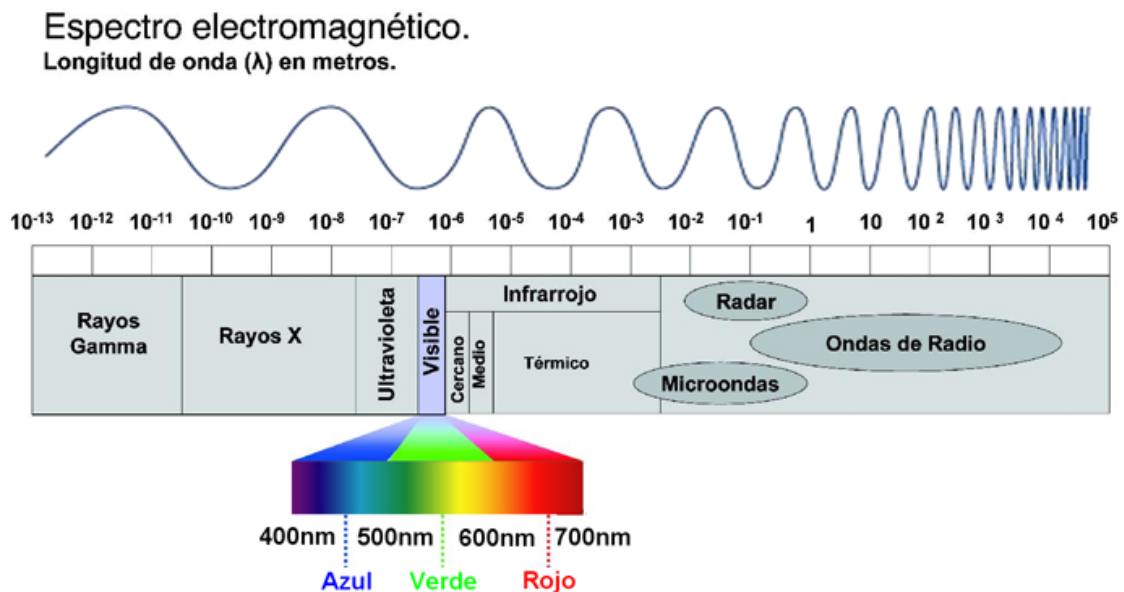


Figura 2.2: Espectro electromagnético

Desde el punto de vista de la teledetección, conviene destacar una serie de bandas espectrales, que son las más frecuentemente empleadas con la tecnología actual:

Espectro visible: (0,4 a 0,7 μm). Se denomina así por tratarse de la única radiación electromagnética que pueden percibir nuestros ojos, coincidiendo con las longitudes de onda en donde es máxima la radiación solar. Dentro de esta región, suelen distinguirse tres bandas elementales, que se denominan azul (A: 0,4 – 0,5 μm), verde (V: 0,5 – 0,6 μm), y rojo (R: 0,6 – 0,7 μm), en razón a los colores primarios que nuestros ojos perciben a esas longitudes de onda.

Infrarrojo Cercano (IRC): (0,7 a 1,2 μm). También se denomina infrarrojo próximo, reflejado o fotográfico.

Infrarrojo medio: (1,2 a 8 μm). En esta región se entremezclan los procesos de reflexión de la luz solar y de emisión de la superficie terrestre. En el primer caso, hablamos de infrarrojo de onda corta (ShortWave InfraRed, SWIR), que se sitúa entre 1,2 y 2,5 μm . La segunda banda de interés está comprendida entre 2,5 y 8 μm y suele denominarse Infrarrojo Medio (IRM) o infrarrojo de onda media (MidWare InfraRed, MWIR).

Infrarrojo lejano o Térmico (IRT): (8 a 14 μm). Que incluye la porción emisiva del espectro terrestre, en donde se detecta el calor proveniente de la mayor parte de las cubiertas terrestres.

Microondas (M): por encima de 1 mm. Con gran interés por ser un tipo de energía bastante transparente a la cubierta nubosa.

Las regiones espectrales ópticas usadas en teledetección están resumidas en la tabla 2.1. Estas regiones espectrales son particularmente de interés porque contienen “ventanas” atmosféricas relativamente transparentes.

Nombre	Rango de longitud de onda	Fuente de radiación	Propiedad de la superficie de interés
Visible	0,4 – 0,7 μm	Solar	Reflectancia
IRC	0,7 – 1,2 μm	Solar	Reflectancia
SWIR	1,2 – 1,35 μm 1,4 – 1,8 μm 2 – 2,5 μm	Solar	Reflectancia
MWIR	3 – 4 μm 4,5 – 5 μm	Solar, termal	Reflectancia, temperatura
IRT	8 – 9,5 μm 10 – 14 μm	Termal	Temperatura
M	1mm – 1m	Termal (pasivo), artificial (activo)	Temperatura (pasivo), aspereza (activo)

Tabla 2.1: Las principales regiones espectrales usadas en teledetección.

Los términos “activo” y “pasivo” se explican en la sección 2.3.

2.2. Representación de imágenes digitales

Denotamos las *imágenes* por una función bidimensional de la forma $f(x,y)$. El valor o amplitud de f en las coordenadas espaciales (x,y) es una cantidad escalar positiva cuyo significado físico está determinado por la fuente de la imagen. Cuando una imagen se genera de un proceso físico, sus valores son proporcionales a la energía irradiada por la fuente de iluminación (ejemplo, ondas electromagnéticas). Como consecuencia, $f(x,y)$ debe ser positivo y finito; esto es,

$$0 \leq f(x,y) < \infty \quad (2.4)$$

Es conocido que de acuerdo a las propiedades físico-químicas de una superficie u objeto no toda la energía incidente es reflejada, parte de ella también es absorbida. Es importante observar que la radiación reflejada representa, en general, una porción “muy pequeña” del objeto, siendo la integración

de todas estas porciones, acorde a la longitud de onda de la radiación incidente, la que conforme el objeto en su totalidad [Sossa Azuela y Rodriguez Morales, 2011].

La función $f(x,y)$ puede ser caracterizada por dos componentes [Gonzalez y Woods, 2002]:

1. la cantidad de incidencia de la fuente de iluminación en la escena siendo vista
2. la cantidad de iluminación reflejada por los objetos en la escena.

Apropiadamente, estos son llamados los componentes de iluminación y reflectancia y son denotados por $i(x,y)$ y $r(x,y)$, respectivamente. Las dos funciones se combinan como un producto para formar $f(x,y)$:

$$f(x,y) = i(x,y)r(x,y) \quad (2.5)$$

donde

$$0 \leq i(x,y) < \infty \quad (2.6)$$

y

$$0 \leq r(x,y) \leq 1 \quad (2.7)$$

La ecuación 2.7 indica que la reflectancia está limitada por 0 (absorción total) y 1 (reflectancia total). La naturaleza de $i(x,y)$ es determinada por la fuente de iluminación, y $r(x,y)$ es determinada por las características del objetos a retratarse.

Una *imagen digital* es una matriz de números. Cada celda de una imagen digital es llamada *píxel* (del inglés picture element) y el número representando el brillo del píxel es llamado *Nivel Digital (ND)*. Cuando se visualiza un píxel, la intensidad de gris o de color con la que aparece en el monitor depende de su ND; la relación entre ambos puede modificarse a voluntad del intérprete. En ese sentido, conviene distinguir el ND de cada píxel adquirido por el sensor de su Nivel Visual (NV), que corresponde a la intensidad de gris o color con el que se visualiza ese píxel en el monitor.

Asumamos que una imagen digital $f(x,y)$ tiene M filas y N columnas. Los valores de las coordenadas (x,y) son valores discretos. De esta forma, los valores de las coordenadas en el origen son $(x,y) = (0,0)$. Los siguientes valores de coordenadas a lo largo de la primer fila de la imagen está representada como $(x,y) = (0,1)$. Es decir, el origen de coordenadas se sitúa en la esquina superior izquierda.

La notación introducida en el párrafo anterior nos permite escribir la imagen digital completa $M \times N$ en la siguiente forma:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.8)$$

El lado derecho de esta ecuación es por definición una imagen digital. Cada elemento de esta matriz es llamado píxel.

El proceso de digitalización requiere decisiones sobre los valores de M , N , y para el número, L , de ND permitidos para cada píxel. No hay requisitos sobre M y N , más que tienen que ser enteros positivos. Sin embargo, debido a consideraciones de hardware, el número de niveles de gris típicamente es un entero potencia de 2:

$$L = 2^k \quad (2.9)$$

Asumimos que los niveles discretos son espaciados equitativamente y que son enteros en el intervalo $[0, L - 1]$, donde 0 será el nivel de gris más oscuro mientras que el valor $L - 1$ será el nivel de gris más claro. El número, b , de bits requeridos para guardar una imagen digitalizada es

$$b = M \times N \times k \quad (2.10)$$

Cuando una imagen puede tener 2^k niveles de gris, es una práctica común referirse a la imagen como una “imagen de k bits”.

Los datos de la imagen digital pueden tener también una tercer dimensión: capas. Las capas son las imágenes de la misma escena pero conteniendo diferente información, como se muestra en la figura 2.3. En *imágenes multispectrales*, las capas son las imágenes de rangos espectrales diferentes llamados bandas o canales. Dicho de otra forma, la tercer dimensión corresponde a la dimensión espectral. El término “banda” es usado más frecuentemente que “capas” para referirse a imágenes multispectrales [Liu y Mason, 2009].

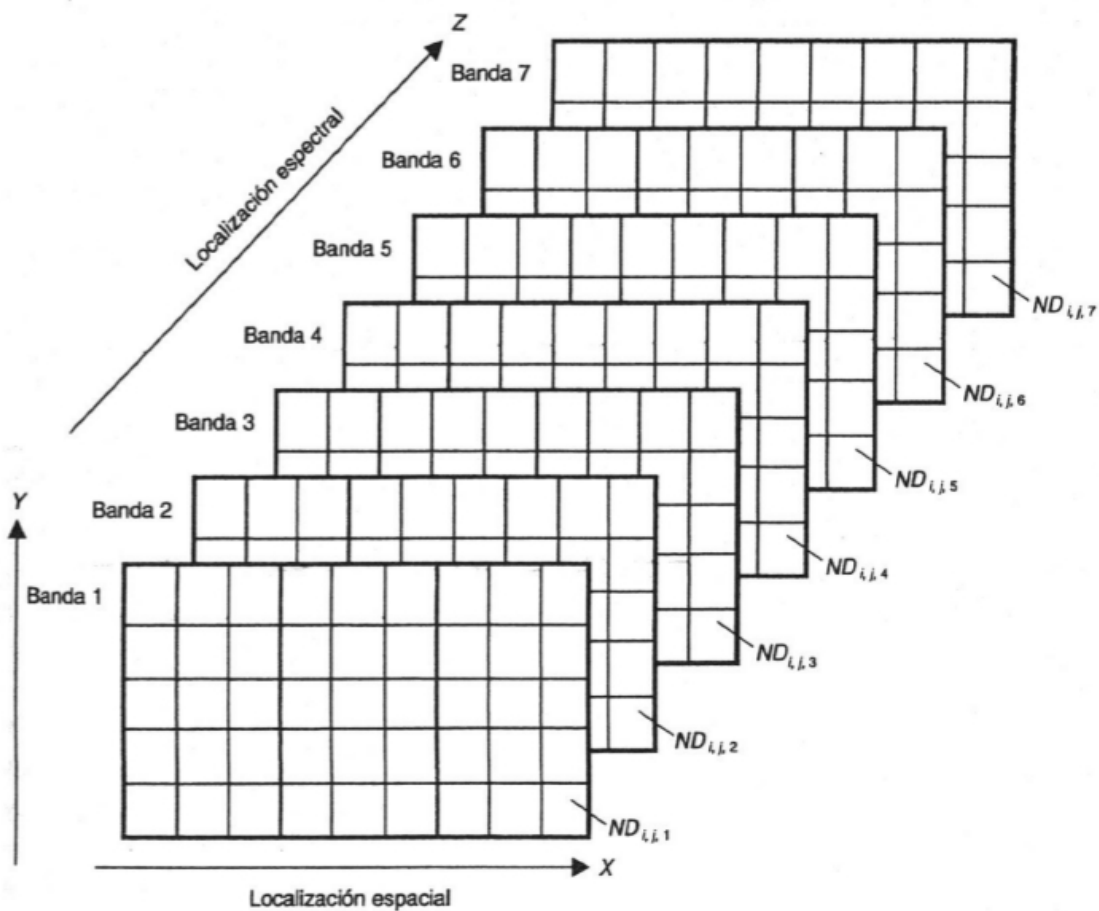


Figura 2.3: Representación de una imagen multispectral.

2.2.1. Vecinos de un píxel

Un píxel p en las coordenadas (x, y) tiene cuatro *vecinos* horizontales y verticales cuyas coordenadas están dadas por

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

Los cuatro vecinos diagonales de p tienen coordenadas

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

Algunos de estos puntos caen fuera de la imagen si p está en el borde de la misma.

Las operaciones sobre *vecindarios* determinan el valor de un píxel dado a partir del brillo de un conjunto de los píxeles que lo rodean.

2.3. Sensores

El elemento clave en la tecnología de teledetección es el *sensor*. Un sensor detecta y graba cuantitativamente la radiación electromagnética de un objeto remotamente; de aquí el término "sensado". Para que un objeto sea sensado, debe radiar energía ya sea reflejando la radiación impactada sobre este por una fuente de iluminación, o "brillando" por si mismo.

Existen dos tipos de sensores:

1. Los *sensores pasivos* se limitan a recibir energía proveniente de un foco exterior a ellos.
2. Los *sensores activos* son capaces de emitir su propio haz de energía.

Ambos tipos de sensores tienen como objetivo la observación de la superficie terrestre en toda su extensión y se los puede clasificar como [Rabolli y Gulich, 2005]:

- No generadores de imágenes: trabajan a una sola longitud de onda.
- Generadores de imágenes: operan en varias bandas, barriendo por franjas perpendiculares al desplazamiento del satélite.

A su vez los sensores pasivos pueden clasificarse según su procedimiento para grabar la energía recibida. De acuerdo a ello, puede hablarse de sensores fotográficos, óptico-electrónicos y de antena. Los sensores ópticos pancromáticos y multispectrales más comúnmente usados son los sensores pasivos típicos [Chuvieco, 2010].

Este trabajo está orientado al tratamiento de imágenes obtenidas por sistemas sensores pasivos, por lo que no se ahondará en información relativa a sensores activos.

2.3.1. Resolución de un sistema sensor

La *resolución* de un sensor depende del efecto combinado de sus distintos componentes. El concepto de resolución implica diversos aspectos como el espacial, espectral, radiométrico, temporal y angular [Chuvieco, 2010]. En las siguientes secciones se verán solo los tres primeros ya que son los que más influencia tienen sobre el objetivo de este trabajo.

2.3.1.1. Resolución espacial

El término *resolución espacial* designa al objeto más pequeño que puede ser distinguido sobre una imagen. En un sistema fotográfico, suele medirse como la mínima separación a la cual los objetos aparecen distintos y separados en la fotografía. Se mide en milímetros sobre la foto (o metros sobre el

terreno), y depende de la longitud focal de la cámara y de su altura sobre la superficie. En los sensores óptico-electrónicos, se prefiere utilizar el concepto de *campo de visión instantáneo* (Instantaneous Field Of View, IFOV). El IFOV se define como la sección angular que observa el sensor en un momento determinado. En lugar de esta definición, habitualmente suele utilizarse la distancia sobre el terreno que corresponde a ese ángulo (d), que puede calcularse a partir de él, conociendo la distancia del suelo del sensor (h):

$$d = 2 h \tan(IFOV/2) \quad (2.11)$$

asumiendo que se trate de una observación vertical. La distancia del IFOV proyectada al suelo se corresponde con el tamaño de la mínima unidad de información incluida en la imagen, que se denomina píxel.

La resolución espacial de un sensor depende de varios factores, como son la altura orbital, la longitud focal, y el número de detectores. En cuanto a los sensores de antena, su resolución depende del tamaño de la misma, de la altura de la plataforma y del ángulo de incidencia. Cuanto mayor sea el radio, y menor la altitud y la longitud de onda, la resolución será tanto más detallada.

La resolución espacial tiene un papel protagonista en la interpretación de la imagen, por cuanto marca, de alguna forma, el nivel de detalle que ésta ofrece. Parece obvio decir que sólo serán identificados sobre la imagen elementos que superen el tamaño de varios píxeles, si bien pueden detectarse rasgos del tamaño similar o incluso más pequeño al del píxel cuando exista suficiente contraste radiométrico. En consecuencia, la resolución espacial está estrechamente ligada con la escala de trabajo y con la fiabilidad finalmente en la interpretación.

2.3.1.2. Resolución espectral

La *resolución espectral* indica el número y anchura de las bandas espectrales que puede discriminar el sensor. En este sentido, un sensor será tanto más idóneo cuanto mayor número de bandas proporcione, ya que facilita la caracterización espectral de las distintas cubiertas. A la vez, conviene que esas bandas sean suficientemente estrechas, con objeto de recoger la señal sobre regiones coherentes del espectro. Bandas muy amplias suponen registrar un valor promedio, que puede encubrir la diferenciación espectral entre cubiertas de interés.

La elección del número, anchura y localización de las bandas que incluye el sensor está estrechamente relacionada con los objetivos de su diseño.

2.3.1.3. Resolución radiométrica

La *resolución radiométrica* (o de nivel de gris) se refiere al cambio discernible más pequeño en nivel de gris, esto es, a su capacidad para detectar variaciones en la radiancia espectral que recibe. Pero medir cambios discernibles en nivel de gris es un proceso altamente subjetivo.

En el caso de los sistemas fotográficos, la resolución radiométrica del sensor se indica por el número de niveles de gris distintos que aparecen en la fotografía. Para los equipos digitales, la imagen se codifica en un formato binario, por lo que la resolución radiométrica suele identificarse con el rango posible de valores que almacena el sensor, medido como el número de bits que necesita cada valor numérico para almacenarse. Debido a las consideraciones de hardware, el número de niveles de gris es usualmente un entero potencia de 2. El número más común es 8 bits, con 16 bits siendo usado en algunas aplicaciones donde la mejora de rangos específicos de niveles de gris es necesario.

La resolución radiométrica, como ocurre con las dos anteriormente vistas, resulta de notable ayuda a la interpretación de imágenes, especialmente cuando realizamos análisis digital. Es preciso consi-

derar que el número de niveles de gris que son capaces de discriminar nuestros ojos no excede de 64, mientras las tonalidades de color no superan las 200.000. Sin embargo, cuando la interpretación es digital, la computadora aprovecha todo el rango disponible, con lo que una mayor resolución radiométrica puede permitir discriminar objetos con niveles muy similares de reflectividad, que no serían separables con otros sensores menos sensibles.

2.3.2. Sensores pasivos

Los *sensores pasivos* se limitan a recoger la energía electromagnética procedente de las cubiertas terrestres, ya sea ésta reflejada de los rayos solares, ya emitida en virtud de su propia temperatura. Puede establecerse una división en estos sensores en función del procedimiento que emplean para recibir la radiación procedente de los objetos. En el grupo de los sensores fotográficos estarían las cámaras analógicas. En el grupo de los óptico-electrónicos están los exploradores de barrido y de empuje, y las cámaras de video. Mientras que en el grupo de los sensores de antena se conocen con el nombre de radiómetros de microondas.

2.3.2.1. Exploradores de barrido

Los *exploradores de barrido* (scanners), como se muestra en la figura 2.4, constan de un espejo móvil que oscila perpendicularmente a la dirección de trayectoria que les permite explorar una franja de terreno a ambos lados de la traza del satélite. La radiancia recibida por este componente óptico se dirige a una serie de detectores, que la amplifican y convierten a una señal eléctrica. Ésta, a su vez, se transforma en un valor numérico, que puede almacenarse a bordo o transmitirse a la red de antenas receptoras.

El número y características de los detectores que incorpora un equipo de barrido es muy importante para explicar las condiciones de la imagen resultante. La señal enviada por el equipo óptico a estos detectores se muestrea a intervalos regulares, de tal forma que únicamente se graba un valor numérico cada cierta distancia. Ese intervalo marca el tamaño del píxel. En la mayor parte de los equipos de exploración por barrido, la señal recibida se descompone a bordo en varias longitudes de onda, cada una de las cuales se envía a un tipo especial de detectores, sensibles a ese tipo de energía. En ese caso, se habla de exploradores de barrido multiespectral (o incluso hiperespectral), pues detectan la misma superficie de terreno en distintas bandas del espectro. El número y anchura de esas bandas (la resolución espectral) está en función del diseño del propio sensor: del número de detectores con que cuenta, de su óptica, capacidad de almacenamiento y transmisión de datos, etc.

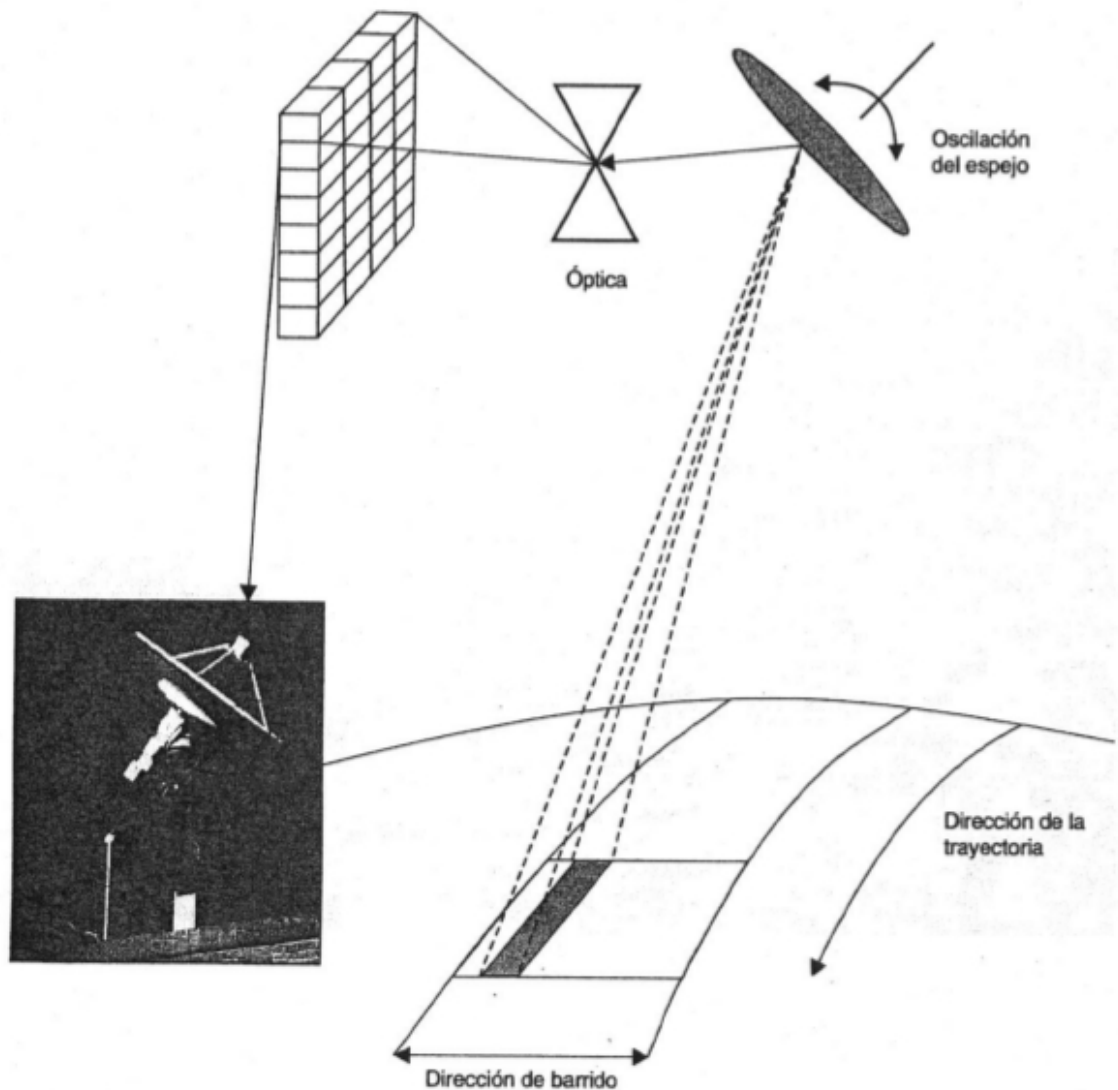


Figura 2.4: Representación de un explorador de barrido.

2.3.2.2. Exploradores de empuje

En la figura 2.5 se muestra que en los equipos de *exploración por empuje* se elimina el espejo oscilante, gracias a disponer de una cadena de detectores que cubre todo el campo de visión del sensor. Estos detectores se van excitando con el movimiento orbital del satélite, de ahí el nombre de explorador de empuje, ya que explora, en cada momento, una línea completa, desplazándose ésta simultáneamente con la plataforma. Los detectores sólidos que forman un explorador por empuje se denominan dispositivos de acoplamiento por carga (Charge Couple Devices, CCD).

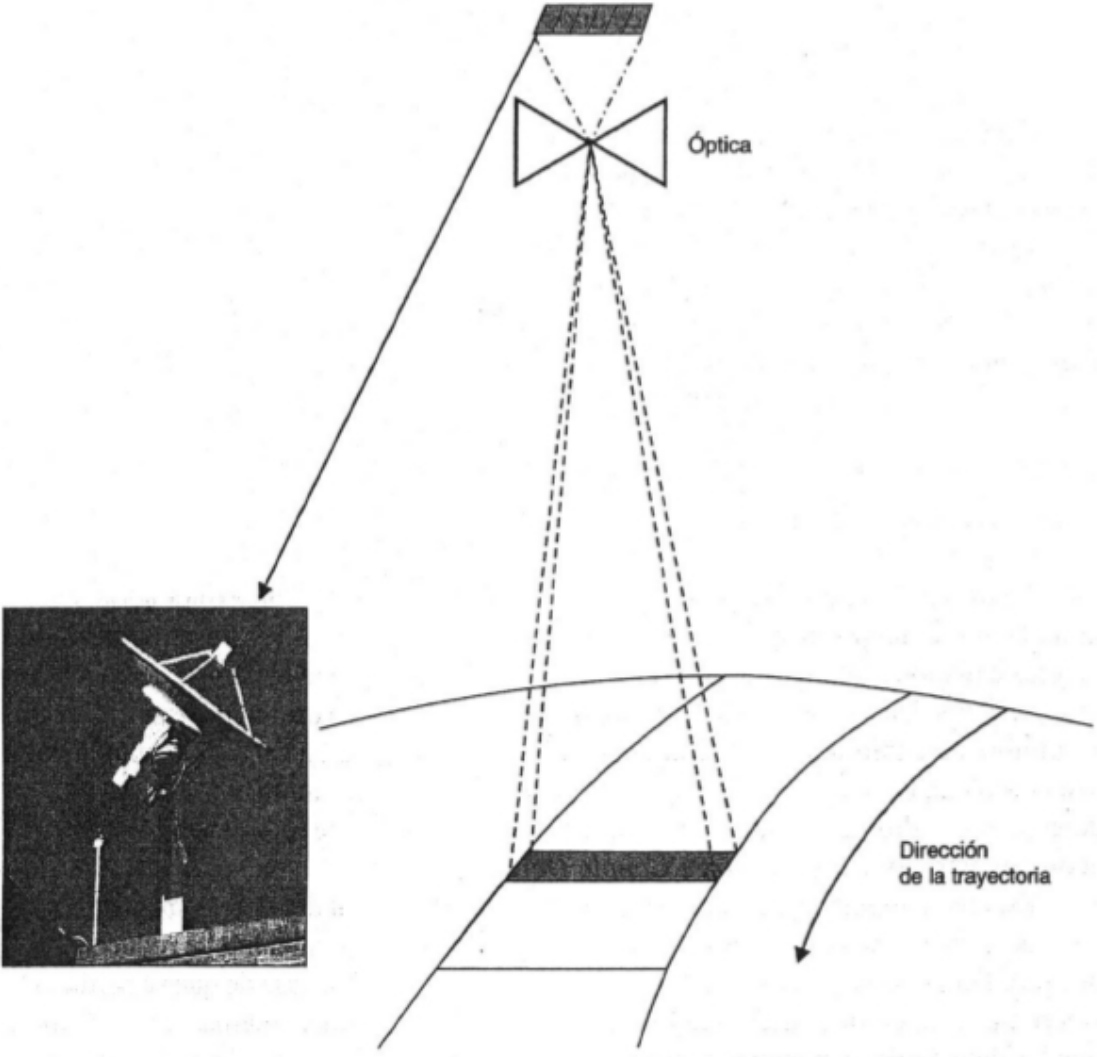


Figura 2.5: Representación de un explorador de empuje.

CAPÍTULO 3

CORRECCIONES DE RUIDO EN IMÁGENES

La restauración intenta reconstruir o recuperar una imagen degradada usando a priori conocimiento del fenómeno de degradación. De esta forma las técnicas de restauración están orientadas a modelar la degradación y aplicar el proceso inverso para recuperar la imagen original. Esta estrategia usualmente involucra formular criterios de aceptación que producirán una estimación óptima del resultado deseado.

Algunas veces el procesamiento de imágenes digitales puede recuperar la información espectral descalibrada y hacerla relativamente compatible con los datos adquiridos correctamente en la escena. Desafortunadamente, algunas veces solo ajustes cosméticos pueden hacerse para compensar el hecho que ningún dato de valor fue adquirido.

Este capítulo en la sección 3.1 tiene como objetivo mostrar los diferentes tipos de ruido que pueden presentarse en imágenes satelitales, mostrando sus características más importantes y la forma en que estos pueden producirse. También se verá, en la sección 3.2, los diferentes filtros que serán usados para reconstruir imágenes alteradas por ruidos.

3.1. Ruido en la imagen

Como muestra la figura 3.1, el proceso de degradación es modelado como una función que, junto a un término de ruido aditivo, opera sobre una imagen de entrada $f(x,y)$ para producir una imagen degradada $g(x,y)$. Dada $g(x,y)$, algún conocimiento sobre la función de degradación H , algún conocimiento sobre el término de ruido aditivo $\eta(x,y)$, el objetivo de la restauración es obtener una estimada $\hat{f}(x,y)$ de la imagen original. Queremos que la estimada sea lo más cercana posible a la imagen original de entrada y, en general, mientras más sabemos de H y $\eta(x,y)$ más cerca estará $\hat{f}(x,y)$ de $f(x,y)$.

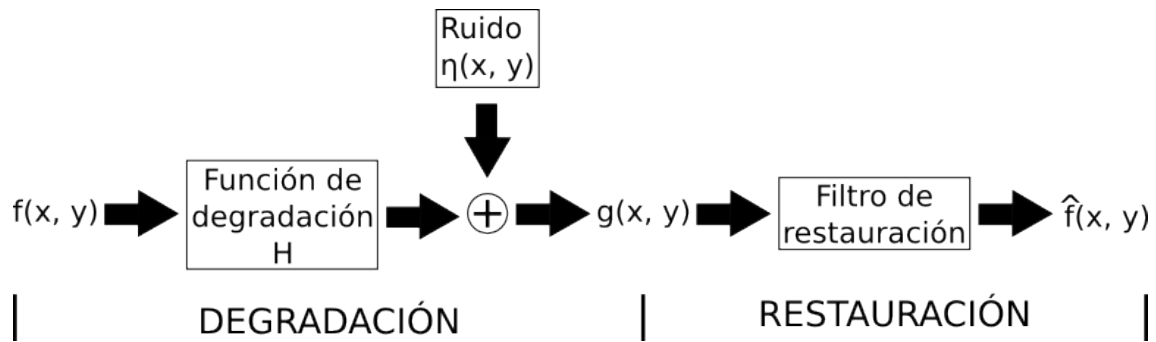


Figura 3.1: Un modelo del proceso de degradación/restauración de una figura

La imagen degradada está dada por

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \quad (3.1)$$

donde $h(x,y)$ es la representación espacial de la función de degradación y el símbolo “*” indica convolución. En esta sección asumimos que H es el operador identidad, y solo nos ocupamos de las degradaciones producidas por el ruido.

En esta sección veremos las diferentes fuentes de error y un modelado del ruido para explicar la forma en que estos afectan la imagen.

3.1.1. Fuentes de error en una imagen espacial

Idealmente, la radiancia grabada por un sistema de sensado remoto es una representación exacta de la radiancia emitida por el objeto de interés en la superficie de la Tierra. Desafortunadamente, ruido (error) puede introducirse en el sistema de colección de datos en varios puntos. Algunos de los errores radiométricos inducidos por el sistema de sensado remoto más comunes son [Jensen, 2005]:

- píxeles erróneos aleatorios (ruido de disparo)
- problemas de comienzo o finalización de línea
- pérdida de línea o columna
- pérdida parcial de línea o columna
- bandeo vertical u horizontal

Las principales fuentes de ruido en una imagen se producen durante la adquisición y/o transmisión. El desempeño de los sensores de imagen son afectados por una variedad de factores, como las condiciones de entorno en el momento de la adquisición, y la calidad de los elementos sensores mismos.

3.1.1.1. Líneas o píxeles perdidos

Presentaremos cinco situaciones en las que una imagen puede presentar pérdida de información y la forma en que esto se produce.

Algunas veces un *detector individual* no guarda datos espectrales para píxeles individuales. Cuando esto ocurre puede resultar que *numerosos píxeles erróneos aleatorios* aparezcan en la imagen, lo cual es llamado ruido de disparo porque aparenta que la imagen fue disparada por una escopeta. Al visualizarla, esta imagen muestra una serie de píxeles aislados de aspecto muy contrastado con los vecinos. Normalmente, no suele encontrarse un patrón que indique cuando un píxel es erróneo, ya que estos pueden contener cualquier valor de ND y se puede presentar en una o más bandas.

Por otro lado, una *línea entera sin información espectral* puede producirse si un detector individual en un *sistema de escaneo* falla al funcionar. También, si un detector en un *arreglo lineal* falla al funcionar, puede resultar en una *columna entera de datos sin información espectral*. Una pérdida de línea o columna se presenta como una serie de líneas anómalas (negras o blancas) en la imagen.

También suele ocurrir que un detector individual funcione perfectamente a lo largo de una línea de escaneo y luego por una razón desconocida no funcionará correctamente para una cierta cantidad de columnas. Luego el detector funciona correctamente de nuevo por el resto de la línea de escaneo. El resultado es una *porción de la línea de escaneo sin datos*. Es decir, una pérdida parcial de línea o columna.

Además, los *sistemas de escaneo* ocasionalmente fallan al recolectar datos al comienzo de la línea de escaneo, o ubican el pixel de dato en lugares inapropiados a lo largo de la línea de escaneo. Esto es llamado un *problema de comienzo de línea*. También, un detector puede parar abruptamente de recolectar datos en algún lado a lo largo de un escaneo y producir resultados similares a la pérdida de línea o columna. Idealmente, cuando los datos no son recolectados, el sistema sensor puede programarse para recordar que no fue recolectado y poner cualquier dato bueno en la ubicación geométrica adecuada a lo largo del escaneo. Desafortunadamente, este no es siempre el caso.

Por último, fruto de un error en la *recepción de datos*, la imagen puede mostrar *píxeles con valores anómalos*, es decir, esto también se expresa con píxeles cuyo ND es significativamente más alto o más bajo que el promedio de sus vecinos. Las imágenes que se corrompen durante la transmisión principalmente son debido a la interferencia en el canal usado para la misma.

En todos estos casos, se trata de información irreparablemente perdida. Por lo tanto, los procesos de corrección se dirigen a mejorar artificialmente el aspecto visual de la imagen, facilitando su posterior interpretación.

3.1.1.2. Bandedo de la imagen

En algunas imágenes obtenidas por equipos de *barrido secuencial*, se observa un *bandedo de la imagen*, especialmente perceptible en las zonas de baja radiancia. Este bandedo se debe a un mal calibrado entre detectores del sensor; esto es que alguno de ellos codifique la radiancia que recibe en ND distintos a los demás, es decir, pierde su ajuste radiométrico. Este efecto aparece en la imagen con líneas más oscuras o claras que el resto. El efecto además es periódico.

Para imágenes tomadas por *sensores de empuje*, la calibración puede llegar a ser aún más problemática, ya que para explorar simultáneamente una franja de terreno amplia, el sensor cuenta con una cadena de detectores muy numerosa. En este caso, una falta de calibración entre detectores se observará en un *bandedo vertical*, ya que cada columna se adquiere por un detector distinto.

La línea mal ajustada contiene información valiosa, pero debe ser corregida para tener aproximadamente la misma escala radiométrica que los datos recolectados por detectores calibrados correctamente asociados con la misma banda.

3.1.2. Modelado del ruido

Como muestra la figura 3.1, el descriptor del ruido espacial es el comportamiento estático de los valores de gris en el componente de ruido del modelo. Estas pueden ser consideradas variables aleatorias, caracterizadas por una *función de densidad de probabilidad* (de aquí en adelante las llamaremos simplemente *densidad*). Las siguientes secciones muestran las densidades más comunes encontradas en aplicaciones de procesamiento de imágenes.

Cuando la única degradación presente en una imagen es ruido, la ecuación (3.1) se vuelve

$$g(x,y) = f(x,y) + \eta(x,y) \quad (3.2)$$

El filtrado espacial es el método de elección en situaciones cuando solo el ruido aditivo está presente.

3.1.2.1. Ruido Gaussiano

El modelo de *ruido Gaussiano* (también llamado *normal*) es usado frecuentemente. La densidad de una variable aleatoria Gaussiana, z , está dada por

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2} \quad (3.3)$$

donde z representa el nivel de gris, μ es la media del promedio de z , y σ es la desviación estándar. Cuando z está descrita por (3.3), aproximadamente el 70% de sus valores estarán en el rango $[(\mu - \sigma), (\mu + \sigma)]$, y cerca del 95% estarán en el rango $[(\mu - 2\sigma), (\mu + 2\sigma)]$

El ruido Gaussiano se presenta en una imagen por factores como ruidos de circuitos electrónicos y sensores debido a iluminación pobre y/o temperatura alta [Gonzalez y Woods, 2002].

Este tipo de ruido produce pequeñas variaciones en la imagen y afecta a la imagen completa. La intensidad de todos los píxeles se ve alterada.

3.1.2.2. Ruido uniforme

La densidad del *ruido uniforme* está dada por

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{si } a \leq z \leq b \\ 0 & \text{caso contrario} \end{cases} \quad (3.4)$$

La media de esta función de densidad está dada por

$$\mu = \frac{a+b}{2} \quad (3.5)$$

y su varianza por

$$\sigma^2 = \frac{(b-a)^2}{12} \quad (3.6)$$

La densidad uniforme es quizás la menos descriptiva en situaciones prácticas, sin embargo, es muy útil como base para generadores de números aleatorios que son usados en simuladores [Gonzalez y Woods, 2002].

3.1.2.3. Ruido impulsivo (sal y pimienta)

La densidad del *ruido impulsivo* (bipolar) está dada por

$$p(z) = \begin{cases} P_a & \text{si } z = a \\ P_b & \text{si } z = b \\ 0 & \text{caso contrario} \end{cases} \quad (3.7)$$

Si $b > a$, el nivel de gris b aparecerá como un punto de luz en la imagen. En cambio, el nivel de gris a aparecerá como un punto oscuro. Si P_a o P_b es cero, el ruido impulsivo es llamado unipolar. Si ninguna de estas probabilidades es cero, y especialmente si son aproximadamente iguales, los valores de ruido impulsivo se parecerán a granos de sal y pimienta distribuidos aleatoriamente sobre la imagen. Por esta razón, el ruido impulsivo bipolar también es llamado *ruido de tipo “sal y pimienta”*.

Los impulsos de ruido pueden ser positivos o negativos. La suposición usualmente es que a y b son valores “saturados”, en el sentido que son iguales al mínimo y máximo valor permitido en la imagen digital. Como resultado, los impulsos negativos aparecen como puntos negros (pimienta) y los positivos como ruido blanco (sal).

El ruido de impulso se encuentra en situaciones en las que los transitorios rápidos, como la conexión errónea, se llevan a cabo durante la exploración [Gonzalez y Woods, 2002].

3.2. Filtros espaciales

La forma más lógica de estimar los ND de los píxeles erróneos se basa en utilizar los ND de los píxeles vecinos. De acuerdo al fenómeno de la autocorrelación espacial, muchas variables geográficas tienden a presentar una fuerte asociación en el espacio [Chuvieco, 2010]. Por esta razón, para lograr una reducción de ruido, algunas operaciones trabajan con los valores de los píxeles en un vecindario de la imagen y los valores correspondientes de una subimagen que tiene las mismas dimensiones que el vecindario. La subimagen es llamada filtro, máscara, núcleo, plantilla o ventana, con las tres primeras siendo la terminología prevalente. Los valores en una subimagen de filtro son referidas como coeficientes, en vez de píxeles.

El filtrado puede ser implementado basado en “filtros de caja” fundamentados en el concepto de convolución en el dominio espacial o usando la transformada de Fourier en el dominio de las frecuencias. En la práctica del sensado remoto, los filtros de caja basados en convolución son más útiles por su eficiencia en el cálculo, sus resultados confiables [Liu y Mason, 2009], son relativamente fáciles de entender [Jensen, 2005] y tienen un costo computacional menor cuando la máscara que se utiliza es pequeña [Richards y Jia, 2006]. Estamos interesados en operaciones de filtrado que se realizan directamente en los píxeles de una imagen. Para este tipo de filtros usamos el término *filtros espaciales*.

En general, el proceso de filtrado resulta en una imagen con transiciones “bruscas” reducidas en niveles de gris. Ya que el ruido aleatorio típicamente consiste en transiciones bruscas, la aplicación más obvia del filtrado es la reducción de ruido. Sin embargo, los bordes (los cuales casi siempre son

rasgos deseables de una imagen) también están caracterizadas por transiciones bruscas en niveles de gris, por lo que algunos tipos de filtros suelen tener un efecto de cola no deseado que desdibuja bordes.

Una característica de las imágenes de sensorado remoto, es un parámetro llamado frecuencia espacial, definida como el número de cambios en valores de brillo por unidad de distancia para cualquier parte particular de una imagen. Si hay muy pocos cambios en valores de brillo sobre un área dada, es llamada comunmente como área de frecuencia baja. Los filtros de suavizado son diseñados para eliminar información de frecuencias altas y retener la de frecuencias bajas, y de este modo reducir el ruido pero con el costo de degradar detalle de la imagen. A estos filtros se los llama *filtros de paso bajo* o *filtros de frecuencia baja*. Para este tipo de filtros mientras más grande sea la máscara, es menor la información de frecuencia baja retenida [Liu y Mason, 2009]. Para la reducción de ruido sin hacer borrosas las imágenes, el suavizado preservando bordes se vuelve un tema de investigación importante en el filtrado.

Además de la reducción de ruido, los filtros son usados para difuminar la imagen. El difuminado es usado en pasos de preprocesamiento, tales como remoción de pequeños detalles de una imagen anterior a la extracción de objetos (grandes), y zanjado de pequeños huecos en líneas y curvas [Gonzalez y Woods, 2002]. Por otro lado, los filtros espaciales son útiles para mejorar rasgos que pueden representar estructuras geológicas significativas. También pueden mejorar la textura de imagen para discriminar litologías y patrones de drenaje. Para estudios de uso de suelo, el filtrado puede sobresaltar las texturas de urbanización, sistemas de ruta, áreas de agricultura, y para visualización general. El filtrado es ampliamente usado para mejorar imágenes. Sin embargo, se debe tener cuidado porque el filtrado no es tan confiable al retener la información de la imagen original [Liu y Mason, 2009].

Si los valores de brillo cambian dramáticamente en distancias cortas, esta es un área de detalle de frecuencia alta. Los bordes y texturas en una imagen son ejemplos típicos de información de frecuencia alta. Los *filtros de paso alto* o *filtros de frecuencia alta* eliminan información de frecuencia baja en la imagen y, de esta forma resaltan la información de frecuencia alta como los bordes o líneas. Debido a la motivación de este trabajo, este tipo de filtros no serán tenidos en cuenta.

3.2.1. Bases del filtrado espacial

La mecánica del *filtrado espacial* se ilustra en la figura 3.2. El proceso consiste simplemente en mover la máscara de filtro de píxel a píxel en una imagen. En cada píxel (x,y) , la respuesta del filtro es calculada usando una relación predefinida. Para filtrado espacial lineal, la respuesta está dada por la suma de productos de los coeficientes de filtro y los píxeles de imagen correspondientes en el área alcanzada por la máscara de filtro.

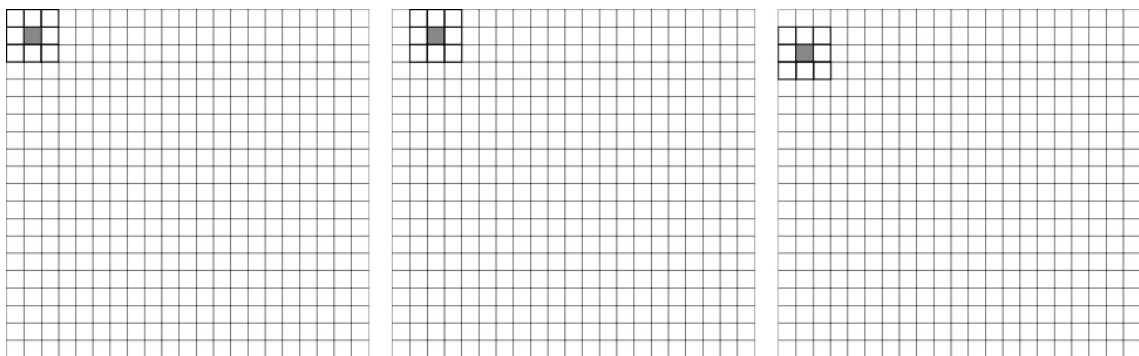


Figura 3.2: Una ventana de un filtro espacial. El primer píxel de salida es calculado (izquierda), el siguiente píxel de salida es calculado para la misma fila (centro), y luego de que la fila es completada el proceso se repite para la siguiente fila (derecha). El píxel de salida es ubicado en la coordenada del píxel sombreado en la imagen de salida.

En general, el filtrado lineal de una imagen f de tamaño $M \times N$ con una máscara de filtro de tamaño $m \times n$ está dada por la expresión:

$$g(x,y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s,t)f(x+s,y+t) \quad (3.8)$$

donde, $a = (m - 1)/2$ y $b = (n - 1)/2$. Para generar una imagen filtrada completa esta ecuación debe aplicarse para $x = 0, 1, \dots, M - 1$ e $y = 0, 1, \dots, N - 1$. De esta forma, nos aseguramos que la máscara procesará todos los píxeles en la imagen. Notar en particular que el coeficiente $w(0,0)$ coincide con el valor de imagen $f(x,y)$, indicando que la máscara está centrada en (x,y) cuando se realiza el cálculo de la suma de productos. Para una máscara de tamaño $m \times n$, asumimos que $m = 2a + 1$ y $n = 2b + 1$, donde a y b son enteros no negativos. Todo esto dice que las máscaras tendrán tamaños impares, con el tamaño significativo más chico siendo 3×3 .

El proceso de *filtrado lineal* dado en la expresión (3.8) es similar al concepto de dominio de frecuencia llamado convolución. Por esta razón, el filtrado espacial lineal suele referenciarse como “convolucionando una máscara con una imagen”. Similarmente, las máscaras de filtro son a veces llamadas *máscaras de convolución* y como consecuencia los filtros espaciales también son llamados *filtros de convolución*.

Los *filtros no lineales* también operan sobre vecindarios, y la mecánica de deslizado de máscara por una imagen son las mismas. En general, sin embargo, la operación de filtrado está basado condicionalmente en los valores de los píxeles en el vecindario en consideración, y no usan explícitamente coeficientes en la manera suma de productos descrita en la expresión (3.8). La reducción de ruido puede lograrse difuminando tanto con filtros lineales como no lineales [Gonzalez y Woods, 2002].

Una consideración importante al implementar operaciones para filtrado espacial es el problema de qué pasa cuando el centro del filtro se acerca al borde de la imagen, planteado en la figura 3.3. Consideremos por simplicidad una máscara cuadrada de tamaño $n \times n$. Al menos un borde de tal máscara coincidirá con el borde de la imagen cuando el centro de la máscara esté a una distancia de $(n - 1)/2$ píxeles del borde de la imagen. Si el centro de la máscara se mueve más cerca del borde, una o más filas o columnas de la máscara se ubicarán fuera del plano de la imagen. La manera más simple de manejar esta situación es limitando las ejecuciones del centro de la máscara a una distancia no menos a $(n - 1)/2$ píxeles del borde. La imagen filtrada resultante será más pequeña que la original, pero todos los píxeles en la imagen filtrada serán procesadas con la máscara completa.

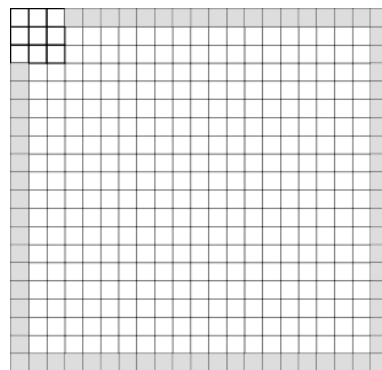


Figura 3.3: El problema de la región del borde de una imagen para un filtro.

3.2.2. Filtros espaciales lineales

La salida de un *filtro espacial lineal* es simplemente el promedio de los píxeles contenidos en el vecindario definido por la máscara. Estos filtros a veces son llamados filtros de promediado o filtros de paso bajo. Cuando todos los coeficientes de la máscara son iguales a veces el filtro es llamado un filtro de caja.

3.2.2.1. Filtro de media aritmética

Sea S_{xy} el conjunto de coordenadas en una ventana rectangular de tamaño $m \times n$, centrado en el punto (x, y) . El *filtro de media aritmética* calcula el valor promedio de la imagen corrupta $g(x, y)$ en el área definida por S_{xy} . El valor de la imagen restaurada $\hat{f}(x, y)$ en cualquier punto (x, y) es simplemente la media aritmética calculada usando los píxeles en la región definida por S_{xy} . En otras palabras,

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t) \quad (3.9)$$

Esta operación puede ser implementada usando una máscara de convolución en la cual todos los coeficientes tienen valor $1/mn$. El filtro de media simplemente suaviza variaciones locales en una imagen. El ruido es reducido como resultado de hacer borrosa la imagen [Gonzalez y Woods, 2002].

Una aplicación del promediado espacial es difuminar una imagen con el propósito de obtener una representación más grosera de objetos de interés, tal que la intensidad de los objetos más pequeños se mezcla con el fondo y los objetos más grandes se “manchan” y son más fáciles de detectar. El tamaño de la máscara establece el tamaño relativo de los objetos que serán mezclados con el fondo, es decir, el difuminado se vuelve más severo a medida que el tamaño del núcleo aumenta [Gonzalez y Woods, 2002]. Si se aumenta mucho el tamaño de la máscara el resultado que se obtendrá no será bueno. Mientras más grande sea la máscara más suaviza la imagen y más detalle de frecuencia alta se pierde [Richards y Jia, 2006]. Cuando la imagen aparece muy ruidosa lo mejor es aplicar máscaras de tamaño pequeños en forma de cascada, o sea, al resultado de un filtrado volverle aplicar la misma máscara [Sossa Azuela y Rodriguez Morales, 2011].

Para el ruido impulsivo el filtro de media no tiene, prácticamente, ningún efecto [Sossa Azuela y Rodriguez Morales, 2011].

3.2.2.2. Filtro de media geométrica

Una imagen restaurada usando un *filtro de media geométrica* está dada por la expresión

$$\hat{f}(x, y) = \left[\prod_{(s, t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}} \quad (3.10)$$

Aquí, cada pixel restaurado está dado por el producto de los píxeles en la ventana, elevado a la potencia $1/mn$. Un filtro de media geométrica consigue un suavizado comparable con el filtro de media aritmética, pero tiende a perder menos detalle de imagen en el proceso [Gonzalez y Woods, 2002]. Este filtro es bueno para eliminar el ruido Gaussiano, reteniendo mejor los detalles de la información de el filtro de media aritmética, pero falla con el ruido impulsivo.

3.2.2.3. Filtro de media armónica

El *filtro de media armónica* está dado por la expresión

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}} \quad (3.11)$$

Este filtro tiene buenos resultados para el ruido de tipo “sal”, pero falla para el ruido de tipo “pimienta”. También trabaja bien con otros tipos de ruidos como el Gaussiano [Gonzalez y Woods, 2002].

Este filtro puede remover algunos tipos de ruido eficientemente sin deteriorar características importantes de una imagen, ya que tiende a preservar bordes en las imágenes [Kundu et al., 1984].

3.2.2.4. Filtro de media contrarmónica

El *filtro de media contrarmónica* produce una imagen restaurada basada en la expresión

$$\hat{f}(x,y) = \frac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^Q} \quad (3.12)$$

donde Q es llamado el *orden del filtro*. Este filtro es adecuado para reducir o virtualmente eliminar los efectos del ruido de tipo “sal y pimienta”. Para valores positivos de Q el filtro elimina ruido de tipo “pimienta”. Para valores negativos de Q elimina ruidos de tipo “sal”. No puede eliminar ambos simultáneamente [Gonzalez y Woods, 2002]. El filtro contrarmónico se reduce al filtro de media aritmética si $Q = 0$, y al filtro de media armónica si $Q = -1$.

Este filtro exhibe un buen desempeño en remover ruido impulsivo sin deteriorar las características importantes de la imagen, ya que tiende a preservar bordes en las imágenes [Kundu et al., 1984].

3.2.2.5. Filtro Gaussiano

Un *filtro Gaussiano* es un filtro de suavizado con una máscara de filtrado definida por una función bidimensional Gaussiana

$$w(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.13)$$

donde σ es la desviación estándar de la función Gaussiana.

La atenuación del ruido por medio de un filtro Gaussiano es menos agresiva a los bordes debido a la forma de esta función, pues la parte central alisará el ruido con mayor fuerza que en la periferia [Sossa Azuela y Rodriguez Morales, 2011]. Es mejor utilizar para el filtrado un valor de σ pequeño.

Obviamente, este filtro es esencialmente un filtro de media ponderado.

3.2.3. Filtros espaciales de orden

Los *filtros de orden* son filtros espaciales no lineales cuya respuesta está basada en ordenar (posicionar) los píxeles contenidos en el área de la imagen incluida en el filtro, y luego reemplazando el valor del píxel central con el valor determinado por el resultado de la clasificación. La respuesta del filtro en cualquier punto está determinado por el resultado del posicionado.

3.2.3.1. Filtro de mediana

El *filtro de mediana* reemplaza el valor del píxel por la mediana de los niveles de gris de los vecinos de ese píxel:

$$\hat{f}(x, y) = \text{mediana}_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (3.14)$$

El valor original del píxel se incluye en el cálculo de la mediana. Los filtros de mediana son muy populares porque, para ciertos tipos de ruidos aleatorios, proveen excelente capacidades de reducción de ruidos, con considerablemente menos borrosidad que los filtros lineales de tamaño similar. El filtro de mediana es un filtro de suavizado que preserva bordes, es decir, puede remover ruido aislado sin desenfocar mucho la imagen [Liu y Mason, 2009; Richards y Jia, 2006; Chuvieco, 2010]. Este filtro es particularmente efectivo en la presencia de ruido de disparo (valores de píxel que no tienen relación con la escena de la imagen) [Jensen, 2005] y ruidos impulsivos bipolares y unipolares [Gonzalez y Woods, 2002; Richards y Jia, 2006]. Sin embargo, en una ventana pequeña donde los píxeles son más o menos homogéneos, solamente una reducida porción de estos son ruidosos, los cuales tienden a tener valores extremos en uno y otro rango. Es por ello que muchas veces este ruido no es eficientemente eliminado por el filtro de mediana [Sossa Azuela y Rodriguez Morales, 2011].

El filtro de mediana tiene ciertas ventajas con respecto a los filtros de convolución ponderado [Jensen, 2005], incluyendo que no desplaza límites y que la degradación mínima que produce a los bordes le permite ser aplicado reiteradamente. Aunque como desventaja suele perder detalles como puntos y líneas finas, y redondear las esquinas de los objetos.

3.2.3.2. Filtros de mínima y máxima

El *filtro de máxima* está dado por

$$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (3.15)$$

Este filtro es útil para encontrar los puntos más brillantes en una imagen. Como el ruido pimienta tiene valores muy bajos, es reducido por este filtro como resultado del proceso de selección *max* en el área S_{xy} [Gonzalez y Woods, 2002]. Aunque tiende a aclarar la imagen.

El *filtro de mínima* es el siguiente

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (3.16)$$

Este filtro es útil para encontrar los puntos más oscuros en una imagen, por lo que reduce el ruido sal como resultado de la operación *min* [Gonzalez y Woods, 2002]. Este filtro tiende a oscurecer la imagen.

Es recomendable para ambos filtros utilizar tamaños de ventana pequeños y filtrar en forma de cascada. Para el caso del ruido aditivo uniforme, estos filtros suavizan tres veces menos que el filtro de media o el de Gauss. La combinación de ambos filtros aplicados en forma de cascada resultan efectivos para atenuar o eliminar el ruido de tipo “sal y pimienta” o el ruido impulsivo, a la vez que suele ser menos agresivo a las esquinas y bordes que el filtro de mediana. Se puede demostrar que repetidas aplicaciones de los filtros de máxima y mínimo van cambiando la imagen, se va convirtiendo en una imagen homogénea [Sossa Azuela y Rodriguez Morales, 2011].

Estos filtros son para análisis visual y no deben ser aplicados para extraer información biofísica.

3.2.3.3. Filtro olímpico

El *filtro olímpico* o *filtro de punto medio* simplemente calcula el punto medio entre los valores máximo y mínimo en el área abarcada por el filtro:

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right] \quad (3.17)$$

Este filtro trabaja mejor para ruido distribuido aleatoriamente, como ruido Gaussiano o uniforme [Gonzalez y Woods, 2002]. Este algoritmo es útil para eliminar la mayoría de los ruidos de disparo [Jensen, 2005].

Una imagen a la que se le aplica este filtro presenta una nitidez disminuida y pérdida de detalles de forma.

3.2.3.4. Filtro de media alfa recortada

Supongamos que borramos los $d/2$ valores de niveles de gris más bajos y los $d/2$ más altos de $g(s, t)$ en la vecindad S_{xy} . Sea S_{xy}^r los $mn - d$ píxeles remanentes. Un filtro formado promediando estos píxeles remanentes es llamado *filtro de media alfa recortada*:

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}^r} g(s, t) \quad (3.18)$$

donde el valor de d puede estar en el rango de 0 a $mn - 1$. Cuando $d = 0$, el filtro alfa recortado se reduce al filtro de media aritmética. Si elegimos $d = (mn - 1)$ el filtro se vuelve un filtro de mediana. Para otros valores de d , el filtro de alfa recortada es útil en situaciones que envuelven tipos múltiples de ruido, tales como una combinación de ruido de tipo “sal y pimienta” y Gaussiano [Gonzalez y Woods, 2002].

3.2.3.5. Filtro de media de los k valores más cercanos

Supongamos que elegimos los k valores de niveles de gris más cercanos a $g(x, y)$. Sea S_{xy}^r el conjunto de los k píxeles seleccionados. El *filtro de media de los k valores más cercanos* está dado por:

$$\hat{f}(x, y) = \frac{1}{k} \sum_{(s,t) \in S_{xy}^r} g(s, t) \quad (3.19)$$

donde el valor de k puede estar en el rango de 2 a mn . Cuando $k = mn$, est filtro se reduce al filtro de media aritmética. El valor típico de k es $(mn/2) + 1$.

Esta aproximación evita ND extremos, los cuales son probable que sean ruido, y asegura su eliminación. Por otro lado, si el píxel en la ventana es un píxel de borde, tomando la media de los k ND más cercanos es más posible preservar el borde. Este filtro entonces es un filtro de suavizado que preserva bordes [Liu y Mason, 2009].

3.2.3.6. Filtro de mediana de los k valores más cercanos

Supongamos que elegimos los k valores de niveles de gris más cercanos a $g(x,y)$. Sea S_{xy}^r el conjunto de los k píxeles seleccionados. El *filtro de mediana de los k valores más cercanos* está dado por:

$$\hat{f}(x,y) = \text{mediana}_{(s,t) \in S_{xy}^r} \{g(s,t)\} \quad (3.20)$$

donde el valor de k puede estar en el rango de 2 a mn . Cuando $k = mn$, est filtro se reduce al filtro de mediana. El valor típico de k es $(mn/2) + 1$.

Este filtro es más efectivo para eliminar ruido y preservar bordes que el filtro de media de los k valores más cercanos [Liu y Mason, 2009].

3.2.4. Filtros adaptativos

La máscara de filtro puede definirse dinámicamente por las estadísticas locales. Los filtros que actúan de esta forma son llamados *filtros adaptativos*. El filtrado adaptativo puede ser usado tanto para resaltado de contraste, resaltado de bordes y suavizado preservando los bordes como para eliminación de ruido [Liu y Mason, 2009].

Los filtros adaptativos son capaces de un rendimiento superior. El precio a pagar por mejorar el poder de filtrado es un incremento en la complejidad del filtro.

3.2.4.1. Filtro de reducción de ruido local

Nuestro filtro operará en una región local S_{xy} . La respuesta del filtro en cualquier punto (x,y) en el cual la región está centrada debe estar basada en cuatro cantidades: (a) $g(x,y)$, el valor de la imagen ruidosa en (x,y) ; (b) σ_{η}^2 , la varianza del ruido corrompiendo $f(x,y)$ para formar $g(x,y)$; (c) m_L , la media local de los píxeles en S_{xy} ; (d) σ_L^2 , la varianza local de los píxeles en S_{xy} . Queremos que el comportamiento del filtro sea el siguiente:

1. Si σ_{η}^2 es cero, el filtro debe retornar simplemente el valor de $g(x,y)$. Esto es lo trivial, el caso ruido cero en el cual $g(x,y)$ es igual a $f(x,y)$.
2. Si la varianza local es alta relativa a σ_{η}^2 , el filtro debe devolver un valor cercano a $g(x,y)$. Una varianza local alta típicamente está asociada con sus bordes, y esto se debe preservar.
3. Si las dos varianzas son iguales, queremos que el filtro devuelva el valor de la media aritmética del píxel en S_{xy} . Esta condición ocurre cuando el área local tiene las mismas propiedades que la imagen global, y el ruido local debe ser reducido simplemente promediando.

Una expresión adaptativa para obtener $\hat{f}(x, y)$ basada en estas suposiciones puede ser escrita como

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_{\eta}^2}{\sigma_L^2} [g(x, y) - m_L] \quad (3.21)$$

La única cantidad que necesita ser conocida o estimada es la varianza del ruido global, σ_{η}^2 . Los otros parámetros son calculados desde los píxeles en S_{xy} en cada ubicación (x, y) en la cual la ventana está centrada. Un supuesto tácito en la expresión (3.21) es que $\sigma_{\eta}^2 \leq \sigma_L^2$. El ruido en nuestro modelo es aditivo e independiente de la posición, por lo que es un supuesto razonable para hacer ya que S_{xy} es un subconjunto de $g(x, y)$. Sin embargo, rara vez tenemos exacto conocimiento de σ_{η}^2 . Por esta razón, se puede definir en una implementación de que la división se establezca en 1 si ocurre la condición $\sigma_{\eta}^2 > \sigma_L^2$. Esto lo hace un filtro no lineal.

Esta técnica puede aumentar detalles locales sutiles en una imagen pero no va a conservar la información de imagen original [Liu y Mason, 2009].

3.2.4.2. Filtro de mediana adaptativo

El filtro de mediana actúa bien mientras la densidad espacial del ruido impulsivo no es largo (como una regla de aprobación, P_a y P_b menos de 0.2) [Gonzalez y Woods, 2002]. El *filtro de mediana adaptativo* puede manejar ruido impulsivo con probabilidades mayores que esas. Un beneficio adicional del filtro de mediana adaptativo es que busca preservar detalle mientras suaviza ruido no impulsivo, algo que el filtro de mediana “tradicional” no hace. Si bien el filtro de mediana adaptativo también usa una ventana rectangular S_{xy} , ésta cambia (incrementa) su tamaño durante la operación, dependiendo de ciertas condiciones.

Consideremos la siguiente notación:

z_{min} : valor de nivel de gris mínimo en S_{xy}

z_{max} : valor de nivel de gris máximo en S_{xy}

z_{med} : mediana de los niveles de gris en S_{xy}

z_{xy} : nivel de gris en las coordenadas (x, y)

S_{max} : tamaño máximo permitido de S_{xy}

El algoritmo de filtrado de mediana adaptativo trabaja como se describe en Algoritmo 1. La clave para entender el mecanismo de este algoritmo es mantener en mente que tiene tres propósitos principales: eliminar ruido sal y pimienta (impulsivo), suavizar otros ruidos que pueden no ser impulsivos, y reducir distorsiones, tales como afinamiento o ensanchamiento excesivo de bordes de objetos. Los valores z_{min} y z_{max} son considerados estáticamente por el algoritmo por ser componentes de ruido “impulsivos”, aun si estos no son los valores de píxel más altos o más bajos posibles en la imagen.

Con estas observaciones en mente, podemos ver que el propósito principal es determinar si la salida del filtro de mediana, z_{med} , es un impulso (negro o blanco) o no. Si la condición $z_{min} < z_{med} < z_{max}$ se mantiene, entonces z_{med} no puede ser un impulso por la razón mencionada en el párrafo anterior. En este caso, probamos ver si el punto en el centro de la ventana, z_{xy} , es un impulso en sí mismo. Si la condición $z_{min} < z_{xy} < z_{max}$ es verdadera, entonces z_{xy} no puede ser un impulso por la misma razón que z_{med} no lo era. En este caso, el algoritmo devuelve el valor de píxel sin cambiar, z_{xy} . Evitando

Algoritmo 1 Filtro de mediana adaptativo

```
function FILTRO( $z_{xy}$ , tamañoVentana)
  if  $z_{med} > z_{min}$  and  $z_{med} < z_{max}$  then
    if  $z_{xy} > z_{min}$  and  $z_{xy} < z_{max}$  then return  $z_{xy}$ 
    else return  $z_{med}$ 
    end if
  else
    tamañoVentana = INCREMENTAR(tamañoVentana)
    if tamañoVentana  $\leq S_{max}$  then
      FILTRO( $z_{xy}$ , tamañoVentana)
    else return  $z_{xy}$ 
    end if
  end if
end function
```

cambiar estos puntos con “niveles intermedios”, la distorsión se reduce en la imagen. Si la condición $z_{min} < z_{xy} < z_{max}$ es falsa, entonces $z_{xy} = z_{min}$ o $z_{xy} = z_{max}$. En cualquier caso, el valor del píxel es un valor extremo y el algoritmo devuelve el valor de la mediana z_{med} , el cual sabemos que no es un impulso.

Supongamos que se encuentra un impulso. El algoritmo incrementa el tamaño de la ventana y vuelve a ejecutar el algoritmo. Este bucle continúa hasta que el algoritmo encuentra un valor de mediana que no es un impulso, o el tamaño de ventana máximo es alcanzado. Si el tamaño máximo de ventana es alcanzado, el algoritmo devuelve el valor z_{xy} . Notar que no hay garantía que este valor no sea un impulso. Mientras más chica sea la probabilidad de ruido P_a y/o P_b , o más largo S_{max} sea permitido ser, menos posible es que la condición de salida prematura ocurra.

Cada vez que el algoritmo devuelve un valor, la ventana S_{xy} se mueve a la siguiente posición en la imagen.

La *clasificación* es la fase hacia donde se orientan buena parte de los reales, correcciones y transformaciones de las imágenes, ya que la clasificación se beneficia notablemente de algunos de éstos procesos. El resultado de la clasificación digital es una cartografía y un inventario de las categorías de objetos en estudio. Una imagen multibanda se convierte en otra imagen, del mismo tamaño y características de las originales, con la diferencia que el ND que define cada píxel no tiene relación con la radiancia detectada por el sensor, sino que se trata de una etiqueta que identifica la categoría asignada a ese píxel. A partir de una imagen como ésta puede generarse cartografía temática, así como un inventario estadístico del número de píxeles -y por lo tanto de la superficie- asignados a cada categoría.

Un método de clasificación debería contar con las siguientes características [Chuvieco, 2010]:

1. Fiable.
2. Reproducible por otros, dadas las mismas variables de entrada.
3. Robusto, es decir, no sensible a pequeños cambios en las condiciones de entrada, aunque permita explorar enteramente la información de esas variables.
4. Exhaustivo, que cubra todo el territorio de estudio.
5. Objetivo, que no esté marcado por las decisiones del intérprete.

La clasificación digital no busca una definición absoluta de cada cubierta, que pudiera ser aplicable a cualquier imagen, sino más bien una caracterización relativa, válida para una determinada imagen y un territorio concreto [Chuvieco, 2010].

Tradicionalmente se han dividido los métodos de clasificación en dos grupos: supervisado y no supervisado. El método supervisado parte de un conocimiento previo del terreno, a partir del cual se seleccionan las muestras para cada una de las categorías. Por su parte, el método no supervisado procede a una búsqueda automática de grupos de valores homogéneos dentro de la imagen. Queda al usuario, en este caso, la labor de encontrar correspondencias entre esos grupos y sus categorías de interés.

Suelen distinguirse dos tipos de categorías que pueden intervenir en una clasificación temática [Chuvieco, 2010]: informacionales y espectrales. Las primeras son las que constituyen la leyenda

de trabajo que pretende deducir el intérprete. Las segundas, corresponden a los grupos de valores espectrales homogéneos dentro de la imagen, en función de ofrecer una reflectividad similar, para las bandas consideradas y en la fecha concreta de la imagen. Las categorías informacionales se establecen por el usuario, mientras las espectrales se deducen de los ND presentes en la imagen.

Idealmente habría de producirse una correspondencia biunívoca entre las dos; esto es, que cada clase informacional le corresponda un único grupo espectral y viceversa. Desgraciadamente este caso es muy poco frecuente, produciéndose, normalmente, alguna de las siguientes situaciones:

1. Que una categoría de cubierta esté expresada en varias clases espectrales. Este problema no es muy complicado de resolver. Basta, tener en cuenta esa heterogeneidad espectral para generar dos o más grupos espectrales para cada clase informacional, fusionándolas después de la clasificación. Esto evita contar con clases muy heterogéneas espectralmente, fácilmente sujetas por tanto a confusión con otras similares.
2. Otra cosa sería que dos o más categorías informacionales compartan una sola clase espectral. Esto implicaría, en pocas palabras, que algunas categorías de la leyenda no sean discriminables, a partir de la imagen con que se cuenta, con grave peligro de confusión. En este caso, se pueden plantear varias opciones para solucionar el problema. La más sencilla sería optar por una clave más general, evitando descender a un nivel de desagregación no abordable a partir de esos datos. En otros casos, habría que intentar otras soluciones, como incluir en la clasificación una imagen perteneciente a otra fecha del año, en donde la separación entre categorías problemáticas sea posible, o acudir a criterios no espectrales, como pueden ser los derivados de la textura o situación espacial o añadir alguna variable auxiliar.
3. Por último, puede también ocurrir que varias clases informacionales compartan clases espectrales. En este caso convendría replantear la estrategia de trabajo, o acudir a las soluciones parciales comentadas en párrafos anteriores.

Como consecuencia, parece claro que el método supervisado pretende definir clases informacionales, mientras el no supervisado tiende a identificar las clases espectrales presentes en la imagen. También puede deducirse que ninguno de los dos métodos proporciona una solución inmediata a todos los problemas que entraña una clasificación digital. La elección entre uno y otro método dependerá de los datos y medios disponibles y de las propias preferencias personales.

4.1. Clasificación supervisada

La *clasificación supervisada* es el procedimiento más frecuentemente usado para análisis cuantitativo de datos de imagen de sensado remoto [Richards y Jia, 2006]. Consiste en usar algoritmos para etiquetar los píxeles de una imagen representando tipos de cobertura de suelo particulares, o *clases de información*. Independientemente del algoritmo que se elija para la clasificación, los pasos prácticos esenciales incluyen:

1. Decidir el conjunto de tipos de cobertura de suelo en las cuales la imagen será segmentada. Estos son las clases de información.
2. Elegir píxeles representativos de cada una de los conjuntos de clases decididos. Estos píxeles son llamados *datos de entrenamiento*. Frecuentemente los píxeles de entrenamiento para una clase dada estarán en una región en común cercada por un borde. Esta región suele llamarse

campo de entrenamiento. A la etapa donde se eligen los datos de entrenamiento se la llama *fase de entrenamiento*.

3. Usar los datos de entrenamiento para estimar los parámetros del algoritmo de clasificación particular a ser usado. El conjunto de parámetros para una clase dada suele llamarse la *firma* de esa clase.
4. Usando el clasificador entrenado, etiquetar o clasificar cada píxel en la imagen en alguno de los tipos de cobertura de suelo decididos. Esta etapa se llama *fase de asignación*.
5. Producir resúmenes tabulares o mapas (de clase) temáticos que resuman los resultados de la clasificación.
6. Evaluar la exactitud del producto final usando un conjunto de datos de prueba etiquetados.

4.1.1. Fase de entrenamiento

Para que una fotografía aérea pueda ser interpretada en detalle es preciso que exista una experiencia previa, que nos permita identificar cada una de las categorías de interés por una serie de rasgos, como son tono, textura, situación o tamaño. De la misma forma, la clasificación digital se inicia caracterizando los patrones que definen en la imagen las distintas categorías objetivo. Como se trata de una clasificación basada en los valores numéricos de los píxeles esta caracterización también debe ser numérica; esto es, se trata de obtener el rango de ND que identifica a cada categoría, para todas las bandas que intervienen en la clasificación.

Diversos factores introducen una cierta dispersión en torno al comportamiento espectral medio de cada cubierta. Esto implica que las distintas categorías se definen por un conjunto de ND, más o menos próximos entre sí. En términos de su clasificación digital, esto supone que existe una cierta dispersión en torno al ND medio de cada categoría, más o menos significativa según las clases que se consideren y la propia complejidad del paisaje donde estas se asienten. Por ello, la fase de entrenamiento resulta compleja: se trata de definir con rigor cada una de las categorías que pretendan discriminarse, teniendo en cuenta su propia variabilidad en la zona de estudio.

Este objetivo se logra seleccionando una muestra de píxeles de la imagen, que representen adecuadamente a las categorías de interés. A partir de esos píxeles pueden calcularse los ND medios y la variabilidad numérica de cada categoría, en todas las bandas que intervienen en la clasificación. A igual que en cualquier otro muestreo, el objetivo de esta fase es obtener los resultados más precisos con el mínimo costo. En otras palabras, identificar debidamente las distintas categorías, reduciendo a lo imprescindible los trabajos de campo. Al igual que en otras aplicaciones de muestreo, las estimaciones posteriores se basan sobre la muestra seleccionada, por lo que una incorrecta selección de ésta conducirá inexorablemente a pobres resultados en la clasificación posterior. Varios autores han comprobado que los resultados de la clasificación están mucho más influidos por la definición previa de las categorías, que por el criterio con el que éstas son posteriormente discriminadas [Chuvienco, 2010]. En definitiva, la fase de entrenamiento constituye la columna vertebral de la clasificación numérica.

4.1.2. Fase de asignación

En esta fase se trata de asignar cada uno de los píxeles de la imagen a una de las clases previamente seleccionadas. Esta asignación se realiza en función de los ND de cada píxel, para cada una de las bandas que intervienen en el proceso. Fruto de esta fase será una nueva imagen, cuyos ND expresen la categoría temática a la que se ha asignado cada uno de los píxeles de la imagen original.

Desde un punto de vista estadístico, todos ellos definen un área de dominio, en torno al centro de cada categoría a discriminar, mediante un conjunto de funciones discriminantes. Estas ecuaciones pueden considerarse, por tanto, como las fronteras que definen cada categoría. El píxel será asignado a una clase i , si sus ND se encuentran dentro del área de dominio señalado para dicha clase.

Los criterios más comunes para establecer estas fronteras estadísticas entre clases son:

1. Mínima distancia, por el cual el píxel se asigna a la clase más cercana espectralmente.
2. Paralelepípedos, que permite señalar al usuario unos umbrales de dispersión espectral asociados a cada clase.
3. Máxima probabilidad, en donde el píxel se asigna a aquella clase con la que posee mayor probabilidad de pertenencia.

Frecuentemente se suelen agrupar en métodos no paramétricos, los dos primeros, y paramétricos, el último, en razón de si asumen o no que los ND en cada categoría se distribuyen normalmente.

4.1.2.1. Clasificación de máxima probabilidad

La *clasificación de máxima probabilidad* asume que los ND en el seno de cada clase se ajustan a una distribución normal. Esto es una suposición, más que una propiedad demostrable de clases espectrales [Richards y Jia, 2006]; sin embargo, nos permite describir esa categoría por una función gaussiana, a partir de su vector de medias y matriz de varianza-covarianza. Esto nos permite calcular la probabilidad de que un píxel pertenezca a esa categoría, conociendo sus valores en las distintas bandas que empleamos para clasificar. El cálculo se realiza para todas las categorías que intervienen en la clasificación, asignando el píxel a aquella que maximice la función de probabilidad. Este criterio de clasificación es algo más complejo que los que se verán posteriormente y requiere mayor volumen de cálculo, pero es el más empleado en teledetección [Chuvieco, 2010; Richards y Jia, 2006], por su robustez y por ajustarse con más rigor a la disposición original de datos.

Para la clasificación de categorías temáticas es preciso estimar esa distribución de probabilidad para cada categoría, a partir de las medidas realizadas sobre las áreas de entrenamiento. Con este fin, se considera que los ND en el seno de cada clase se distribuyen normalmente, por lo que basta conocer la media y la desviación estandar de cada categoría para calcular la probabilidad asociada con un determinado ND. En el caso de trabajar con una sola banda, podríamos calcular esa probabilidad de acuerdo a:

$$p(x/A) = \frac{1}{\sqrt{2\pi\sigma_A^2}} e^{-(ND_x - \overline{ND}_A)^2 / 2\sigma_A^2} \quad (4.1)$$

Esto es, la probabilidad de que un píxel x , definido por un determinado ND_x , pertenezca a la clase A , puede calcularse a partir de la media (\overline{ND}_A) y la varianza (σ_A^2) de esa clase. Una vez determinada la probabilidad de pertenencia al resto de las categorías que intervienen en la clasificación, ese píxel será asignado a la clase A sólo si:

$$p(x/A) \geq p(x/B) \quad \forall A \neq B \quad (4.2)$$

variando B desde 1 hasta c , siendo c el número de categorías.

Con el propósito de agilizar el cálculo, puede transformarse la fórmula anterior en una función lineal, mediante el empleo de logaritmos:

$$\ln p(x/A) = -0,5\ln(2\pi) - 0,5\ln\sigma_A^2 - \{(ND_x - \overline{ND_A})^2/2\sigma_A^2\} \quad (4.3)$$

La expresión puede simplificarse aún más si eliminamos el primer término ($-0,5\ln(2\pi)$), por cuanto nos interesa sólo un cálculo de la probabilidad relativo a las distintas clases, y ese valor es una constante para todas ellas. Por la misma razón no es necesario reconvertir a antilogaritmos el valor calculado, ya que si el logaritmo de la probabilidad es mayor en una categoría que en otra, también lo será su antilogaritmo. Por último, el segundo término ($-0,5\ln\sigma_A^2$) puede calcularse inicialmente para cada categoría, quedando así reducido el cálculo, para cada píxel, a la última expresión.

Algunas de las categorías cuentan con mayor extensión superficial, y, por tanto, presentan mayor probabilidad de albergar píxeles desconocidos. Al considerar la *probabilidad a priori* de cada categoría, $p(A)$, se introduce un nuevo factor de criterio de decisión enunciado en (4.2). De esta forma, un píxel pertenece a la clase A , si:

$$p(x/A)p(A) \geq p(x/B)p(B) \quad \forall A \neq B \quad (4.4)$$

donde $p(x/A)$ indica la probabilidad de que un píxel x con un ND determinado, pertenezca a la clase A , y $p(A)$ la probabilidad a priori de que un píxel cualquiera se incluya en esa clase. Introduciendo esta opción el punto de corte entre las distribuciones de probabilidad, se desplaza hacia un lado u otro del original, en función del valor de la probabilidad a priori.

Con ser de gran interés, el empleo de este parámetro resulta bastante complejo, especialmente para establecer un criterio riguroso que permita estimar los valores de probabilidad a priori para cada clase. Un posible criterio sería conceder a cada categoría una probabilidad equivalente a la superficie que ocupa en la zona de estudio. Esto requiere, naturalmente, contar con un inventario previo, que raras veces está disponible. En este caso, se asume que las probabilidades a priori de todas las categorías son iguales; como resultado se puede obviar el uso de esta probabilidad.

La fórmula de probabilidad normal utilizada anteriormente puede ser extendida para que la clasificación pueda aplicarse a más de una banda, de la siguiente forma:

$$p(x/A) = (2\pi)^{-m/2} |V_A|^{-0,5} e^{-0,5(M_x - M_A)^T V_A^{-1} (M_x - M_A)} \quad (4.5)$$

donde m indica el número de bandas, $|V_A|$ la determinante de la matriz de varianza-covarianza para la clase A ; M_x el vector de ND asociado con el píxel x ; M_A , el vector de ND medios de la categoría A , y los superíndices T y -1 , la transpuesta y la inversa de las matrices a las que afectan, respectivamente. Esta fórmula también puede simplificarse mediante el uso de logaritmos:

$$\ln p(x/A) = \ln V_A - (M_x - M_A)^T V_A^{-1} (M_x - M_A) \quad (4.6)$$

una vez suprimidos los términos constantes de la fórmula anterior.

Una vez que se culmine la asignación de los píxeles de la imagen, es posible establecer unos umbrales de probabilidad para cada categoría, de tal forma que se rechacen aquellos píxeles con una probabilidad de pertenencia muy baja. Sin embargo, aplicar este criterio supone la aparición de píxeles no clasificados.

El criterio de máxima probabilidad asume que las categorías presentan una distribución normal. No obstante, aún en el caso de que las bandas de entrada no cumplan este criterio, se ha mostrado que los resultados siguen siendo bastante consistentes [Chuvieco, 2010].

4.1.2.2. Clasificación de mínima distancia

El criterio más sencillo para asignar un píxel a una de las categorías consiste en incluirlo en la más cercana espectralmente, consecuencia de comparar los ND de cada píxel con los del centro de las distintas categorías, para todas las bandas que intervienen en el análisis.

Pueden considerarse varias formas de medir esa distancia espectral entre píxeles y clases. La más empleada, en este sentido es la distancia euclidiana [Chuvieco, 2010; Richards y Jia, 2006]:

$$d_{x,A} = \sqrt{\sum_{i=1,m} (ND_{x,i} - \overline{ND}_{A,i})^2} \quad (4.7)$$

donde $d_{x,A}$ es la medida de la distancia entre el píxel x y la categoría A ; $ND_{x,i}$ el valor correspondiente al píxel a clasificar en la banda i , y el valor medio de esa categoría en la misma banda. Tras calcular esta medida para todas las categorías, puede obtenerse el valor $d_{x,A}$ más pequeño, asignando el píxel x a esa categoría. Como se trata de realizar comparaciones entre distancias, la fórmula anterior puede simplificarse eliminando la raíz cuadrada, ya que si un píxel es más cercano a una categoría por la distancia lineal (hipotenusa de sus valores espectrales), también lo será por la suma del cuadrado de sus catetos.

En resumen, el *clasificador de mínima distancia* los datos de entrenamiento son usados solo para determinar las medias de las clases; la clasificación es luego realizada poniendo un píxel en la clase de la media más cercana.

Este algoritmo es bastante sencillo y rápido de ejecutar [Chuvieco, 2010; Richards y Jia, 2006], y puede ofrecer buenos resultados cuando no se observe un gran solape entre categorías. Puesto que siempre existe una clase más cercana, no deja lugar a píxeles sin clasificar, incluso aunque éstos se alejen notablemente del centro de clase. Esto puede provocar errores de comisión importantes, sobre todo cuando no se incluya en la leyenda alguna categoría claramente presente en la imagen. Este clasificador es de gran valor cuando el número de muestras de entrenamiento es limitada [Richards y Jia, 2006].

La distancia que calcula este algoritmo es lineal; además, no considera la dispersión de cada categoría o la correlación que exista entre las bandas. Tiende, por tanto, a delimitar grupos circundantes de radio proporcional a la situación de cada categoría.

Los umbrales pueden ser aplicados a la clasificación de distancia mínima al asegurar que no sólo es un píxel más cercano a una clase de candidato, sino también de que está dentro de una distancia determinada de esa clase. Esta técnica es usada con regularidad [Richards y Jia, 2006]. A menudo, el umbral de distancia se especifica de acuerdo con un número de desviaciones estándar a partir de una media de la clase.

4.1.2.3. Clasificación de paralelepípedos

El *clasificador de paralelepípedos* es muy simple. En este método el usuario fija un área de dominio para cada categoría, teniendo en cuenta sus valores de centralidad y dispersión. Posteriormente, un píxel es asignado a dicha clase si sus ND están dentro de ese área de dominio, en todas las bandas consideradas. Esquemáticamente, el píxel x será asignado a la clase A si:

$$\overline{ND}_{A,i} - R_{A,i} \leq ND_{x,i} \leq \overline{ND}_{A,i} + R_{A,i} \quad (4.8)$$

para todo i , variando entre 1 y el número de banda. $R_{A,i}$ indica el rango de dispersión señalado para la categoría A en cada banda i .

En suma, un píxel es asignado a una determinada clase si sus ND se encuentran a menos de una determinada distancia del centro de esa clase. Esa distancia se fija por el intérprete, aunque suele identificarse con la desviación estándar, o con un múltiplo suyo. Conviene considerar que cuanto mayor sea ese umbral, mayor también el número de píxeles incluidos en esa clase, pero también más grande el riesgo de confusiones con categorías vecinas. De igual forma, un valor pequeño garantiza mayor fiabilidad en las asignaciones, pero supone un mayor número de píxeles sin clasificar.

Este criterio de asignación tiende a delinear unos polígonos multidimensionales de lados paralelos (como se muestra en la figura 4.1), de ahí su nombre, en torno al centro de cada categoría. Es un algoritmo bastante rápido de ejecutar, puesto que sólo requiere operaciones lógicas sencillas.

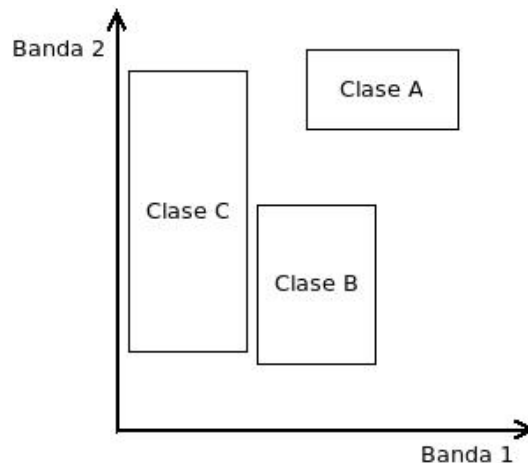


Figura 4.1: Clasificador de paralelepípedos. Un ejemplo de un conjunto de paralelepípedos bidimensionales.

Los principales problemas parten de que el diseño de áreas de dominio puede implicar, por un lado, que haya píxeles en zonas comunes a dos o más categorías, y, por otro, que algunos píxeles queden sin clasificar. Ambos problemas son ejemplificados en la figura 4.2. Respecto al primer problema, el solape entre paralelepípedos supone que un píxel pueda asignarse a dos categorías. Si el programa no cuenta con otro criterio, lo incluirá en la primera que cumpla los requisitos marcados en la ecuación (4.8). En consecuencia, algo tan intrascendente como el orden en el que se produce la asignación tiene una importante repercusión sobre los resultados.

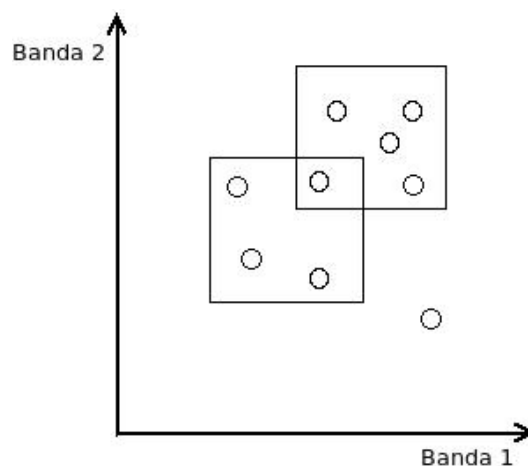


Figura 4.2: Clasificador de paralelepípedos. Posibles problemas.

Respecto a los píxeles sin clasificar, la solución pasa por ampliar paulatinamente el rango de dispersión, estableciendo un equilibrio entre los píxeles no clasificados y los solapes entre clases

vecinas. También conviene, en estos casos, repasar la definición espectral de las categorías, puesto que la presencia de píxeles sin asignación es indicio de clases espectrales mal definidas o no incluidas en la leyenda de trabajo.

4.2. Clasificación no supervisada

Los usuarios de datos de sensado remoto solo pueden especificar las clases informacionales. En cambio, la *clasificación no supervisada* se dirige a definir las clases espectrales presentes en la imagen. No requiere conocimiento del área de estudio, por lo que la intervención humana se centra más en la interpretación que en la consecución de los resultados. En esta técnica una imagen es segmentada en clases desconocidas. Es tarea del usuario etiquetar esas clases.

En esta estrategia, se asume que los ND de la imagen forman una serie de agrupaciones o conglomerados, más o menos nítidos según los casos. Estos grupos equivaldrían a píxeles con un comportamiento espectral homogéneos y, por tanto, deberían definir clases temáticas de interés. Desgraciadamente, estas categorías espectrales no siempre pueden equipararse a las clases informacionales que el usuario pretende deducir, por lo que resta a éste interpretar el significado temático de esas categorías espectrales.

4.2.1. Clasificación ISODATA

El método para definir los agrupamientos espectrales se basa en la selección de tres parámetros:

1. Variables que intervienen en el análisis.
2. Criterio para medir la similitud o distancia entre casos.
3. Criterio para agrupar casos similares.

En nuestro contexto, las variables son las bandas espectrales seleccionadas para la clasificación, ya sean originales o fruto de alguna transformación. Los casos son los píxeles que componen la imagen, cada uno de ellos definido por tantos ND como bandas intervienen en el análisis. En este espacio multivariado se trata de encontrar los grupos de píxeles con ND similares, para luego equiparlos con alguna de las clases informacionales de nuestra leyenda.

La delimitación de cada grupo espectral se inicia señalando dos criterios: uno que mida la similitud entre píxeles, y otro que marque las condiciones del proceso de agrupamiento. En cuanto al primer aspecto, se han propuesto diversos criterios para medir la distancia entre píxeles. El más utilizado se basa en la distancia Euclidiana [Chuvieco, 2010; Richards y Jia, 2006]:

$$d_{a,b} = \sqrt{\sum_{i=1,m} (ND_{a,i} - ND_{b,i})^2} \quad (4.9)$$

donde $d_{a,b}$, indica la distancia entre dos píxeles cualquiera a y b ; $ND_{a,i}$ y $ND_{b,i}$, los niveles digitales de esos píxeles en la banda i , y m el número de bandas.

En cuanto al algoritmo de agrupamiento, las opciones también son muy numerosas. Uno de los más extendido es el denominado *ISODATA*, gracias a contar con un mecanismo iterativo bastante sólido. En esencia, este algoritmo procede a partir de las siguientes fases:

1. Se señalan una serie de centros de clase, de acuerdo al número y forma indicados por el usuario. La selección de los centros de clase en esta etapa es arbitraria con la excepción que dos centros de clase no pueden ser el mismo. Para evitar la generación de clases anómalas con conjuntos de datos inusuales es recomendable generalmente espaciar las medias de las clases iniciales uniformemente sobre todos los datos.
2. Se asignan todos los píxeles de la imagen al centro de la clase más próximo, utilizando el criterio de distancia antes comentado.
3. Se calculan de nuevo los centros de clase, teniendo en cuenta los valores de todos los píxeles que se le hayan incorporado en la fase anterior.
4. Se reordenan los centros de clase en función de los parámetros de control indicados por el usuario (distancia mínima entre grupos, tamaño mínimo de un grupo, varianza máxima de un grupo y número de grupos).
5. Se vuelven a asignar todos los píxeles de la imagen al centro más cercano, el cual -debido al desplazamiento registrado en (3)- no tiene por qué coincidir con la asignación realizada en (2).
6. Si el número de píxeles que han cambiado de categoría es inferior al señalado por el usuario se termina el proceso, en caso contrario, se retorna a la fase (3).

La elección de los centros iniciales no es crítica, aunque evidentemente tendrá una influencia en el tiempo que tardará en obtener una clasificación final aceptable. Como en general no hay ninguna guía, el siguiente es un procedimiento lógico [Richards y Jia, 2006]. Los centros iniciales son elegidos espaciados uniformemente a lo largo de la diagonal multidimensional del espacio de píxel multispectral. Esta es una línea desde el origen al punto correspondiente al valor de brillo máximo en cada componente espectral. Esta elección puede ser refinada si el usuario tiene alguna idea del rango real de valores de brillo en cada componente espectral, por ejemplo habiendo computado previamente histogramas. En ese caso los centros deberían iniciarse a lo largo de una diagonal a través de las extremidades multidimensionales reales de los datos.

La elección de los centros iniciales en la manera descrita es razonable y efectiva ya que están bien esparcidas sobre el espacio multispectral en una región en la cual muchas clases espectrales ocurren, especialmente para datos correlacionados.

Desde el punto de vista estadístico, este algoritmo parece bastante adecuado para delimitar los grupos de valores homogéneos dentro de la imagen, si bien pueden darse en ocasiones bucles indefinidos, cuando los parámetros de control no están bien definidos y el algoritmo entra en una cadena de separaciones y fusiones interminables.

Obviamente, la limitación mayor de la técnica ISODATA es la necesidad de especificar el número de centros iniciales [Richards y Jia, 2006]. Si esta especificación es muy grande después una fusión a posteriori puede ser usada; sin embargo, esta es una estrategia costosa. Por el otro lado, si muy pocas son elegidas inicialmente algunas clases espectrales multimodales se obtendrán, lo cual perjudicará la precisión de la clasificación.

Otro problema, más habitual para el usuario, atañe a la elección de los parámetros de control más convenientes para la imagen que se interpreta, puesto que lo habitual será que no conozcamos de entrada el número de grupos espectrales presentes en la imagen, ni el valor idóneo de dispersión interna o de distancia entre grupos. De esta forma, el análisis no supervisado se convierte en un proceso exploratorio, en donde el intérprete establece unos valores arbitrarios que se van modificando a la vista de los resultados.

El método no supervisado también presenta como inconveniente el notable volumen de cálculo que precisa. Habitualmente, un buen agrupamiento se consigue tras un número alto de iteraciones, ya que cada píxel debe ser controlado contra los centros en cada iteración, por lo que este criterio resulta muy lento para clasificar imágenes muy extensas. Por esta razón, suele extraerse una muestra sistemática de píxeles (1 de cada 4, 8 o 16, por ejemplo) para calcular sobre ellos los centros de clase.

El análisis no supervisado se orienta no tanto a la clasificación de la imagen, como a definir precisamente los grupos o categorías que intervendrán en la clasificación propiamente dicha. En este sentido, el método no supervisado podría considerarse como exploratorio, dirigido a deducir las clases espectrales presentes en la imagen. Eso es especialmente importante cuando se adoptan criterios de clasificación que asumen una distribución normal de los datos. El análisis no supervisado podría indicarnos si efectivamente esas clases informacionales tienen un sentido espectral claro o, por el contrario, algunas de ellas presentan una distribución bimodal o multimodal (o, lo que es lo mismo, se presentan en varias categorías espectrales).

Como en el caso del método supervisado, es necesario analizar los píxeles incluidos en cada grupo para obtener sus medidas estadísticas elementales: media, desviación típica, mínimo, máximo, etc.

4.3. Verificación de resultados

Para realizar correctamente una evaluación de la exactitud (o error) de una clasificación, es necesario comparar sistemáticamente dos fuentes de información:

1. Píxeles o polígonos en un mapa de clasificación.
2. Información de prueba de referencia del suelo (la cual, de hecho, puede contener errores).

Es importante aclarar que no hay una sola medida de exactitud universalmente aceptada sino una variedad de índices, cada uno sensible a diferentes características [Jensen, 2005].

Supongamos que tenemos algunos tipos de datos de referencia de suelo verdaderos; un método usado para describir la exactitud relativa de la clasificación es la *matriz de confusión*:

$$\begin{pmatrix} C_{0\ 0} & C_{0\ 1} & \cdots & C_{0\ m-1} \\ C_{1\ 0} & C_{1\ 1} & \cdots & C_{1\ m-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m-1\ 0} & C_{m-1\ 1} & \cdots & C_{m-1\ m-1} \end{pmatrix} \quad (4.10)$$

Aquí, cada uno de los elementos, C_{ii} , en la diagonal representa el número de píxeles que están correctamente clasificados para la clase i . Cada elemento marginal (que se encuentra fuera de la diagonal), C_{ij} , representa el número de píxeles que deberían estar en la clase j pero fueron incorrectamente clasificados como de clase i . Las columnas de la matriz suelen indicar las clases de referencia de suelo, y las filas las categorías deducidas de la clasificación. Obviamente, si todos los píxeles fueron correctamente clasificados, deberíamos tener una matriz de confusión donde todos los elementos marginales tienen valor cero. La suma de todos los elementos en la matriz de confusión es el número total de píxeles de la muestra, N :

$$N = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} C_{ij} \quad (4.11)$$

La proporción entre la sumatoria de los elementos de la diagonal y el número total de píxeles representa el porcentaje de la clasificación correcta o fiabilidad global del mapa

$$\hat{F} = \frac{1}{N} \sum_{i=0}^{m-1} C_{ii} \quad (4.12)$$

Conviene tener en cuenta que la fiabilidad global puede ocultar importantes diferencias entre categorías, que escondan niveles de exactitud muy diversos [Chuvienco, 2010]. Por ello, un análisis más riguroso debe también considerar las celdas marginales. La suma de cualquier columna j de la matriz de confusión proporciona el total de píxeles que, de acuerdo a la referencia de suelo, debería estar en la clase j , N_{r_j} :

$$N_{r_j} = \sum_{i=0}^{m-1} C_{ij} \quad (4.13)$$

Entonces la proporción C_{jj}/N_{r_j} es el porcentaje de clasificación correcta de la clase j , de acuerdo con las referencias de suelo y es usualmente llamada exactitud de usuario [Liu y Mason, 2009; Richards y Jia, 2006]. Desde otro punto de vista, a los píxeles que, perteneciendo a una determinada categoría, no fueron incluidos en ella, se los denominan errores de omisión [Chuvienco, 2010; Jensen, 2005; Richards y Jia, 2006] y se calculan para cada clase como $(N_{r_j} - C_{jj})/N_{r_j}$.

La suma de cualquier fila i de la matriz de confusión proporciona el número total de píxeles que fueron clasificados como clase i , N_{C_i} :

$$N_{C_i} = \sum_{j=0}^{m-1} C_{ij} \quad (4.14)$$

Entonces la proporción C_{ii}/N_{C_i} es el porcentaje de clasificación correcta de la clase i basada en el resultado de la clasificación, y es usualmente llamada exactitud de productor [Liu y Mason, 2009; Richards y Jia, 2006]. De forma similar, los píxeles que se incluyeron en una determinada categoría perteneciendo a otra expresan errores de comisión [Chuvienco, 2010; Jensen, 2005; Richards y Jia, 2006] y se calculan como $(N_{C_i} - C_{ii})/N_{C_i}$.

Ambos errores están bastante relacionados, pero puede ocurrir que uno sea muy bajo cuando el otro es elevado. Ambas medidas son complementarias y de gran trascendencia [Chuvienco, 2010].

A parte de estas medidas de exactitud las cuales están basadas en proporciones simples, otra medida estadística comunmente usada de exactitud y calidad de clasificaciones es el *coeficiente kappa* (κ) que combina las dos estimaciones de exactitud de clase de arriba, basada en las filas y columnas de la matriz de confusión, para producir una estimación de la exactitud de la clasificación total. Este coeficiente mide la diferencia entre el acuerdo mapa-realidad observado, y se obtiene a partir de la siguiente fórmula:

$$\kappa = \frac{N \sum_{i=0}^{m-1} C_{ii} - \sum_{i=0}^{m-1} N_{r_i} N_{C_i}}{N^2 - \sum_{i=0}^{m-1} N_{r_i} N_{C_i}} \quad (4.15)$$

El valor máximo de κ es 1, indicando perfecta concordancia entre la clasificación y los datos de referencia. Un valor cercano a 0 sugiere que la concordancia observada es puramente debida al azar

[Chuvieco, 2010; Liu y Mason, 2009]. El valor mínimo de κ depende del caso [Liu y Mason, 2009], pero $\kappa < 0$ supone también una clasificación pobre, si bien depende de otros factores externos, como el propio tamaño de la muestra [Chuvieco, 2010].

A pesar del hecho de que la exactitud de clasificación derivada de la matriz de confusión no es una exactitud verdadera de clasificación, provee una medida útil. La información en una matriz de confusión es altamente dependiente de la calidad de las áreas de entrenamiento y los datos de campo. Las áreas de entrenamiento bien seleccionadas pueden mejorar tanto la exactitud de la clasificación como la credibilidad de la estimación de la exactitud. La matriz de confusión provee las bases para describir la exactitud de la clasificación y caracterizar errores, lo cual puede ayudar a refinar la clasificación o estimaciones derivadas de ésta [Jensen, 2005]. El interés de estas matrices de confusión proviene de su capacidad para plasmar conflictos entre categorías. Estas matrices proveen una medida para determinar si hace falta orientar la búsqueda a nuevos métodos de clasificación [Chuvieco, 2010]. Estrictamente hablando, este método solo proporciona una estimación de la exactitud de clasificación de la imagen completa [Liu y Mason, 2009]. El coeficiente kappa pretende evaluar si la clasificación ha discriminado las categorías de interés con precisión significativa mayor a la que se hubiera obtenido con una asignación aleatoria [Chuvieco, 2010].

En este capítulo se pretende mostrar los resultados obtenidos a lo largo de este trabajo. El software producido cumple las etapas que recomiendan las normas de la ESA en el documento “Guide to applying the ESA software engineering standards to small software projects”. La documentación de cada una de estas etapas se muestran en los anexos de este trabajo. Algunos de los documentos sufrieron algunas modificaciones durante el ciclo de desarrollo según fueron surgiendo los diferentes problemas o modificaciones basados en los conocimientos que se iban adquiriendo.

Una aclaración sobre este capítulo es que el objetivo es mostrar el efecto que producen los diferentes algoritmos sobre imágenes de teledetección. Es decir, no se intenta resaltar el valor de los datos que muestran las imágenes ni cuál es el mejor algoritmo para aplicar a una imagen particular.

Las imágenes que se mostrarán en este capítulo corresponden a los casos de prueba ejecutados correspondientes al plan de pruebas que se encuentra en el anexo E. Todos los resultados que se muestran son alteraciones que se realizan sobre la misma imagen. Para cada uno de los resultados obtenidos se muestra la imagen original junto al efecto producido por el algoritmo en cuestión. Luego, también se muestra una zona ampliada de la imagen para resaltar aún más el efecto producido.

5.1. Ruidos

La figura 5.1 compara la imagen original con la misma imagen con ruido gaussiano. Arriba a la izquierda se ve la imagen original completa, mientras que a la derecha podemos observar la imagen con ruido gaussiano. En la parte inferior de la figura encontramos un sector ampliado de ambas imágenes donde se puede observar con más detenimiento el cambio de intensidad en los píxeles producidos por el ruido. El ruido generado tiene una media igual a 0 y una desviación estándar igual a 2.

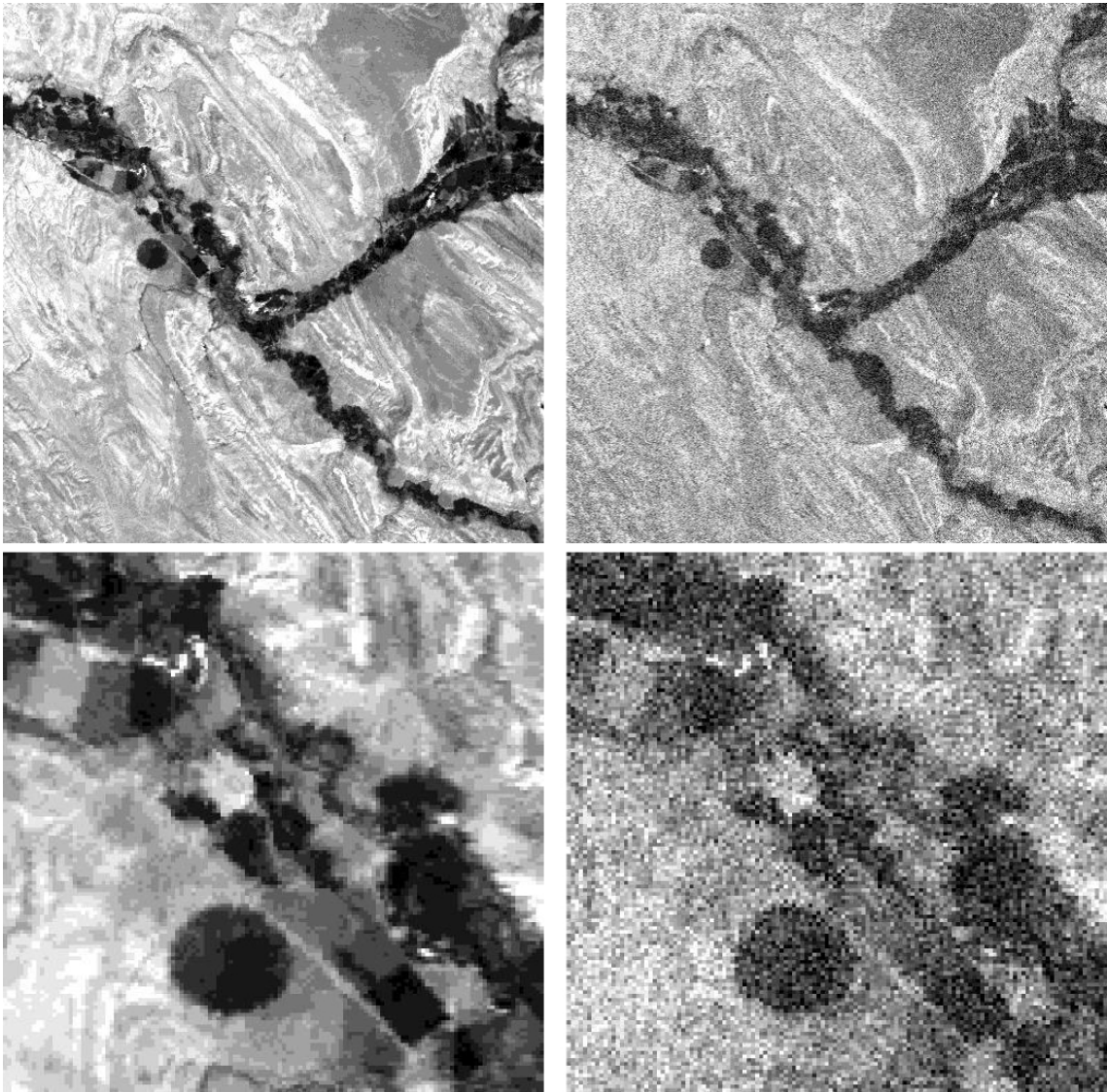


Figura 5.1: Comparación entre imagen original e imagen con ruido gaussiano

En la figura 5.2 se compara la imagen original con la misma imagen con ruido uniforme. de la misma forma que con el ruido gaussiano aquí se observa la imagen original, la que tiene ruido agregado y los respectivos zooms de cada una de las imágenes en la misma zona que antes. En este caso se puede observar como todos los píxeles aparecen con una intensidad mayor. El ruido generado tiene una distribución uniforme con un rango desde 0 a 10.

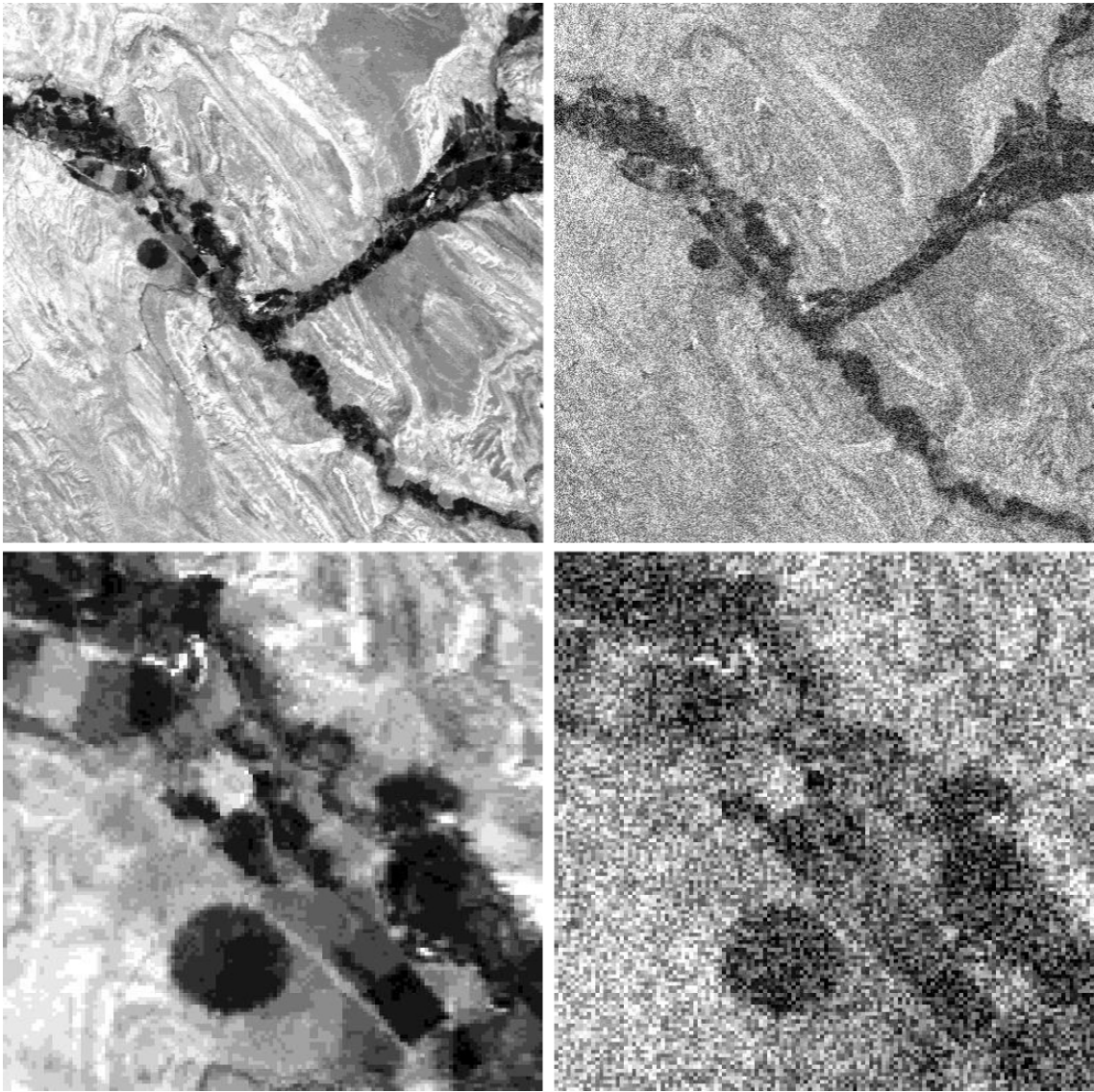


Figura 5.2: Comparación entre imagen original e imagen con ruido uniforme

La figura 5.3 muestra el efecto de generar ruido impulsivo en una imagen. Nuevamente, en parte superior de la imagen se encuentra la imagen original y la imagen con ruido. En la parte inferior encontramos los correspondientes zooms. Aquí podemos apreciar que la imagen con ruido aparece con puntos blancos y negros distribuidos por toda la imagen. El ruido generado tiene una probabilidad de ruido “sal” de 0,1 y una probabilidad de ruido “pimienta” de 0,1.

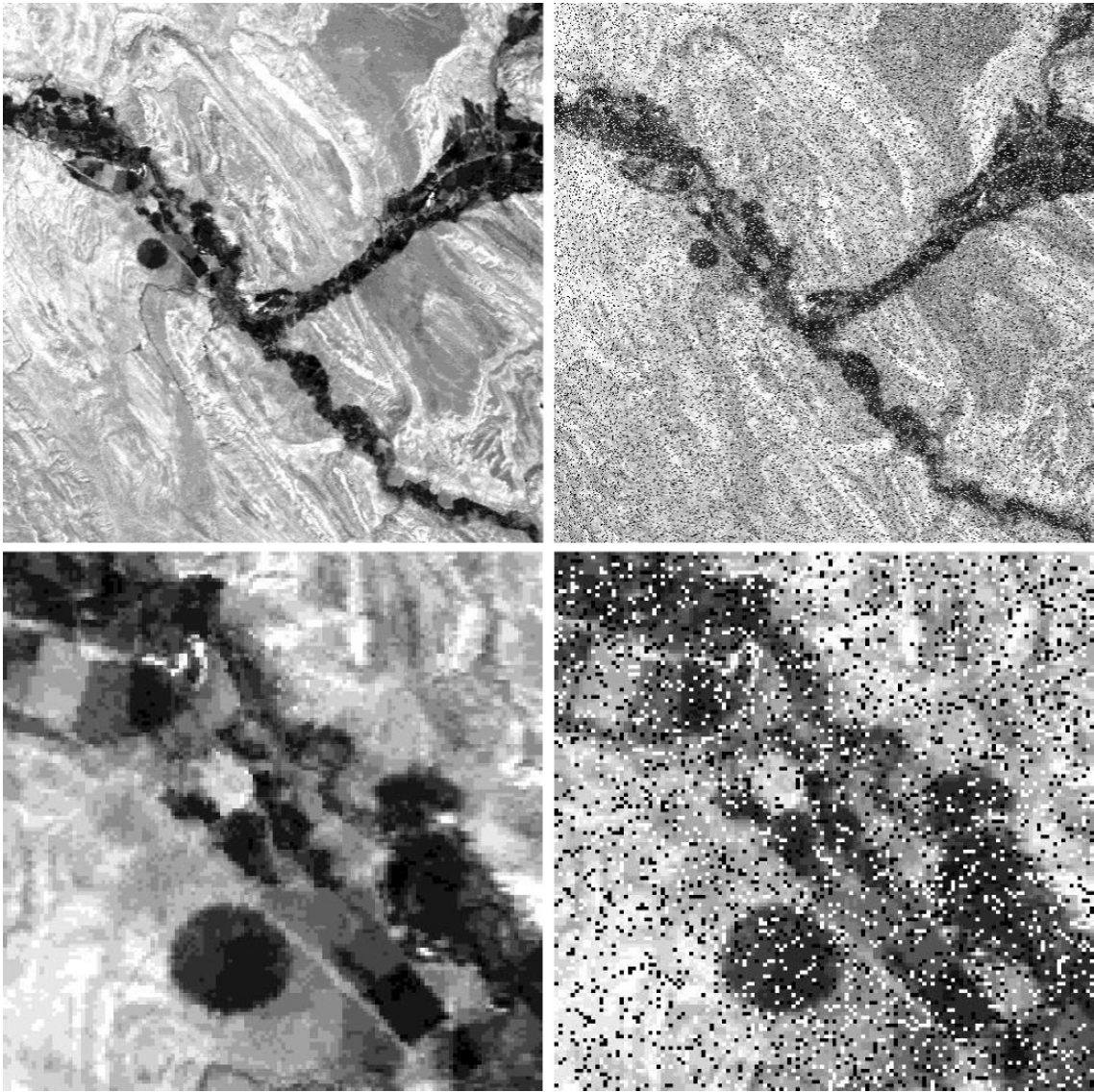


Figura 5.3: Comparación entre imagen original e imagen con ruido uniforme

5.2. Filtros lineales

Dentro de los filtros lineales, uno de los más comunes y más frecuentemente usados es el de media aritmética. Para realizar las pruebas se utilizó una imagen con ruido uniforme y luego se la filtro con este algoritmo. El resultado se puede ver en la figura 5.4 donde se compara la imagen original, la imagen con ruido uniforme y esta última con un filtro de media aritmética aplicado. El filtro utilizado tiene un tamaño de ventana igual a 3 por lado. Como se puede ver, lo que se obtuvo fue una imagen más borrosa, pero con menos ruido.

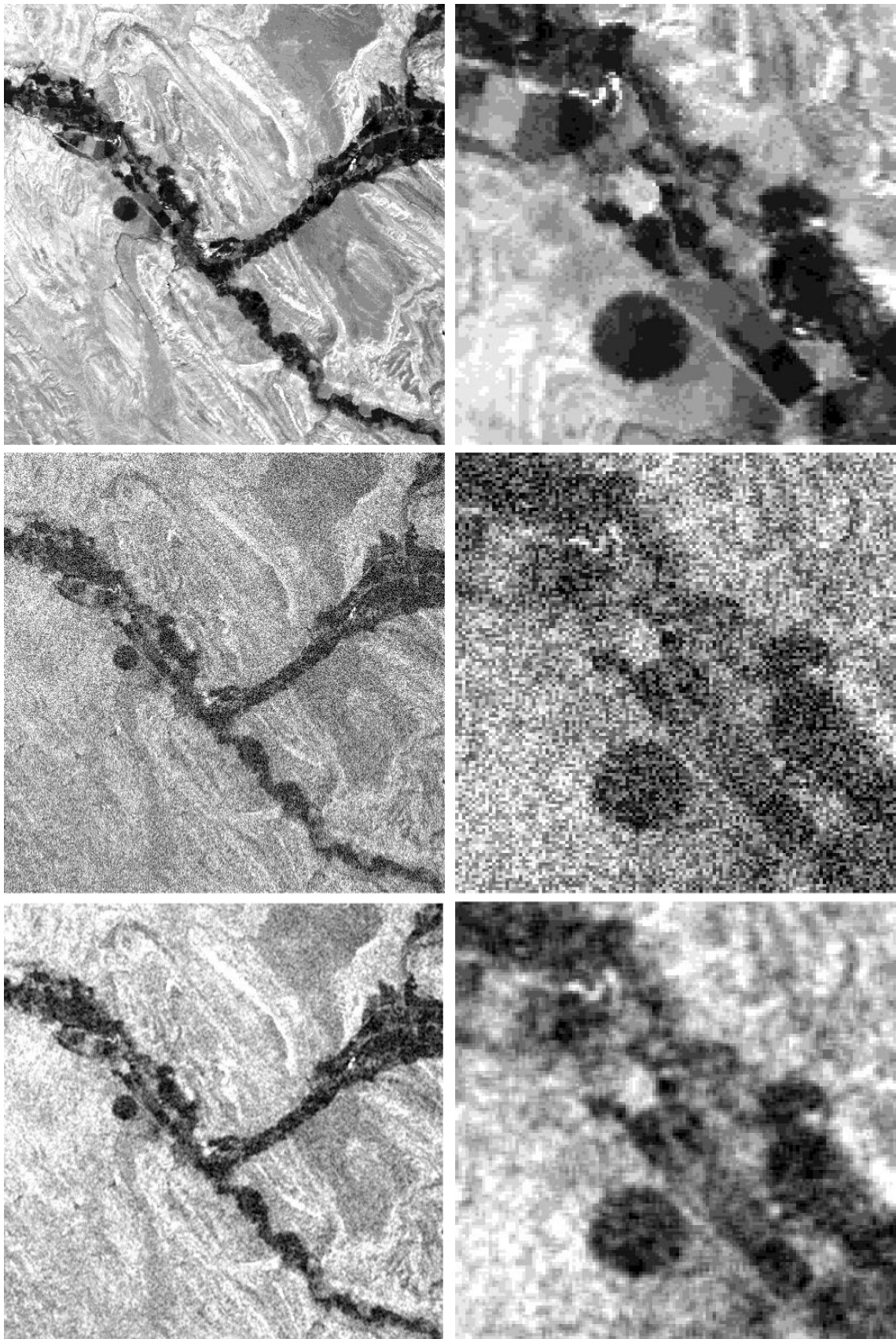


Figura 5.4: Comparación entre imagen original, imagen con ruido uniforme e imagen con ruido uniforme con filtro de media aritmética aplicado

Continuando con los filtros lineales, podemos ver el efecto de un filtro de media geométrica aplicado a una imagen con ruido gaussiano en la figura 5.5. Aquí también se puede notar una pequeña

merma en el error con el costo de la difuminación de la imagen. El filtro utilizado tiene un tamaño de ventana igual a 3 por lado.

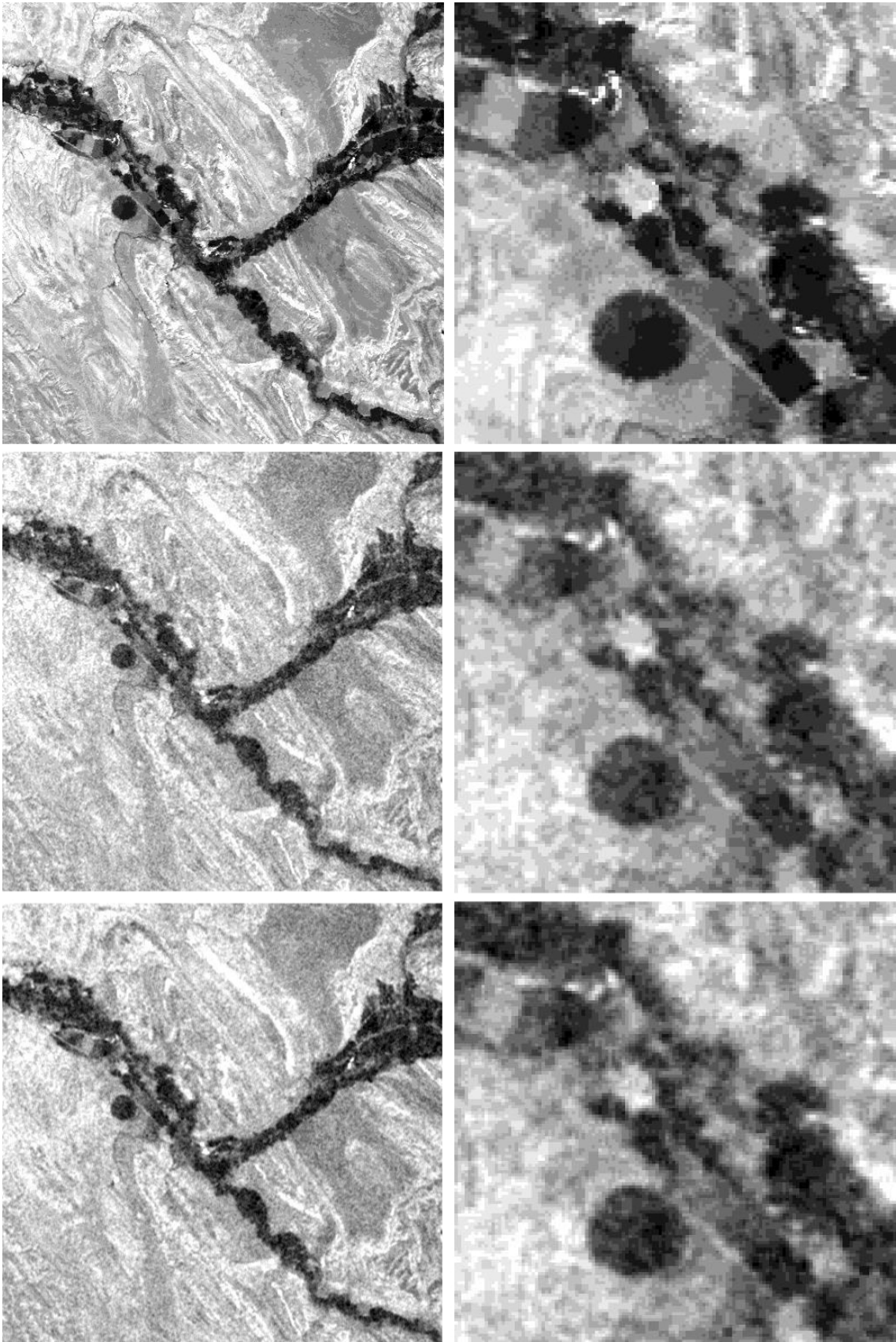


Figura 5.5: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de media geométrica aplicado

La figura 5.6 compara la imagen original, la imagen con ruido gaussiano y esta última con un filtro de media armónica aplicado. El filtro utilizado tiene un tamaño de ventana igual a 3 por lado.

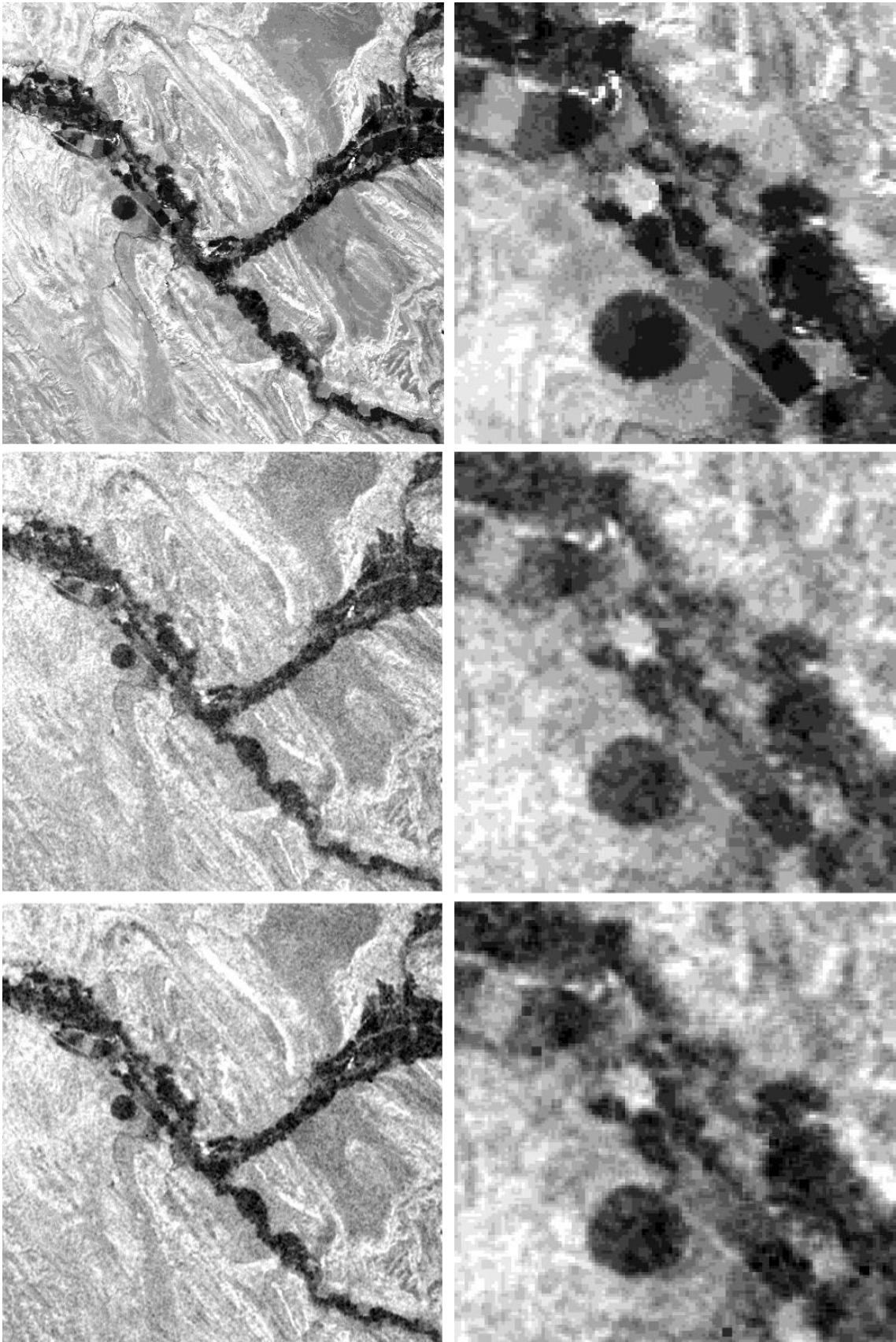


Figura 5.6: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de media armónica aplicado

En la figura 5.7 podemos ver una imagen con ruido gaussiano filtrada con un filtro de media contrarmónica aplicado. El filtro utilizado tiene un tamaño de ventana igual a 3 por lado y orden igual a -1 , lo cual lo hace equivalente al filtro de media armónica, por lo cual podemos ver que el efecto producido es el mismo.

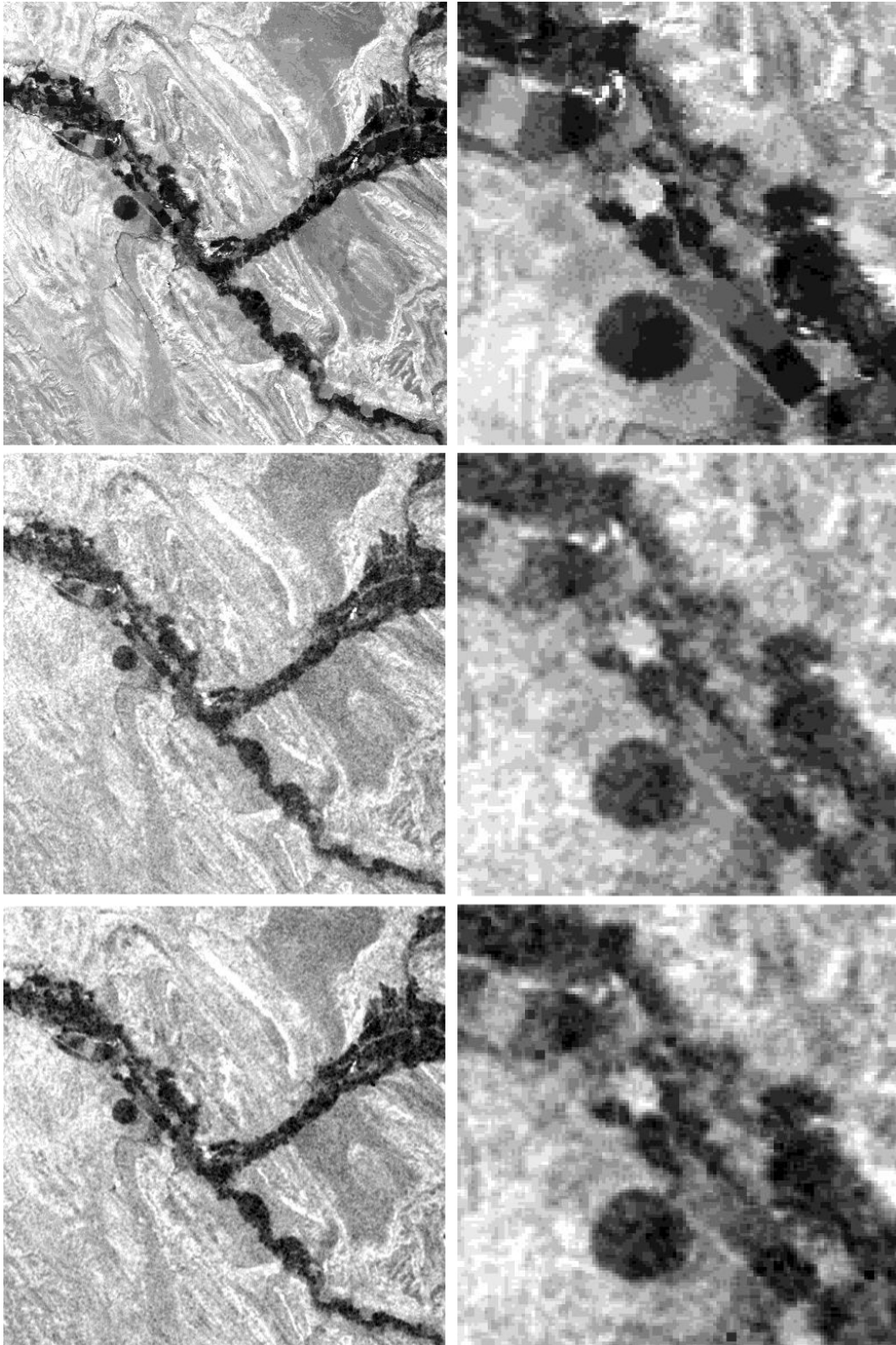


Figura 5.7: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de media contrarmónica aplicado

El último filtro lineal desarrollado es el filtro gaussiano. Para probar los efectos del mismo se utilizó una imagen con ruido gaussiano cuyo resultado se muestra en la figura 5.8. El filtro utilizado tiene un tamaño de ventana igual a 3 por lado y una varianza igual a 1,5.

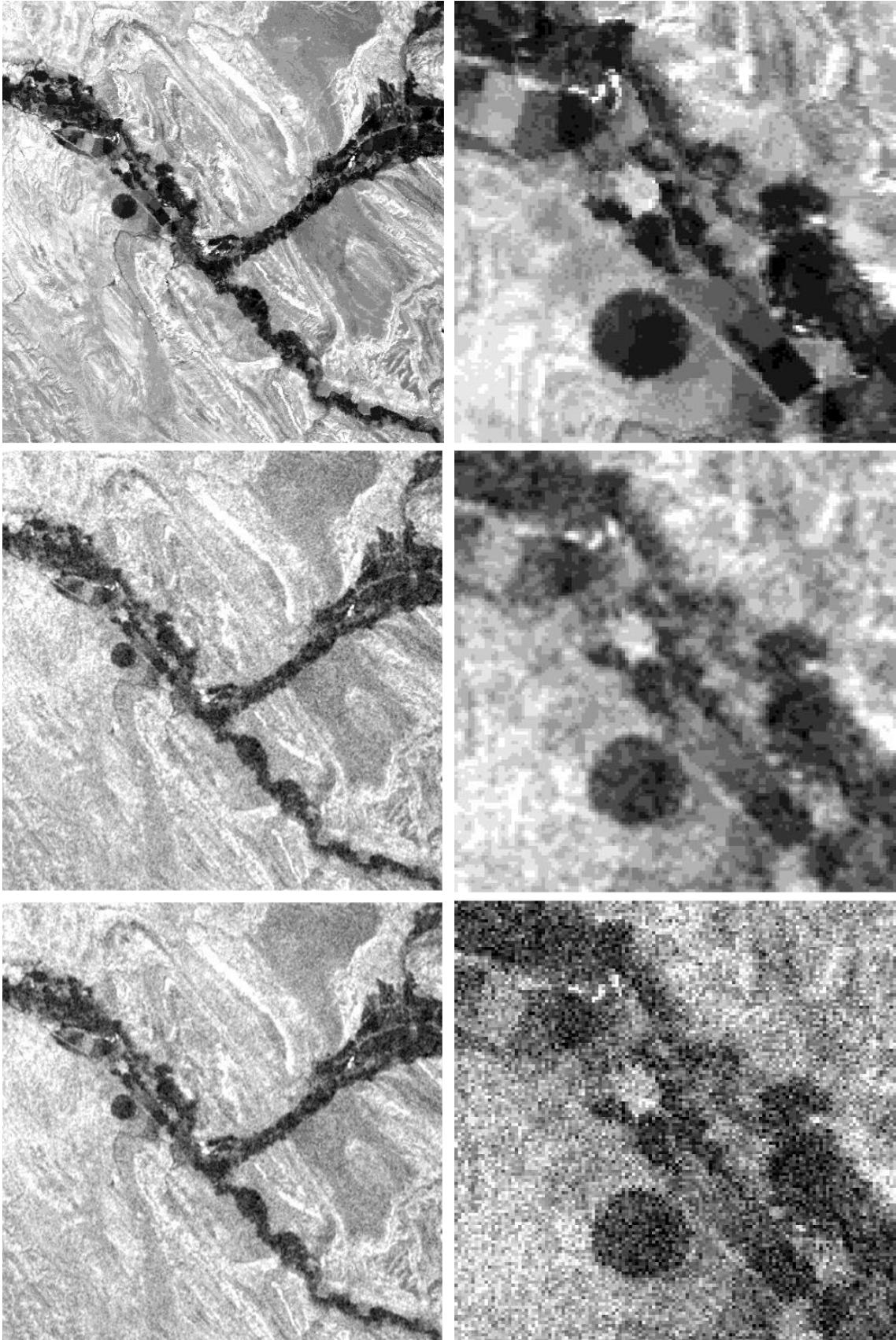


Figura 5.8: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro gaussiano aplicado

5.3. Filtros de orden

Otro tipo de filtros son los de orden. Comenzando con el filtro de mediana podemos ver la reducción del ruido impulsivo que se ha logrado utilizándolo en la figura 5.9. Aquí se puede ver que se ha logrado recuperar parte de la información original, generando relativamente poca borrosidad. El filtro utilizado tiene un tamaño de ventana igual a 3.

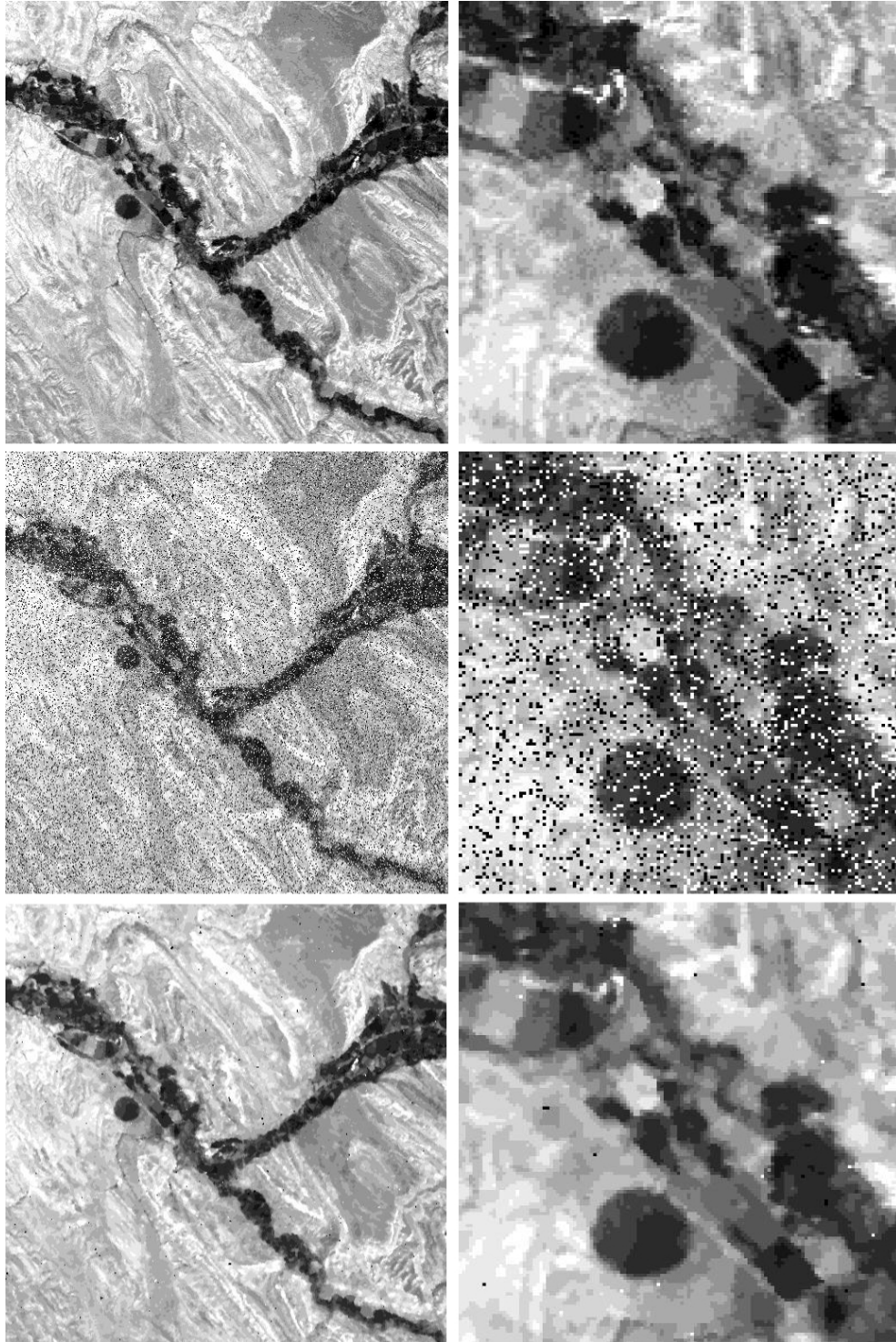


Figura 5.9: Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de mediana aplicado

La figura 5.10 muestra como el filtro de mínima no ha servido de mucho en la aplicación a una

imagen con ruido gaussiano. Aquí se puede ver como la información perdida no solo no se recuperó, sino que se acentuó aún más el ruido. El filtro utilizado tiene un tamaño de ventana igual a 3.

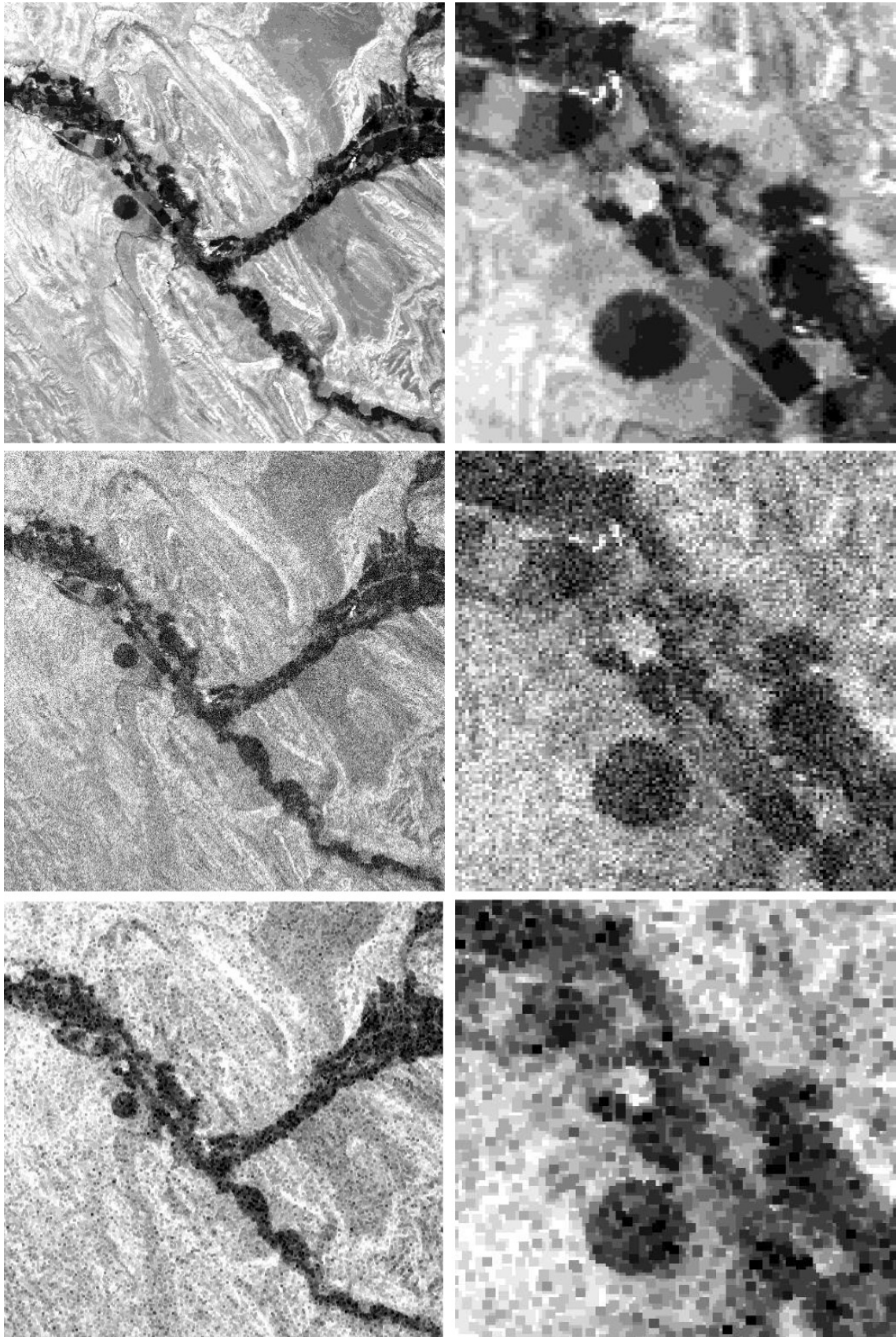


Figura 5.10: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de mínima aplicado

El mismo efecto podemos ver que produce el filtro de máxima en la figura 5.11, aunque contrario. Es decir, el efecto fue el mismo pero generando grandes porciones de píxeles claros. La imagen utilizada fue la misma que se usó con el filtro de máxima y el filtro se aplicó con un tamaño de ventana igual a 3.

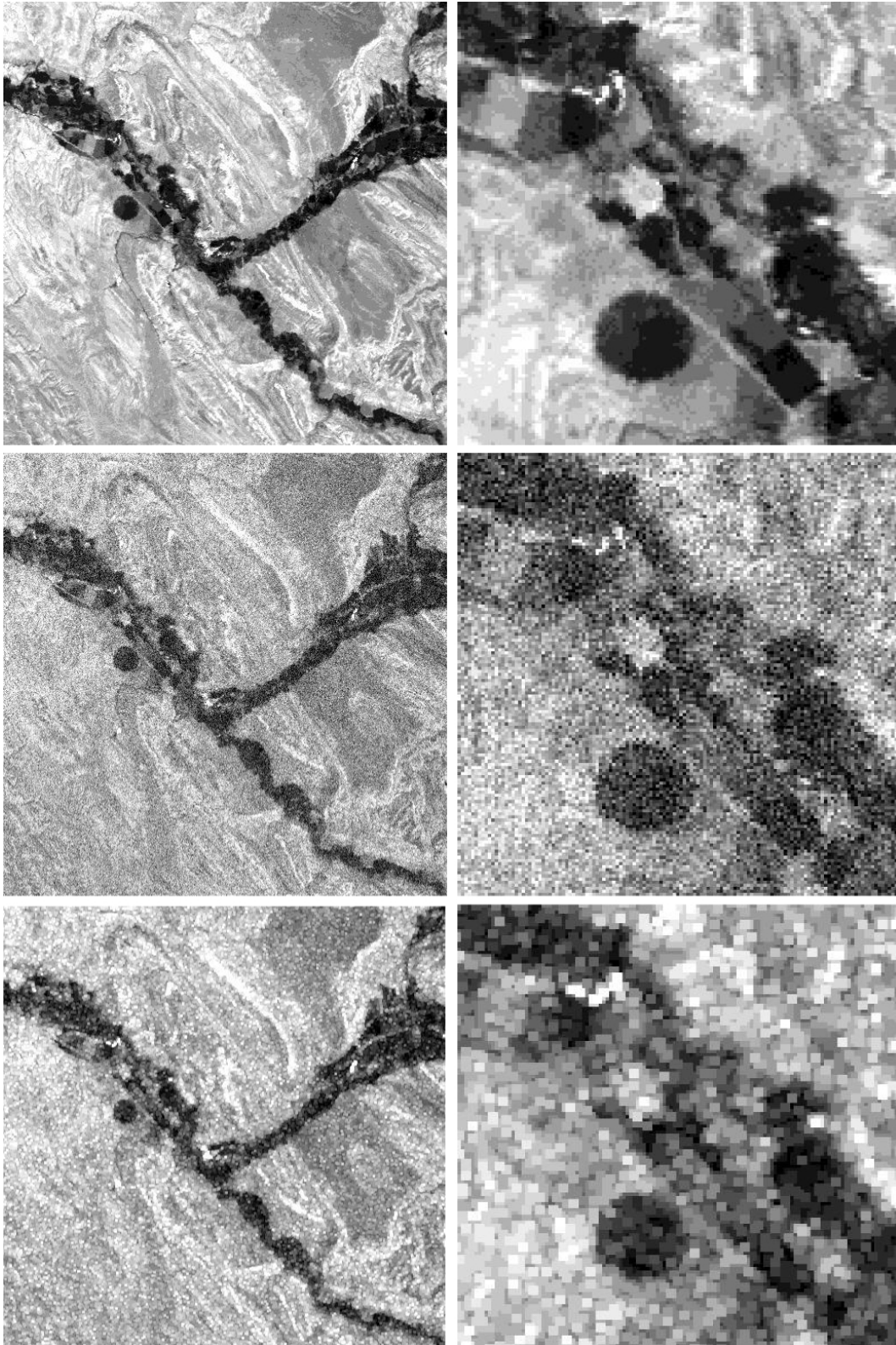


Figura 5.11: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de máxima aplicado

Otro ejemplo muy parecido a los anteriores es el filtro olímpico, cuyo efecto se muestra en la figura 5.12. Aquí vemos, como es de esperar, que el filtro produce efectos similares a los de máxima y mínima, pero con intensidades distintas. El filtro aplicado fue establecido con una ventana de tamaño igual a 3 y también se aplicó sobre una imagen con ruido gaussiano.

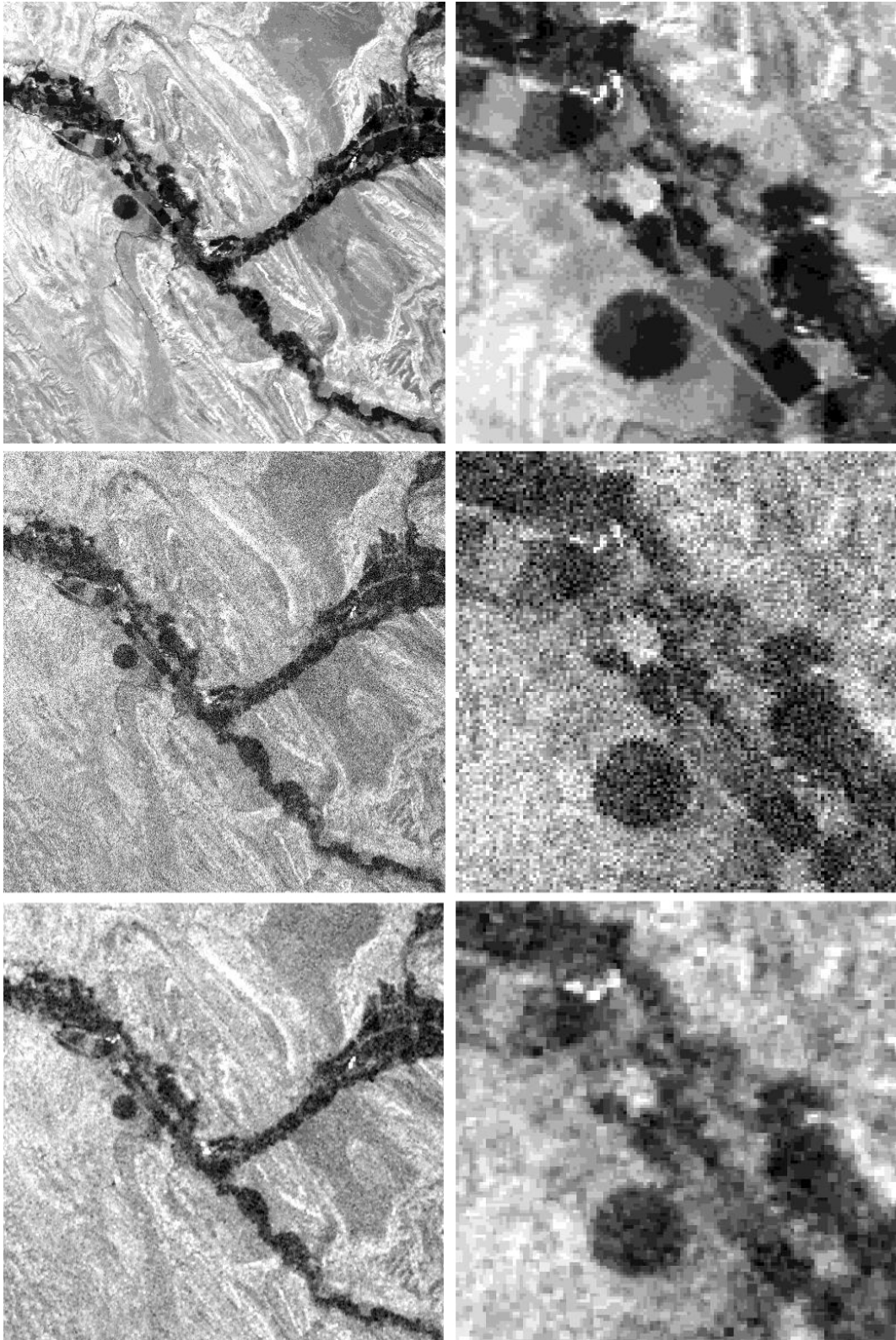


Figura 5.12: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro olímpico aplicado

La figura 5.13 compara la imagen original, la imagen con ruido impulsivo y esta última con un filtro de media alfa recortada aplicado. El filtro utilizado tiene un tamaño de ventana igual a 3 y un valor de alfa igual a 6. Aquí se puede ver como se logra recuperar la imagen en gran parte aunque quedan rastros del ruido impulsivo.

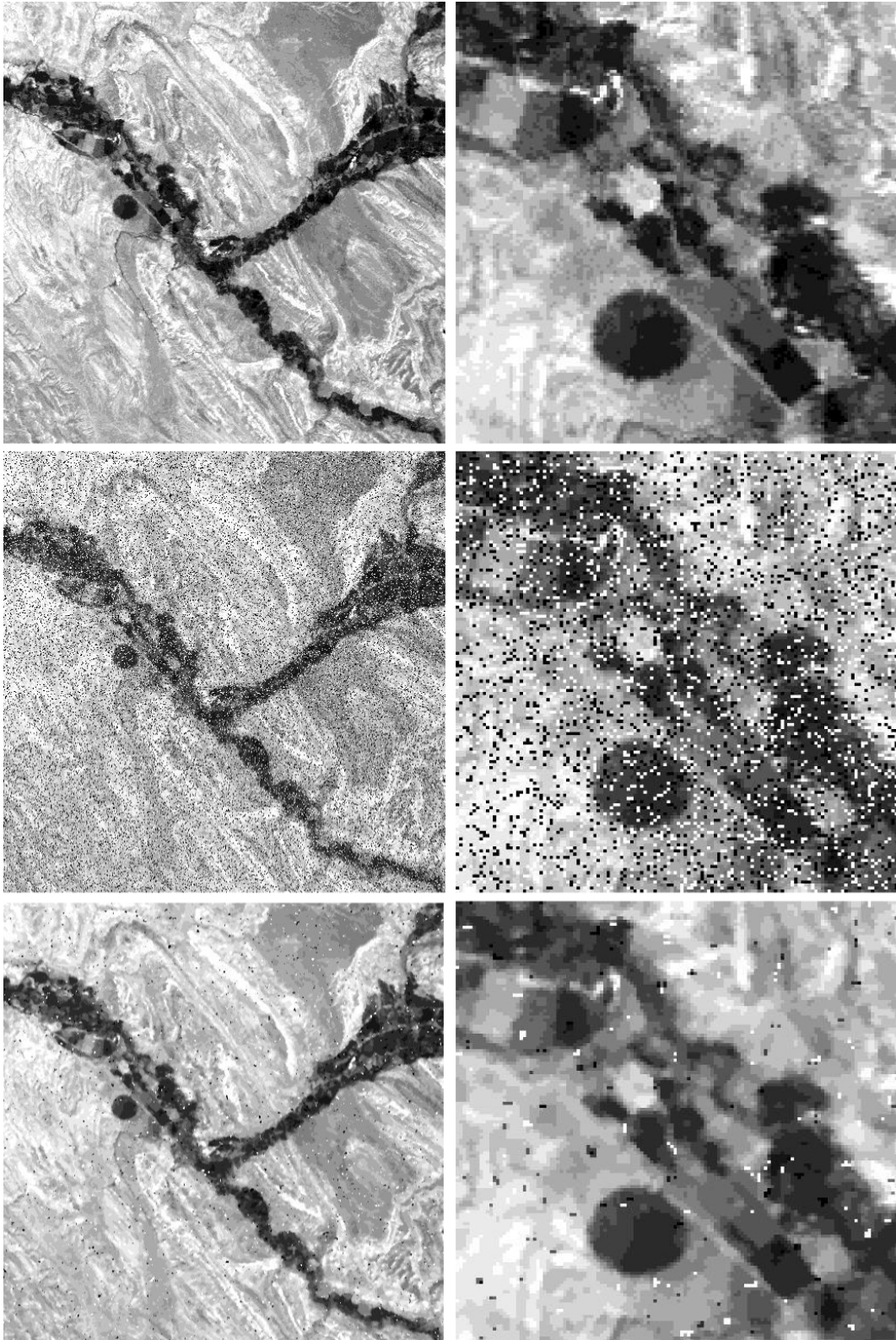


Figura 5.13: Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de media alfa recortada aplicado

Se puede ver en la figura 5.14 que el filtro de media de los k valores más cercanos no tuvo un buen desempeño. Si bien el ruido uniforme logró reducirse en pequeñas proporciones, resultado obtenido sugiere quizás utilizar otros parámetros para el filtrado u otro algoritmo para eliminar completamente el ruido. El filtro utilizado tiene un tamaño de ventana igual a 3 y un valor de k igual a 6.

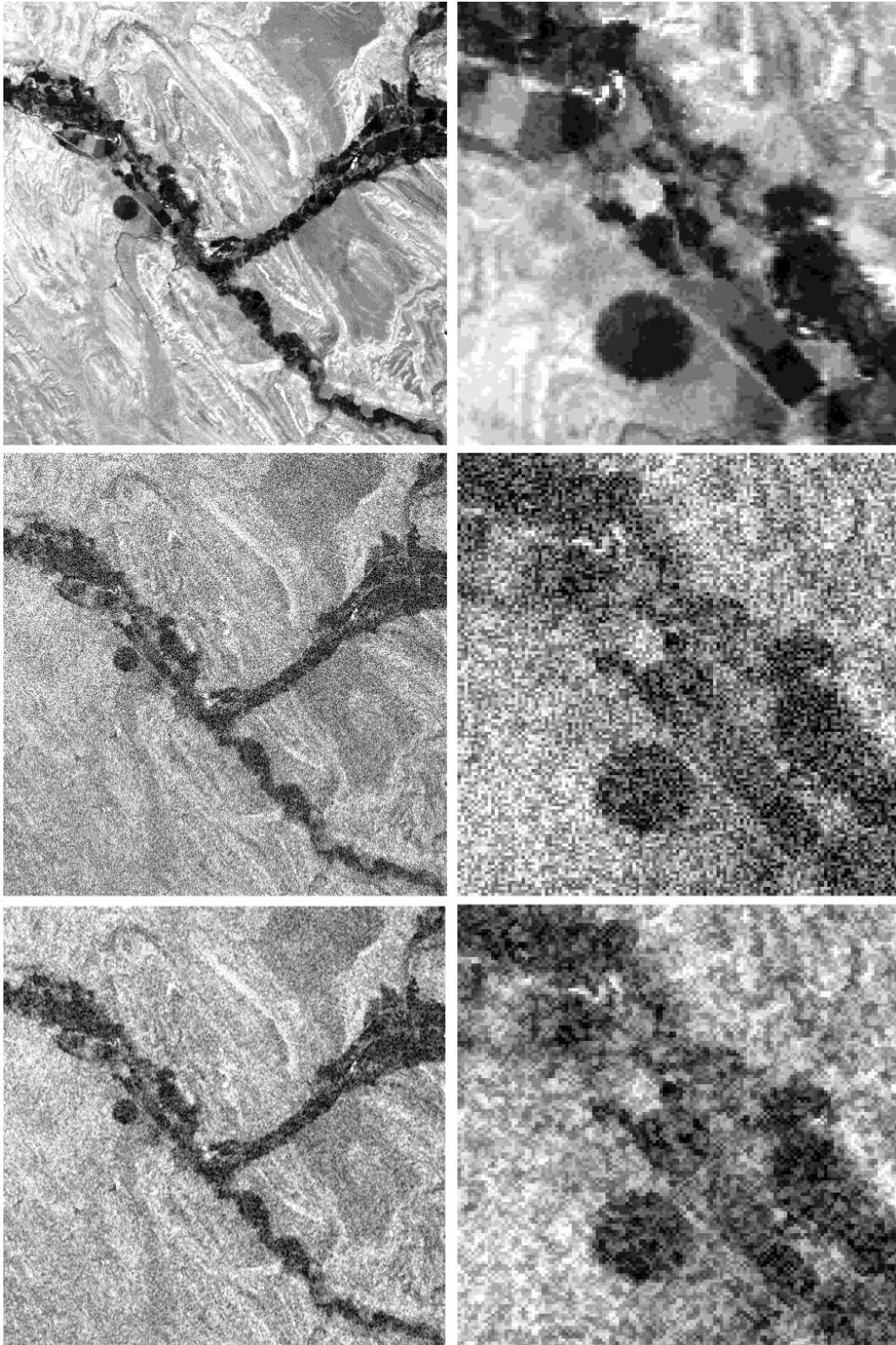


Figura 5.14: Comparación entre imagen original, imagen con ruido uniforme e imagen con ruido uniforme con filtro de media de los k valores más cercanos aplicado

En la figura 5.15 se puede ver como el filtro de mediana de los k valores más cercanos actuó de forma similar al de media alfa recortada para la misma imagen con ruido impulsivo. La diferencia grande aquí radica en que este algoritmo utilizó más datos de la ventana para el filtrado, ya que el valor de k es igual a 6 en una ventana de tamaño 3 por lado.

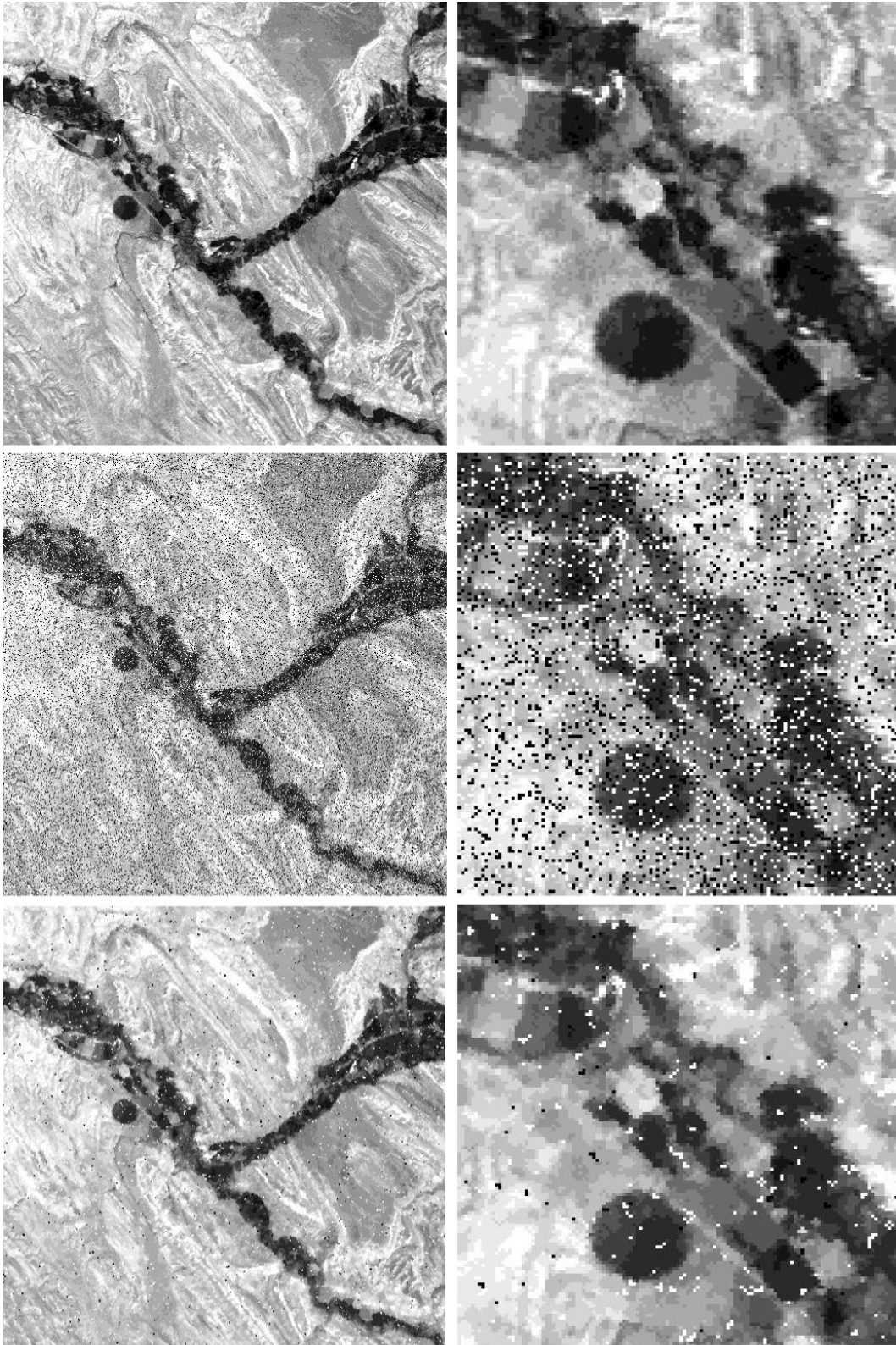


Figura 5.15: Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de mediana de los k valores más cercanos aplicado

5.4. Filtros adaptativos

En esta sección veremos el efecto de aplicar filtros adaptativos sobre imágenes con ruido. Comenzando con el filtro de reducción de ruido local, cuyo efecto sobre una imagen con ruido gaussiano se muestra en la figura 5.16. Para el filtrado se utilizaron los parámetros de tamaño de ventana igual a 3 y un valor de varianza de ruido global igual a 4. Si bien, en parte, el ruido se redujo, el efecto no ha sido el mejor, ya que quedaron rastros de ruido y la imagen se hizo más borrosa.

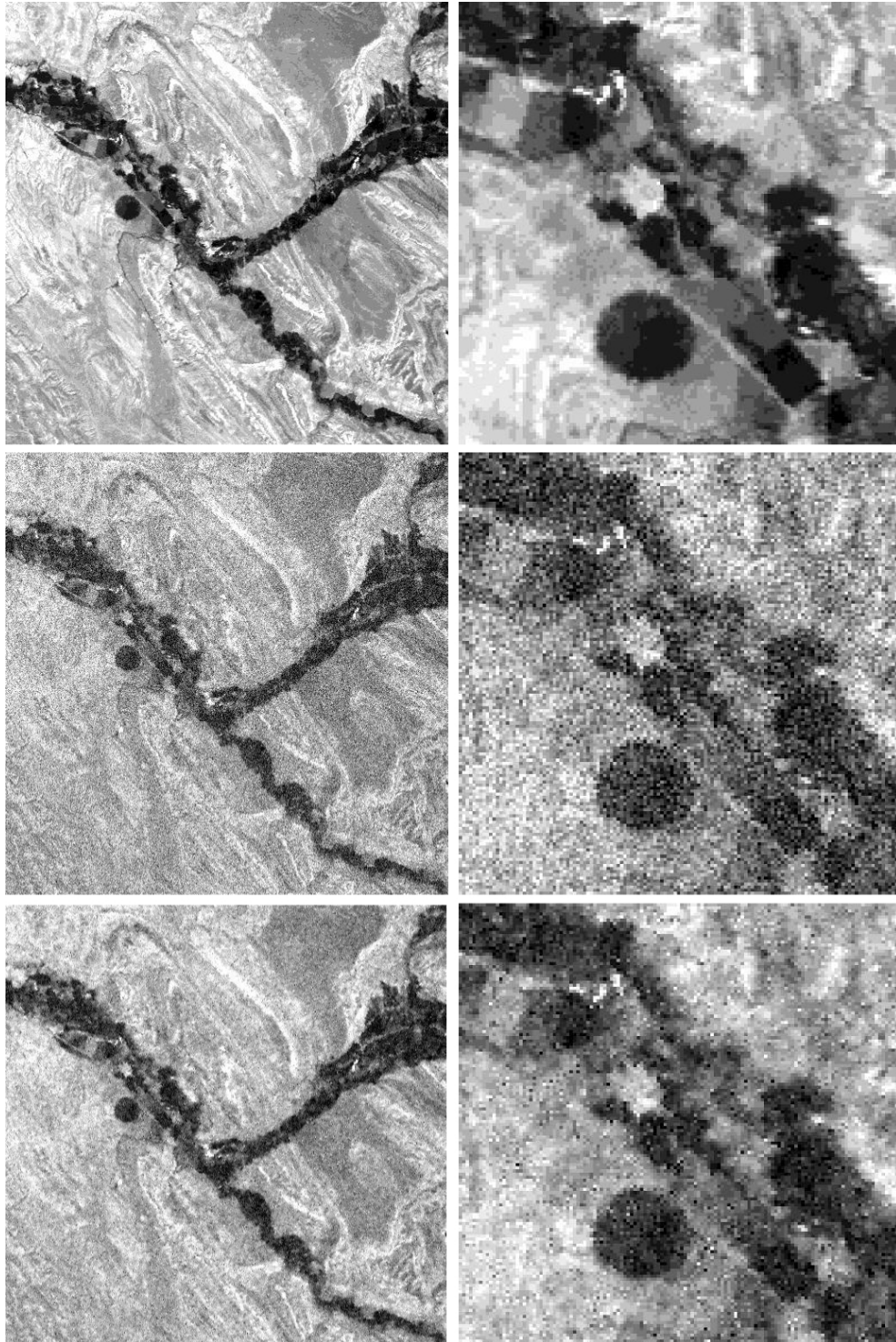


Figura 5.16: Comparación entre imagen original, imagen con ruido gaussiano e imagen con ruido gaussiano con filtro de reducción de ruido local aplicado

El algoritmo más complejo de los desarrollados es sin duda es el filtro de mediana adaptativo. Aunque, por lo visto en la figura 5.17, pareciera justificar su complejidad, ya que el ruido impulsivo fue eliminado casi por completo afectando muy poco la imagen. El filtro utilizado tiene un tamaño máximo de ventana igual a 7.

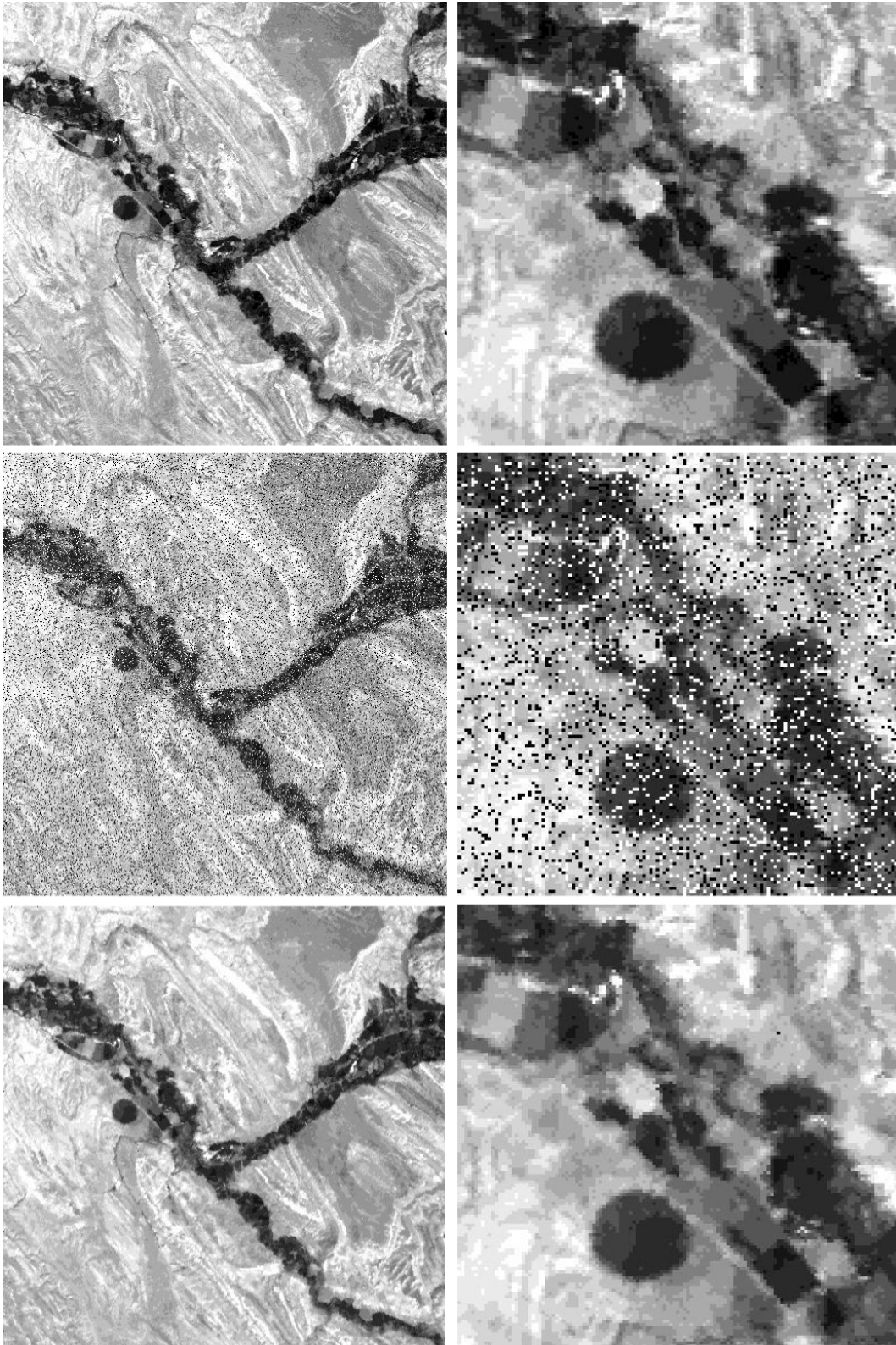


Figura 5.17: Comparación entre imagen original, imagen con ruido impulsivo e imagen con ruido impulsivo con filtro de mediana adaptativo aplicado

5.5. Posclasificación

En la figura 5.18 se puede ver el resultado del cálculo de la matriz de confusión y el coeficiente kappa entre dos clasificaciones realizadas con el método ISODATA, utilizando los mismos parámetros para generarlas. La primera clasificación se realizó sobre la imagen original y la segunda sobre la imagen con ruido uniforme con un filtro de media aritmética aplicado.

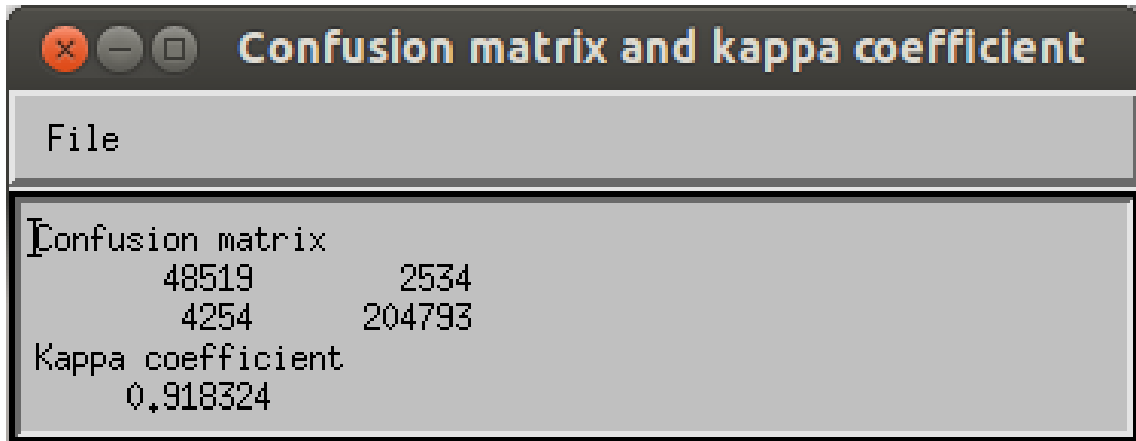


Figura 5.18: Matriz de confusión y coeficiente de Kappa entre dos clasificaciones

Podemos ver que a pesar de que el efecto del filtrado no fue el ideal como vimos en la sección 5.2 el resultado obtenido por ambas clasificaciones no fue tan distinto. Esta es la razón por la cual el estudio, al que se pretende asistir con las herramientas desarrolladas, es tan importante.

La teledetección espacial es de gran utilidad para realizar estudios de distintos eventos que ocurren en nuestro planeta. En este sentido, las herramientas que se utilizan para analizar los datos que se obtienen de esta forma son cada vez más importantes. Por lo que, estas herramientas no solo deben brindar un alto grado de confiabilidad a los usuarios, sino que también debe ser claro cuál es su función, es decir, para qué sirven y para qué no.

Por esta razón se decidió seguir las normas de ingeniería de software de la ESA, particularmente las que son aplicadas a pequeños proyectos, ya que por las características del trabajo propuesto entra en la categoría de pequeño. Esto, sumado a algunas prácticas que son utilizadas y ampliamente aceptadas en los procesos de desarrollo de software, permitió que se pudiera generar un producto con la calidad necesaria y suficiente para la realización del estudio para lo que fue propuesto.

Los tiempos utilizados para el estudio y la documentación de requerimientos, diseño y planes de prueba, fue de alrededor del 80 % del tiempo total. La etapa de prueba del software implementado se llevó a cabo satisfactoriamente, probando que se cubre cada uno de los requerimientos establecidos.

La decisión de lograr que el resultado de este proyecto forme parte de otra aplicación se basó en la posibilidad de potenciarla y simplificarla. En definitiva, con las funcionalidades agregadas, los requerimientos para realizar el estudio propuesto no solo fueron alcanzadas sino que fueron superadas, ya que al formar parte de ENVI la posibilidad de acción es mayor.

Es importante remarcar que con este trabajo el usuario no solo podrá utilizar las operaciones que necesite para realizar el estudio planteado como objetivo, o cualquier otro estudio que involucre estas operaciones, sino que también tiene información detallada de cómo está diseñada, desarrollada y probada cada funcionalidad. Por otro lado, la modularización que tiene este proyecto permite a cualquier desarrollador de software agregar otras funcionalidades con facilidad.

6.1. Trabajo a futuro

Es interesante la posibilidad que genera este proyecto en materia de la adición de nuevas funcionalidades a ENVI para brindar más y mejores herramientas a los usuarios, ya que lo que comenzó como una idea de un software para realizar un estudio particular terminó potenciando una aplicación de gran uso en teledetección espacial.

Como se comentó en el primer capítulo este trabajo fue diseñado para ser el comienzo de un am-

plio estudio. Este estudio implica la comparación de clasificaciones de imágenes previamente filtradas con la intención de reducir los errores. El objetivo es determinar cuál es el momento indicado para intentar eliminar ruido de imágenes e inclusive generar nuevos algoritmos de filtrado.

APÉNDICE A _____

_____ DOCUMENTO DE REQUERIMIENTOS DE USUARIO

Documento de Requerimientos de Usuario (URD)

Software para filtrado y clasificación de imágenes de teledetección

Autor: Mariano E. Della Vedova

Fecha: 10 de Agosto de 2013

Versión: 1.1

HISTORIAL DE CAMBIOS

Versión del Documento	Fecha	Sección	Razón del cambio o revisión
1.0	20/06/2013	Todo	Primera revisión
1.1	10/08/2013	3.1	Cambio en los requerimiento UR1 y UR6. Cambio en la redacción de los requerimientos UR3, UR4 y UR5.

TABLA DE CONTENIDOS

1	Introducción	4
1.1	Propósito del documento	4
1.2	Definiciones, acrónimos y abreviaturas	4
1.3	Referencias	4
1.4	Resumen del documento	4
2	Descripción general	4
2.1	Capacidades generales	4
2.2	Restricciones generales	4
2.3	Características de los usuarios	5
2.4	Entorno operativo	5
3	Requerimientos específicos	5
3.1	Requerimientos de capacidades	5
3.2	Requerimientos de restricciones	6

1. Introduccción

1.1. Propósito del documento

Este documento describe formalmente los requerimientos de usuario del Software para filtrado y clasificación de imágenes de teledetección, el cual será utilizado para realizar diferentes estudios sobre la utilización de filtros y métodos de clasificación sobre imágenes satelitales.

1.2. Definiciones, acrónimos y abreviaturas

Acrónimo	Descripción
ID	Identificación
UR	Requerimiento de usuario

Tabla 1.2.0.-a: Lista de definiciones, acrónimos y abreviaturas

1.3. Referencias

ID	Título	Versión	Fecha
RF.1	Guide to applying the ESA software engineering standards to small software projects	1.0	08/05/1996

Tabla 1.3.0.-a: Documentos de Referencia

1.4. Resumen del documento

Este documento está estructurado de la siguiente forma:

- El *capítulo 2* contiene una descripción general del sistema.
- El *capítulo 3* detalla los requerimientos específicos del sistema.

2. Descripción general

2.1. Capacidades generales

El usuario requiere un sistema por el cual pueda realizar estudios para determinar si la clasificación de imágenes de teledetección tiene un mejor resultado aplicando filtros anteriormente o no.

El sistema deberá ser capaz de agregar ruido, filtrar y clasificar imágenes de teledetección. También deberá comparar una clasificación obtenida con una clasificación modelo o "ideal" definida por el usuario.

2.2. Restricciones generales

El sistema debe ser implementado para poder usarse en el sistema operativo Linux.

2.3. Características de los usuarios

El sistema tendrá una fuerte interacción con el usuario, ya que este deberá indicar qué tipo de error se le quiere aplicar a la imagen, que tipo de filtro se quiere utilizar (si es que esto es requerido) y que tipo de clasificador se quiere utilizar (y sus respectivos parámetros de entrada de ser necesario). Por lo cual, el usuario debe tener conocimientos básicos de tratamiento de imágenes de teledetección. Por otro lado, el sistema estará íntegramente desarrollado en el sistema operativo Linux, por lo que el usuario deberá tener conocimientos mínimos de este sistema.

2.4. Entorno operativo

Este sistema debe ser independiente, en el sentido en que las entradas del sistema no dependen de un sistema externo en particular, por lo que no provee interfaces con otros sistemas externos. Esto quiere decir que la única interacción existente se da entre el sistema y el usuario, y se produce mediante entradas cuyo origen no altera el comportamiento del mismo.

3. Requerimientos específicos

Esta sección describe todos los requerimientos de usuario del Software para filtrado y clasificación de imágenes de teledetección. Es importante explicar la forma en que se priorizan los requerimientos. Cada requerimiento es priorizado de la siguiente manera en orden de importancia:

M - MUST Describe un requerimiento que debe satisfacerse en la solución final para que el sistema se considere exitoso.

S - SHOULD Representa un ítem de prioridad alta que debe ser incluido en la solución si es posible. Este es frecuentemente un requerimiento crítico, pero a su vez puede ser satisfecho de otra forma si es estrictamente necesario.

C - COULD Describe un requerimiento que es considerado deseable pero no necesario. Esto será incluido si el tiempo y los recursos lo permiten.

W - WOULD Representa un requerimiento que los interesados en el sistema acordaron que no se implementará para el producto final, pero puede ser considerado para el futuro.

3.1. Requerimientos de capacidades

UR ID	Descripción	Prioridad
UR1	El sistema deberá ser una interfaz gráfica con un menú interactivo. La aplicación deberá obtener del usuario los archivos y parámetros necesarios para realizar las diferentes acciones requeridas por el sistema.	M
UR2	El sistema deberá procesar imágenes que son productos finales de adquisiciones de distintos satélites. Las imágenes que deberá procesar provienen de los satélites: AQUA, ERS-2, LANDSAT-5, LANDSAT-7, SAC-C, SEAWIFS y TERRA.	S
UR3	El sistema deberá poder generar errores sobre las imágenes de teledetección originales. Los tipos de errores que se podrán generar son ruido Gaussiano, uniforme e impulsivo. El sistema deberá poder guardar en un archivo diferente al original la imagen con ruido agregado.	M

Continúa en la página siguiente ...

Tabla 3.1.0.-a – Continuación

ID	Descripción	Prioridad
UR4	El sistema deberá poder filtrar imágenes de teledetección. Los filtros que podrán ser aplicados son filtro de media aritmética, de media geométrica, de media armónica, de media contrarmónica, Gaussiano, de mediana, de mínima, de máxima, olímpico, de media alfa recortada, de media de los k valores más cercanos, de mediana de los k valores más cercanos, de reducción de ruido local y de mediana adaptativo. El sistema deberá poder guardar en un archivo diferente al original la imagen filtrada.	M
UR5	El sistema deberá poder clasificar imágenes de teledetección. Se deberá poder clasificar con las técnicas de clasificación de máxima probabilidad, de mínima distancia, de paralelepípedos e ISODATA. El sistema deberá poder guardar en un archivo la clasificación realizada.	M
UR6	El sistema deberá poder comparar dos clasificaciones mediante una matriz de confusión y el coeficiente kappa.	M

Tabla 3.1.0.-a: Requerimientos de usuario de capacidades

3.2. Requerimientos de restricciones

UR ID	Descripción	Prioridad
UR7	El sistema deberá correr bajo el sistema operativo Linux.	M
UR8	El desarrollo completo del sistema deberá seguir los estándares ESA para pequeños proyectos.	S

Tabla 3.2.0.-a: Requerimientos de usuario de restricciones

APÉNDICE B _____

_____ DOCUMENTO DE ESPECIFICACIÓN DE SOFTWARE

Documento de Especificación de Software (SSD)

Software para filtrado y clasificación de imágenes de teledetección

Autor: Mariano E. Della Vedova

Fecha: 10 de Agosto de 2013

Versión: 1.1

HISTORIAL DE CAMBIOS

Versión del Documento	Fecha	Sección	Razón del cambio o revisión
1.0	20/06/2013	Todo	Primera revisión
1.1	10/08/2013	2	Cambio en el diagrama de actividades. Aclaración en el primer párrafo. Método de comparación de clasificaciones en el quinto párrafo.
1.1	10/08/2013	3.1	Mejora en la redacción de los requerimientos SR5, SR6, SR7, SR8, SR9, SR10, SR11, SR12, SR13, SR14, SR15, SR16, SR17 y SR18. Cambio de requerimiento SR23. Se agrega el requerimiento SR51.
1.1	10/08/2013	3.2	Corrección de los nombres de ID. Cambio en los requerimientos SR24, SR25, SR27, SR29. Eliminación de los requerimientos SR26, SR28. Se agregan los requerimientos SR52 y SR53.
1.1	10/08/2013	3.3	Cambio en el SR30. Se agregan los requerimientos SR54, SR55, SR56, SR57 y SR58.
1.1	10/08/2013	4	Cambio en la descripción del subsistema de control y el subsistema de procesamiento de imágenes.
1.1	10/08/2013	5	Cambio en la descripción de los componentes unidad de control, gestor de filtros, gestor de clasificaciones y unidad de comparación de clasificaciones.
1.1	10/08/2013	6	Corrección de la matriz de trazabilidad.
1.1	10/08/2013	7	Corrección de la matriz de trazabilidad.

TABLA DE CONTENIDOS

1	Introducción	5
1.1	Propósito del documento	5
1.2	Definiciones, acrónimos y abreviaturas	5
1.3	Referencias	5
1.4	Resumen del documento	5
2	Descripción del modelo	6
3	Requerimientos específicos	7
3.1	Requerimientos funcionales	7
3.2	Requerimientos de interfaces	9
3.3	Requerimientos operacionales	9
3.4	Requerimientos de documentación	10
3.5	Requerimientos de portabilidad	11
4	Diseño de sistema	11
5	Descripción de componentes	12
5.1	C01 - Unidad de control	12
5.1.1	Tipo	12
5.1.2	Función	12
5.1.3	Interfaces	12
5.1.4	Dependencias	12
5.1.5	Procesamiento	13
5.1.6	Datos	13
5.1.7	Recursos	13
5.2	C02 - Gestor de ruido	13
5.2.1	Tipo	13
5.2.2	Función	13
5.2.3	Interfaces	13
5.2.4	Dependencias	13
5.2.5	Procesamiento	13
5.2.6	Datos	13
5.3	C03 - Gestor de filtros	14
5.3.1	Tipo	14
5.3.2	Función	14
5.3.3	Interfaces	14
5.3.4	Dependencias	14
5.3.5	Procesamiento	14
5.3.6	Datos	14
5.4	C04 - Gestor de clasificación	14
5.4.1	Tipo	14
5.4.2	Función	14
5.4.3	Interfaces	14
5.4.4	Dependencias	15
5.4.5	Procesamiento	15
5.4.6	Datos	15
5.5	C05 - Unidad de comparación de clasificaciones	15
5.5.1	Tipo	15
5.5.2	Función	15
5.5.3	Interfaces	15

5.5.4	Dependencias	15
5.5.5	Procesamiento	15
5.5.6	Datos	15
6	Matriz de trazabilidad entre requerimientos de usuario y requerimientos de Software	15
7	Matriz de trazabilidad entre requerimientos de Software y componentes	16

1. Introducción

1.1. Propósito del documento

Este documento describe los requerimientos de Software y el diseño del Software para filtrado y clasificación de imágenes de teledetección.

Este documento pretende proveer información sobre el diseño general para la siguiente fase de codificación y desarrollo, i.e. diseño detallado e implementación.

1.2. Definiciones, acrónimos y abreviaturas

Acrónimo	Descripción
ID	Identificación
SR	Requerimiento de especificación
UR	Requerimiento de usuario
URD	Documento de requerimientos de usuario

Tabla 1.2.0.-a: Lista de definiciones, acrónimos y abreviaturas

1.3. Referencias

ID	Título	Versión	Fecha
RF.1	Documento de Requerimientos de Usuario (URD) - Software para filtrado y clasificación de imágenes de teledetección	1.1	10/08/2013
RF.2	Guide to applying the ESA software engineering standards to small software projects	1.0	08/05/1996

Tabla 1.3.0.-a: Documentos de Referencia

1.4. Resumen del documento

Este documento está estructurado de la siguiente forma:

- El *capítulo 2* describe el modelo lógico del sistema. Este modelo es creado a partir de la interpretación de los requerimientos de usuario establecidos por el URD (RF.1).
- El *capítulo 3* detalla los requerimientos específicos de software para el sistema, se incluyen los requerimientos funcionales, operacionales, de interfaces y de portabilidad.
- El *capítulo 4* describe la arquitectura diseñada para el sistema completo. En él se muestra una representación gráfica que representa al sistema como un conjunto de componentes.
- El *capítulo 5* brinda una descripción detallada de cada uno de los componentes del sistema, detallando cuál es la función de cada uno de ellos y cuáles son sus interfaces con el resto de los componentes.
- El *capítulo 6* contiene la matriz de trazabilidad entre requerimientos de usuario y requerimientos de software.
- El *capítulo 7* contiene la matriz de trazabilidad entre requerimientos de software y componentes de software.

2. Descripción del modelo

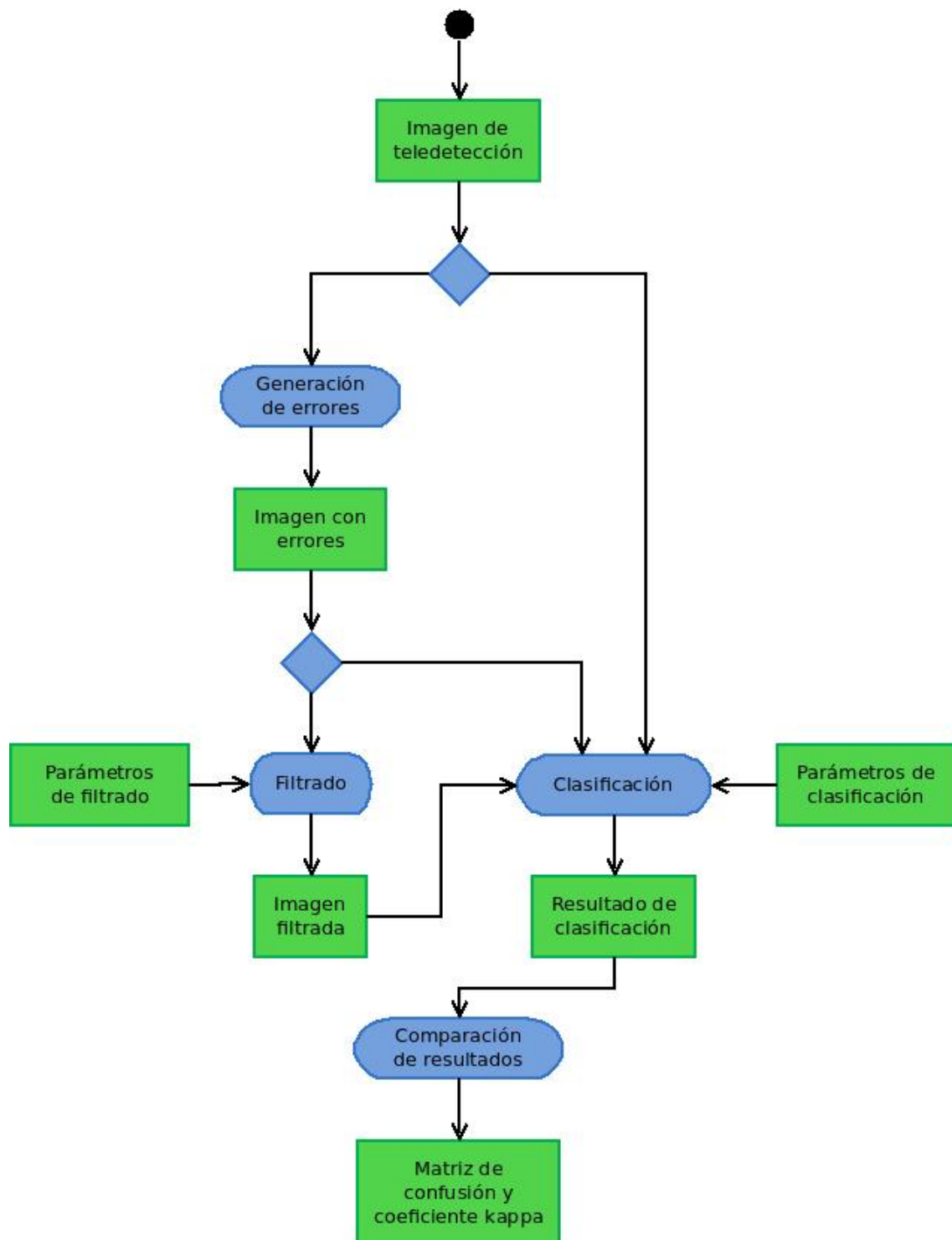


Figura 2.0.0.-a: Diagrama de actividades representando el modelo lógico del sistema

En la figura 2.0.0.-a observamos el modelo lógico del Software para filtrado y clasificación de imágenes de teledetección representado en un diagrama de actividades. El modelo indica que se utilizará una imagen de teledetección como entrada principal. Una copia de la imagen original será utilizada para realizar las diferentes acciones que ofrece el sistema. Es importante aclarar que si bien el sistema debería poder realizar en orden la secuencia de actividades descrita en el diagrama de

actividades, esta secuencia no se realizará necesariamente en el orden establecido. Por lo que, el sistema no realizará un control sobre la secuencia de actividades.

Con la imagen original (sin errores) será posible realizar dos acciones: generar errores sobre la imagen o clasificarla. Si se decide aplicarle errores a las imagen, se deberá indicar que tipo de error se quiere aplicar.

La imagen (con errores) resultante podrá ser filtrada o clasificada. Si lo que se desea es filtrar la imagen, también deberá elegirse qué tipo de filtro se quiere aplicar y los parámetros necesarios para ejecutar el mismo.

Para la etapa de clasificación se puede utilizar una imagen sin errores, una imagen con errores filtrada o una imagen con errores sin filtrar. En esta etapa se debe decidir qué tipo de clasificación se quiere hacer y los parámetros necesarios para ejecutar la misma.

Finalmente, el sistema permite comparar dos clasificaciones tomando una como la "ideal". Esta comparación se realizará mediante una matriz de confusión y el coeficiente de kappa.

Es importante indicar que en ninguna de las etapas se verificará que la entrada es la esperada, así, por ejemplo, en la etapa de generación de errores, no se verificará que la imagen no los tenga. Es tarea del usuario proveer la entrada correcta en cada una de las etapas.

Por último, al finalizar cada una de las etapas, se generará un archivo con el resultado del procesamiento, para que pueda retomarse cualquier etapa del sistema en cualquier momento.

3. Requerimientos específicos

Esta sección describe todos los requerimientos de usuario del Software para filtrado y clasificación de imágenes de teledetección. Es importante explicar la forma en que se priorizan los requerimientos. Cada requerimiento es priorizado de la siguiente manera en orden de importancia:

M - MUST Describe un requerimiento que debe satisfacerse en la solución final para que el sistema se considere exitoso.

S - SHOULD Representa un item de prioridad alta que debe ser incluido en la solución si es posible. Este es frecuentemente un requerimiento crítico, pero a su vez puede ser satisfecho de otra forma si es estrictamente necesario.

C - COULD Describe un requerimiento que es considerado deseable pero no necesario. Esto será incluido si el tiempo y los recursos lo permiten.

W - WOULD Representa un requerimiento que los interesados en el sistema acordaron que no se implementará para el producto final, pero puede ser considerado para el futuro.

3.1. Requerimientos funcionales

SR ID	UR ID	Descripción	Prioridad
SR1	UR2	El sistema deberá reconocer imágenes provenientes de los satélites: AQUA, ERS-2, LANDSAT-5, LANDSAT-7, SAC-C, SEAWIFS y TERRA.	S
SR2	UR3	El sistema deberá poder agregar ruido Gaussiano sobre cualquier tipo de imagen que éste pueda reconocer.	M
SR3	UR3	El sistema deberá poder agregar ruido uniforme sobre cualquier tipo de imagen que éste pueda reconocer.	M

Continúa en la página siguiente ...

Tabla 3.1.0.-a – Continuación

SR ID	UR ID	Descripción	Prioridad
SR4	UR3	El sistema deberá poder agregar ruido impulsivo sobre cualquier tipo de imagen que éste pueda reconocer.	M
SR5	UR4	El sistema deberá poder filtrar una imagen con un filtro de media aritmética.	M
SR6	UR4	El sistema deberá poder filtrar una imagen con un filtro de media geométrica.	M
SR7	UR4	El sistema deberá poder filtrar una imagen con un filtro de media armónica.	M
SR8	UR4	El sistema deberá poder filtrar una imagen con un filtro de media contrarmónica.	M
SR9	UR4	El sistema deberá poder filtrar una imagen con un filtro Gaussiano.	M
SR10	UR4	El sistema deberá poder filtrar una imagen con un filtro de mediana.	M
SR11	UR4	El sistema deberá poder filtrar una imagen con un filtro de mínima.	M
SR12	UR4	El sistema deberá poder filtrar una imagen con un filtro de máxima.	M
SR13	UR4	El sistema deberá poder filtrar una imagen con un filtro olímpico.	M
SR14	UR4	El sistema deberá poder filtrar una imagen con un filtro de media alfa recortada.	M
SR15	UR4	El sistema deberá poder filtrar una imagen con un filtro de media de los k valores más cercanos.	M
SR16	UR4	El sistema deberá poder filtrar una imagen con un filtro de mediana de los k valores más cercanos.	M
SR17	UR4	El sistema deberá poder filtrar una imagen con un filtro de reducción de ruido local.	M
SR18	UR4	El sistema deberá poder filtrar una imagen con un filtro de mediana adaptativo.	M
SR19	UR5	El sistema deberá poder clasificar una imagen mediante la clasificación de máxima probabilidad.	M
SR20	UR5	El sistema deberá poder clasificar una imagen mediante la clasificación de mínima distancia.	M
SR21	UR5	El sistema deberá poder clasificar una imagen mediante la clasificación de paralelepípedos.	M
SR22	UR5	El sistema deberá poder clasificar una imagen mediante la clasificación ISODATA.	M
SR23	UR6	El sistema deberá poder comparar dos clasificaciones mediante la matriz de confusión.	M
SR51	UR6	El sistema deberá poder calcular el coeficiente Kappa entre dos clasificaciones.	M

Tabla 3.1.0.-a: Requerimientos funcionales.

3.2. Requerimientos de interfaces

SR ID	UR ID	Descripción	Prioridad
SR24	UR1	El sistema deberá poder mostrar una imagen guardada en un directorio indicado por el usuario.	M
SR25	UR3	El sistema deberá poder guardar una imagen con ruido adicionado en un archivo cuyo nombre y directorio donde se guardará será indicado por el usuario.	M
SR27	UR4	El sistema deberá poder guardar una imagen con ruido filtrada en un archivo cuyo nombre y directorio donde se guardará será indicado por el usuario.	M
SR29	UR5	El sistema deberá poder guardar una clasificación de una imagen en un archivo cuyo nombre y directorio donde se guardará será indicado por el usuario.	M
SR52	UR6	El sistema deberá poder obtener los datos de una clasificación desde un archivo indicado por el usuario.	M
SR53	UR6	El sistema deberá poder guardar la matriz de confusión y el coeficiente kappa en un archivo cuyo nombre y directorio donde se guardará será indicado por el usuario.	M

Tabla 3.2.0.-a: Requerimientos de interfaces.

3.3. Requerimientos operacionales

SR ID	UR ID	Descripción	Prioridad
SR30	UR1	El sistema deberá correr como una aplicación con interfaz gráfica.	S
SR31	UR1	El sistema deberá ofrecer un menú con las diferentes operaciones disponibles en el sistema.	S
SR54	UR1	El sistema deberá requerir la media y desviación estándar para la generación de ruido Gaussiano sobre una imagen.	M
SR55	UR1	El sistema deberá requerir el rango en el cuál se generarán el ruido que e sumará a una imagen.	M
SR32	UR1	El sistema deberá requerir las probabilidades de cada uno de los tipos de ruidos impulsivos (sal y pimienta) al generar este tipo de ruido sobre una imagen.	M
SR33	UR1	El sistema deberá requerir el tamaño de la ventana para el filtro de media aritmética.	M
SR34	UR1	El sistema deberá requerir el tamaño de la ventana para el filtro de media geométrica.	M

Continúa en la página siguiente ...

Tabla 3.3.0.-a – Continuación

SR ID	UR ID	Descripción	Prioridad
SR35	UR1	El sistema deberá requerir el tamaño de la ventana para el filtro de media armónica.	M
SR36	UR1	El sistema deberá requerir el tamaño de la ventana y el orden del filtro de media contrarmónica.	M
SR37	UR1	El sistema deberá requerir el tamaño de la ventana y la desviación estándar para el filtro Gaussiano.	M
SR38	UR1	El sistema deberá requerir el tamaño de la ventana para el filtro de mediana.	M
SR39	UR1	El sistema deberá requerir el tamaño de la ventana para el filtro de mínima.	M
SR40	UR1	El sistema deberá requerir el tamaño de la ventana para el filtro de máxima.	M
SR41	UR1	El sistema deberá requerir el tamaño de la ventana para el filtro olímpico.	M
SR42	UR1	El sistema deberá requerir el tamaño de la ventana y el valor de recorte para el filtro de media alfa recortada.	M
SR43	UR1	El sistema deberá requerir el tamaño de la ventana y de k para el filtro de media de los k valores más cercanos.	M
SR44	UR1	El sistema deberá requerir el tamaño de la ventana y de k para el filtro de mediana de los k valores más cercanos.	M
SR56	UR1	El sistema deberá requerir el tamaño de la ventana y la varianza global del ruido para el filtro de reducción de ruido local.	M
SR45	UR1	El sistema deberá requerir el tamaño máximo de la ventana permitido para el filtro de mediana adaptativo.	M
SR46	UR1	El sistema deberá requerir los datos de entrenamiento para la clasificación de máxima probabilidad.	M
SR47	UR1	El sistema deberá requerir los datos de entrenamiento para la clasificación de mínima distancia.	M
SR48	UR1	El sistema deberá requerir los datos de entrenamiento para la clasificación de paralelepípedos.	M
SR57	UR1	El sistema deberá requerir dos clasificaciones para generar la matriz de confusión.	M
SR58	UR1	El sistema deberá requerir una matriz de confusión para generar el coeficiente Kappa.	M

Tabla 3.3.0.-a: Requerimientos operacionales.

3.4. Requerimientos de documentación

SR ID	UR ID	Descripción	Prioridad
SR49	UR8	El desarrollo completo del sistema deberá seguir los estándares ESA para pequeños proyectos.	S

Tabla 3.4.0.-a: Requerimientos de documentación.

3.5. Requerimientos de portabilidad

SR ID	UR ID	Descripción	Prioridad
SR50	UR7	El sistema deberá correr bajo el sistema operativo Linux.	M

Tabla 3.5.0.-a: Requerimientos de portabilidad.

4. Diseño de sistema

La figura 4.0.0.-a muestra la división en componentes diseñada para este sistema. Se pueden distinguir tres subsistemas con funciones bien definidas:

- El **Subsistema de Control** es el subsistema principal del sistema. Contiene la *unidad de Control*, la cual es la encargada de determinar la secuencia de ejecución de los distintos componentes de los demás subsistemas. También está encargada de cargar una imagen antes de ser modificada por alguno de los otros módulos.
- El **Subsistema de Procesamiento de Imágenes** contiene todos los gestores de tratamiento de imágenes. Cada uno de estos gestores están presentados en módulos. El módulo de *gestión de ruidos* dispone de todos los algoritmos de generación de ruido en imágenes. El módulo de *gestión de filtros* tiene como objetivo manejar los filtros disponibles. Por último, el módulo de *gestión de clasificación* se ocupa de manejar los distintos métodos de clasificación de imagen.
- El **Subsistema de Verificación de Resultados** es el encargado de la generación de conclusiones basadas en la comparación de resultados dependientes de la secuencia de acciones realizada para obtener las diferentes clasificaciones. Para esto, se utilizará la *unidad de comparación de clasificaciones*.

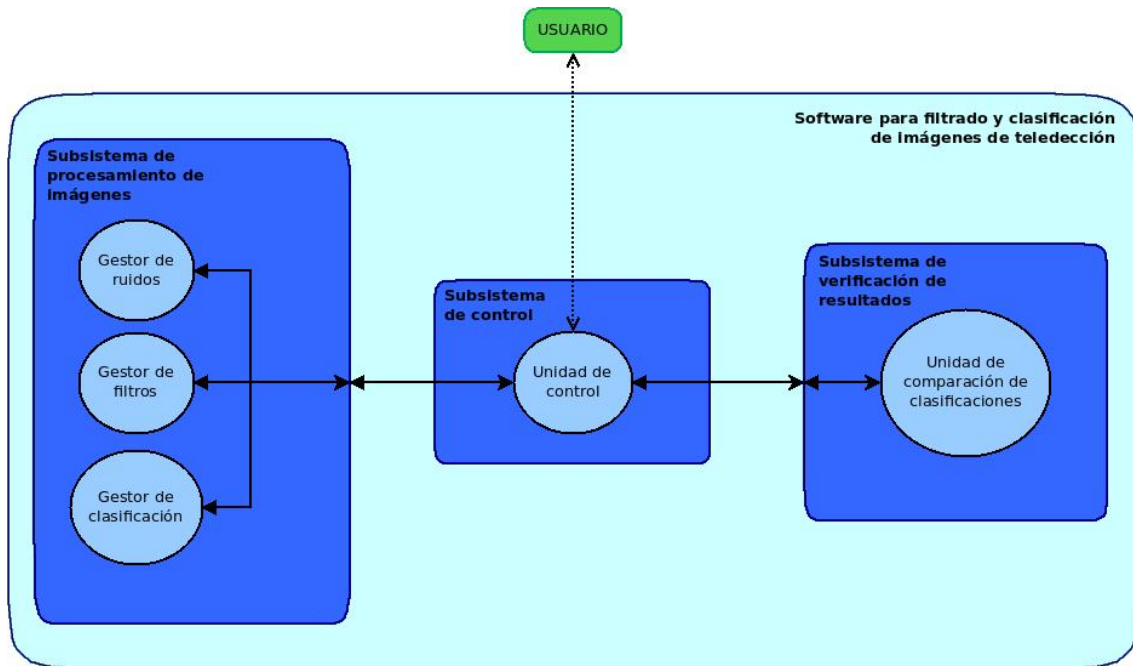


Figura 4.0.0.-a: Diagrama de componentes del Software para filtrado y clasificación de imágenes de teledetección

5. Descripción de componentes

A continuación se describe cada uno de los componentes descritos en el diseño de sistema presentado en la sección 4.

5.1. C01 - Unidad de control

5.1.1. Tipo

Es un módulo encargado del control del sistema. Provee la interacción entre el sistema y el usuario.

5.1.2. Función

La función de este módulo es la de proveer la interfaz con el usuario y mediante ésta obtener las imágenes que quiere modificarse. También es la encargada de decidir la secuencia con la cual se ejecutará el resto de las unidades basada en la decisión del usuario.

5.1.3. Interfaces

La interacción con el usuario será mediante una interfaz gráfica. Está pedirá la información necesaria para ejecutar el resto de los módulos. Esta información comprenderá, por ejemplo, qué secuencia de acciones ejecutar, imágenes a utilizar, etc.

Las interfaces con otros módulos será mediante llamadas a funciones según el lenguaje de programación que se utilizará.

5.1.4. Dependencias

Este módulo no depende de ningún otro.

5.1.5. Procesamiento

Este módulo ofrece al usuario los diferentes tratamientos que se le puede aplicar a una imagen con el Software para filtrado y clasificación de imágenes de teledetección, así como también la posibilidad de comparar clasificaciones. Requiere cierta información necesaria para ejecutar los diferentes procesos solicitados.

En base a las determinaciones del usuario decide con qué módulos interactuar brindándoles la información necesaria para que estos puedan realizar las acciones requeridas.

5.1.6. Datos

Los datos requeridos por este módulo dependerá de los procesos que el usuario quiera ejecutar.

5.1.7. Recursos

Este módulo solo requiere una pantalla, un teclado y un mouse, ya que será el medio por el cual se comunicará con el usuario

5.2. C02 - Gestor de ruido

5.2.1. Tipo

Es un módulo encargado de gestionar todos los algoritmos de generación de ruido sobre imágenes.

5.2.2. Función

La función principal de este módulo es gestionar los diferentes algoritmos de generación de ruido: Gaussiano, uniforme e impulsivo.

5.2.3. Interfaces

Este módulo brindará una interfaz (un método) a la unidad de control que permitirá indicar qué algoritmo de generación de ruido se quiere ejecutar y los parámetros que éste necesita.

5.2.4. Dependencias

Este módulo, en el sistema, depende de la Unidad de Control, ya que ésta será la encargada de realizar las llamadas correspondientes. Aunque la idea de desarrollarlo como un módulo separado radica en la posibilidad de utilizarlo en otros sistemas si esto es requerido, por lo que este módulo en realidad es independiente.

5.2.5. Procesamiento

Este módulo recibe la indicación de qué algoritmo de generación de errores se quiere ejecutar y aplica el error indicado sobre la imagen tomada como entrada.

5.2.6. Datos

Los datos requeridos por este módulo dependerán del error que se quiera aplicar. La entrada que todos los algoritmos requieren es una imagen a la cual se le aplicará el error.

5.3. C03 - Gestor de filtros

5.3.1. Tipo

Es un módulo encargado de gestionar todos los algoritmos de filtrado de imágenes.

5.3.2. Función

La función principal de este módulo es gestionar los diferentes algoritmos de filtrado: filtro de media aritmética, filtro de media geométrica, filtro de media armónica, filtro de media contrarmónica, filtro Gaussiano, filtro de mediana, filtro de mínima, filtro de máxima, filtro olímpico, filtro de media alfa recortada, filtro de media de los k valores más cercanos, filtro de mediana de los k valores más cercanos, filtro de mediana adaptativo y filtro de reducción de ruido local.

5.3.3. Interfaces

Este módulo brindará una interfaz (un método) a la unidad de control que permitirá indicar que algoritmo de filtrado se quiere ejecutar y los parámetros que este necesita.

5.3.4. Dependencias

Este módulo, en el sistema, depende de la Unidad de Control, ya que ésta será la encargada de realizar las llamadas correspondientes. Aunque la idea de desarrollarlo como un módulo separado radica en la posibilidad de utilizarlo en otros sistemas si esto es requerido, por lo que este módulo en realidad es independiente.

5.3.5. Procesamiento

Este módulo recibe la indicación de qué algoritmo de filtrado se quiere ejecutar y aplica el filtro indicado sobre la imagen tomada como entrada.

5.3.6. Datos

Los datos requeridos por este módulo dependerán del filtro que se quiera aplicar. La entrada que todos los algoritmos requieren es una imagen a la cual se le aplicará el filtro.

5.4. C04 - Gestor de clasificación

5.4.1. Tipo

Es un módulo encargado de gestionar todos los algoritmos de clasificación de imágenes.

5.4.2. Función

La función principal de este módulo es gestionar los diferentes algoritmos de clasificación de imagen: clasificación de máxima probabilidad, clasificación de mínima distancia, clasificación de paralelepípedos y clasificación ISODATA.

5.4.3. Interfaces

Este módulo brindará una interfaz (un método) a la unidad de control que permitirá indicar que algoritmo de clasificación se quiere ejecutar y los parámetros que éste necesita.

5.4.4. Dependencias

Este módulo, en el sistema, depende de la Unidad de Control, ya que ésta será la encargada de realizar las llamadas correspondientes.

5.4.5. Procesamiento

Este módulo recibe la indicación de qué algoritmo de clasificación se quiere ejecutar y aplica la clasificación sobre la imagen tomada como entrada.

5.4.6. Datos

Los datos requeridos por este módulo dependerán del tipo de clasificación que se quiera aplicar. La entrada que todos los algoritmos requieren es una imagen a la cual se le aplicará la clasificación.

5.5. C05 - Unidad de comparación de clasificaciones

5.5.1. Tipo

Es un módulo encargado de comparar dos clasificaciones.

5.5.2. Función

La función principal de este módulo es comparar dos clasificaciones, generando una matriz de confusión y calculando el coeficiente kappa.

5.5.3. Interfaces

Este módulo brindará una interfaz (un método) a la unidad de control que permitirá comparar dos clasificaciones entre si.

5.5.4. Dependencias

Este módulo, en el sistema, depende de la Unidad de Control, ya que ésta será la encargada de realizar las llamadas correspondientes. Aunque la idea de desarrollarlo como un módulo separado radica en la posibilidad de utilizarlo en otros sistemas si esto es requerido, por lo que este módulo en realidad es independiente.

5.5.5. Procesamiento

Este módulo compara dos clasificaciones tomando una como la real. De esta forma genera una matriz de confusión y luego calcula el coeficiente kappa.

5.5.6. Datos

Los datos requeridos por este módulo son dos clasificaciones.

6. Matriz de trazabilidad entre requerimientos de usuario y requerimientos de Software

ID Requerimiento de usuario	ID Requerimiento de especificación
UR1	SR24 - SR30 - SR31 SR32 - SR33 - SR34 SR35 - SR36 - SR37 SR38 - SR39 - SR40 SR41 - SR42 - SR43 SR44 - SR45 - SR46 SR47 - SR48 - SR54 SR55 - SR56 - SR57 SR58
UR2	SR1
UR3	SR2 - SR3 - SR4 SR25
UR4	SR5 - SR6 - SR7 SR8 - SR9 - SR10 SR11 - SR12 - SR13 SR14 - SR15 - SR16 SR17 - SR18 - SR27
UR5	SR19 - SR20 - SR21 SR22 - SR29
UR6	SR23 - SR51 - SR52 SR53
UR7	SR50
UR8	SR49

Tabla 6.0.6.-a: Matriz de trazabilidad entre requerimientos de usuario y requerimientos de software

7. Matriz de trazabilidad entre requerimientos de Software y componentes

ID Requerimiento de usuario	ID Componente
SR1 - SR24 - SR25 SR27 - SR29 - SR30 SR31 - SR32 - SR33 SR34 - SR35 - SR36 SR37 - SR38 - SR39 SR40 - SR41 - SR42 SR43 - SR44 - SR45 SR46 - SR47 - SR48 SR50 - SR53 - SR54 SR55 - SR56 - SR57 SR58	C01
SR2 - SR3 - SR4	C02

Continua en la pagina siguiente ...

Tabla 7.0.6.-a – Continuación

ID Requerimiento de usuario	ID Componente
SR5 - SR6 - SR7 SR8 - SR9 - SR10 SR11 - SR12 - SR13 SR14 - SR15 - SR16 SR17 - SR18	C03
SR19 - SR20 - SR21 SR22	C04
SR23 - SR51 - SR52	C05

Tabla 7.0.6.-a: Matriz de trazabilidad entre requerimientos de software y componentes

APÉNDICE C

DOCUMENTO DE DISEÑO DETALLADO

Documento de Diseño Detallado (DDD)

Software para filtrado y clasificación de imágenes de teledetección

Autor: Mariano E. Della Vedova

Fecha: 25 de Agosto de 2013

Versión: 1.0

HISTORIAL DE CAMBIOS

Versión del Documento	Fecha	Sección	Razón del cambio o revisión
1.0	25/08/2013	Todo	Primera revisión

TABLA DE CONTENIDOS

1	Descripción general	4
1.1	Introducción	4
1.1.1	Propósito de este documento	4
1.1.2	Alcance del software	4
1.1.3	Definiciones, acrónimos y abreviaturas	4
1.1.4	Referencias	4
1.1.5	Resumen del documento	5
1.2	Estándares del proyecto, convenciones y procedimientos	5
1.2.1	Estándares de diseño	5
1.2.2	Estándares de documentación	5
1.2.3	Convenciones de nombrado	5
1.2.4	Estándares de programación	5
1.2.5	Herramientas de desarrollo de software	5
2	Especificaciones de diseño de componentes	6
2.1	Diseño	6
2.1.1	Diagrama de casos de uso	6
2.1.2	Diagrama de secuencia	6
2.2	Unidad de control	7
2.2.1	Tipo	7
2.2.2	Propósito	7
2.2.3	Función	7
2.2.4	Subordinados	7
2.2.5	Dependencias	7
2.2.6	Interfaces	7
2.2.7	Procesamiento	7
2.3	Gestor de ruido	7
2.3.1	Tipo	7
2.3.2	Propósito	8
2.3.3	Función	8
2.3.4	Interfaces	8
2.3.5	Procesamiento	8
2.3.6	Datos	8
2.4	Gestor de filtros	8
2.4.1	Tipo	8
2.4.2	Propósito	8
2.4.3	Función	8
2.4.4	Interfaces	8
2.4.5	Procesamiento	9
2.4.6	Datos	9
2.5	Gestor de posclasificación	9
2.5.1	Tipo	9
2.5.2	Propósito	9
2.5.3	Función	9
2.5.4	Interfaces	9
2.5.5	Procesamiento	9
2.5.6	Datos	10

1. Descripción general

1.1. Introducción

1.1.1. Propósito de este documento

Este documento describe el diseño detallado del Software para filtrado y clasificación de imágenes de teledetección, detallando los componentes que lo integran y las interfaces entre ellos. Cabe destacar que este software funcionará con una dependencia directa con una aplicación (ENVI), con la intención de potenciarla, por lo que también se mostrarán las interfaces de este software hacia ENVI.

1.1.2. Alcance del software

El Software para filtrado y clasificación de imágenes de teledetección fue pensado como una solución concreta para un futuro estudio que involucra generar ruido en imágenes de teledetección, aplicarles filtros, clasificarlas (este procedimiento no está incluido en el software ya que forma parte de ENVI) y comparar clasificaciones para determinar que tan buenas han sido.

La idea de potenciar ENVI radica en que este software pueda ser utilizado más allá del estudio mencionado.

1.1.3. Definiciones, acrónimos y abreviaturas

Acrónimo	Descripción
IDL	Interactive Data Language
DDD	Documento de Diseño Detallado
TDD	Test Driven Development
UML	Unified Modeling Language

Tabla 1.1.3.-a: Lista de definiciones, acrónimos y abreviaturas

1.1.4. Referencias

ID	Título	Versión	Fecha
RF.1	Documento de Requerimientos de Usuario (URD) - Software para filtrado y clasificación de imágenes de teledetección	1.1	10/08/2013
RF.2	Documento de Especificación de Software (SSD) - Software para filtrado y clasificación de imágenes de teledetección	1.1	10/08/2013
RF.3	Guide to applying the ESA software engineering standards to small software projects	1.0	08/05/1996
RF.4	Guide to the software detailed design and production phase	1.1	01/03/1995

Tabla 1.1.4.-b: Documentos de Referencia

1.1.5. Resumen del documento

Este documento se divide en dos partes. La primera hace una descripción general del Software para filtrado y clasificación de imágenes de teledetección y de cómo fue desarrollado. Para esto se especifica los estándares usados en la etapa de desarrollado.

La segunda parte detalla el diseño, mostrando un diagrama de secuencia y el diseño de cada módulo.

1.2. Estándares del proyecto, convenciones y procedimientos

1.2.1. Estándares de diseño

Como estándar de diseño se utiliza UML. Para esto se utiliza un diagrama de secuencia y un diagrama de casos de uso.

1.2.2. Estándares de documentación

Para la documentación se siguen los estándares de la Agencia Espacial Europea para el diseño detallado definidos por en el documento "Guide to the software detailed design and production phase" (RF.4), en combinación con la guía para la aplicación a pequeños proyectos "Guide to applying the ESA software engineering standards to small software projects" (RF.3).

1.2.3. Convenciones de nombrado

Para el desarrollo del Software para filtrado y clasificación de imágenes de teledetección se tomó como convención utilizar, para las funciones, nombres que sean acciones. Cuando un nombre de función tiene más de una palabra, se separan con un guión bajo (_). Tanto en la definición de la función como en el uso de la misma, se utiliza todo el nombre en mayúsculas.

Las funciones que responden a acciones del menú, se le agrega al final "MENU.EVENT".

Los nombres de los archivos que contienen funciones llevan el mismo nombre que la función pero en minúsculas (por exigencia del lenguaje de programación).

Para los nombre de variables se utiliza el estilo *lower camel case*.

1.2.4. Estándares de programación

Para el desarrollo del Software para filtrado y clasificación de imágenes de teledetección se utilizó el lenguaje IDL que en esencia es imperativo, aunque se lo puede utilizar como un lenguaje orientado a objetos. Para este trabajo solo se desarrollará de forma imperativa.

Todo el software se dividió en componentes según la funcionalidad que se requería, basado en el documento de especificación de software (RF.2).

El software se fue desarrollando por etapas para lo cual fue necesario un sistema de versionado. Para el desarrollo se utilizó un modelo en cascada iterativo combinado con prácticas de programación sugeridas por TDD. Es decir, se desarrollaron primero casos de test en forma unitaria, y luego la funcionalidad en sí.

1.2.5. Herramientas de desarrollo de software

Como ya se indicó anteriormente, el Software para filtrado y clasificación de imágenes de teledetección se desarrolló completamente en el lenguaje de programación IDL. Para el desarrollo de la aplicación se utilizó la herramienta IDL DE y para el versionado una herramienta online que permitió tener el código disponible en cualquier lugar y momento.

2. Especificaciones de diseño de componentes

2.1. Diseño

2.1.1. Diagrama de casos de uso

La figura 2.1.1.-a muestra un diagrama de casos de uso del Software para filtrado y clasificación de imágenes de teledetección. Aquí se ve claramente que este software permite realizar tres tipos de acciones distintas: generar ruido en una imagen, filtrar una imagen y realizar un análisis de las clasificaciones obtenidas.

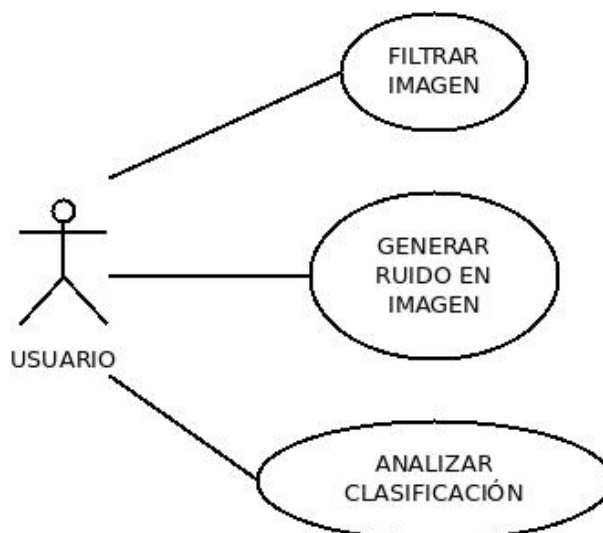


Figura 2.1.1.-a: Diagrama de casos de uso.

2.1.2. Diagrama de secuencia

Ya que hacer un diagrama de secuencias para cada funcionalidad sería tedioso y repetitivo, la figura 2.1.2.-b muestra un diagrama de secuencias que se podría aplicar a cualquiera de las funcionalidades que ofrece el Software para filtrado y clasificación de imágenes de teledetección.

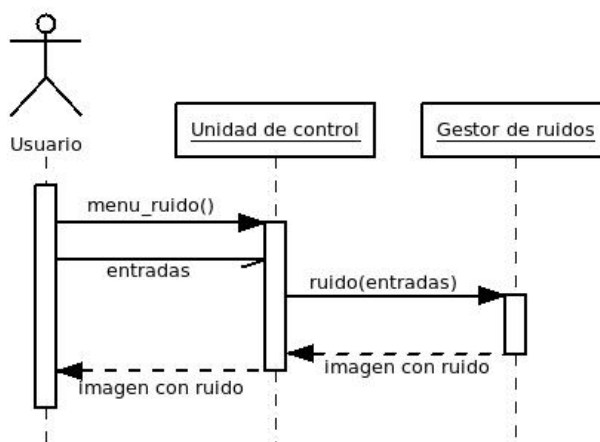


Figura 2.1.2.-b: Diagrama (general) de secuencia para la generación de ruido.

De manera general, el usuario elige que quiere hacer mediante el menú disponible. Luego inserta las entradas requeridas y la unidad de control llama al gestor de ruidos, el gestor de filtros o la unidad de posclasificación utilizando la función requerida por el usuario. El gestor devuelve una imagen con

ruido, una imagen filtrada o una matriz de confusión según el caso. Finalmente, la unidad de control, muestra el resultado de la operación.

2.2. Unidad de control

2.2.1. Tipo

Módulo.

2.2.2. Propósito

Agrupar todas las actividades que se relacionen con el manejo de la interfaz con el usuario. Realiza todas las funciones necesarias para la interacción entre ENVI y el Software para filtrado y clasificación de imágenes de teledetección en sí.

2.2.3. Función

Provee funciones para realizar acciones cuando el usuario elige que funcionalidad quiere ejecutar.

2.2.4. Subordinados

Los subordinados de este módulo son el resto de los módulos, es decir:

- Gestor de ruido
- Gestor de filtros
- Gestor de posclasificación

2.2.5. Dependencias

Depende exclusivamente de la aplicación ENVI.

2.2.6. Interfaces

Provee procedimientos que permiten interactuar con el cliente. Los más importantes son:

- NOISE_MENU_EVENT: permite la interacción con el módulo de ruido.
- FILTER_MENU_EVENT: permite la interacción con el módulo de filtros.
- CONFUSION_MATRIX_EVENT: permite la interacción con el módulo de posclasificación.

2.2.7. Procesamiento

Cuando se selecciona un botón en el menú de ENVI con una funcionalidad del Software para filtrado y clasificación de imágenes de teledetección, este módulo muestra ventanas para que el usuario pueda establecer los parámetros necesarios para su ejecución. Cuando todos los parámetros fueron recibidos, este módulo decide a que gestor invocar y con que función en particular, pasándole los parámetros correspondientes.

2.3. Gestor de ruido

2.3.1. Tipo

Módulo

2.3.2. Propósito

Agrupar todas las funcionalidades relacionadas con la generación de ruidos.

2.3.3. Función

Provee las interfaces necesarias para ejecutar distintos algoritmos que generen ruidos en imágenes de teledetección.

2.3.4. Interfaces

Este módulo provee tres interfaces:

- GENERATE_GAUSSIAN_NOISE: Genera ruido Gaussiano sobre la imagen.
- GENERATE_UNIFORM_NOISE: Genera ruido uniforme sobre la imagen.
- GENERATE_IMPULSE_NOISE: Genera ruido impulsivo sobre la imagen.

2.3.5. Procesamiento

Recorre la imagen agregándole ruido dependiendo del tipo que se eligió. Todo el ruido que se genera será en base a números aleatorios que dependen de la distribución que defina el tipo de ruido.

2.3.6. Datos

Todos los generadores de ruido requieren una imagen para la ejecución. Los parámetros que se deben establecer dependen del tipo de ruido. Para el caso del ruido Gaussiano se requiere la media y la desviación estándar. Para el ruido uniforme se debe indicar el rango en el cual se generará el ruido. Por último, la entrada del ruido impulsivo es la probabilidad de que se genere un punto claro o uno oscuro en la imagen.

2.4. Gestor de filtros

2.4.1. Tipo

Módulo

2.4.2. Propósito

Agrupar todas las funcionalidades relacionadas con el filtrado de imágenes de teledetección.

2.4.3. Función

Provee las interfaces necesarias para ejecutar distintos algoritmos de filtrado.

2.4.4. Interfaces

Las interfaces de este módulo se pueden dividir en dos grupos. Por un lado, tanto los filtros lineales como los filtros de orden están incluidos en la interfaz DO_FILTERING, la cual requiere que se le indique que filtro se quiere aplicar. Los filtros disponibles son filtro de media aritmética, de media geométrica, de media armónica, de media contrarmónica, Gaussiano, de mediana, de mínima, de máxima, olímpico, de media alfa recortada, de media de los k valores más cercanos y de mediana de los k valores más cercanos.

Por otro lado, se encuentran los filtros adaptativos donde se pueden identificar dos interfaces:

- DO_LOCAL_NOISE_REDUCTION_FILTER: Filtra la imagen con el algoritmo de filtrado de reducción de ruido local.
- DO_ADAPTIVE_MEDIAN_FILTER: Filtra la imagen con un filtro de mediana adaptativo.

2.4.5. Procesamiento

Recorre la imagen píxel por píxel con una ventana de filtrado con los vecinos de cada uno de ellos. Utiliza todos los datos de la ventana y realiza una operación con ellos que depende del filtro que se quiere aplicar. El resultado de esa operación es asignado al píxel de la imagen filtrada ubicada en la misma posición del píxel original.

2.4.6. Datos

Todos los filtros requieren una imagen y un tamaño de ventana de filtrado para la ejecución. La excepción es el filtro de mediana adaptativo que requiere el tamaño máximo de la ventana de filtrado. El resto de los parámetros que se deben establecer dependen del filtro.

Los filtros que no requieren parámetros extra son: filtro de media aritmética, de media geométrica, de media armónica, de mediana, de mínima, de máxima, olímpico y de mediana adaptativo.

El filtro de media contrarmónica requiere el orden, el Gaussiano la varianza, el de media alfa recortada el valor de alfa, los filtros de media y mediana de los k valores más cercanos el valor de k y el filtro de reducción de ruido local la varianza del ruido global.

2.5. Gestor de posclasificación

2.5.1. Tipo

Módulo

2.5.2. Propósito

Agrupar todas las funcionalidades relacionadas a la posclasificación, es decir, al análisis de una clasificación.

2.5.3. Función

Provee las interfaces necesarias para ejecutar procesos de posclasificación.

2.5.4. Interfaces

Este módulo ofrece dos interfaces:

- CALCULATE_CONFUSION_MATRIX: Calcula la matriz de confusión entre dos clasificaciones.
- CALCULATE_KAPPA_COEFFICIENT: Calcula el coeficiente kappa de una matriz de confusión.

2.5.5. Procesamiento

Para calcular la matriz de confusión compara las dos clasificaciones y genera la matriz en base a esta comparación. Para el cálculo del coeficiente de kappa lo realiza a partir de la diagonal de la matriz de confusión.

2.5.6. Datos

El cálculo de la matriz de confusión requiere dos clasificaciones, y el cálculo del coeficiente kappa una matriz de confusión.

APÉNDICE D

MANUAL DE USUARIO DEL SOFTWARE

Manual de Usuario del Software (SUM)

Software para filtrado y clasificación de imágenes de teledetección

Autor: Mariano E. Della Vedova

Fecha: 19 de Agosto de 2013

Versión: 1.0

HISTORIAL DE CAMBIOS

Versión del Documento	Fecha	Sección	Razón del cambio o revisión
1.0	19/08/2013	Todo	Primera revisión

TABLA DE CONTENIDOS

1	Introducción	4
1.1	Destinatarios	4
1.2	Aplicabilidad	4
1.3	Propósito	4
1.4	Cómo usar este documento	4
1.5	Referencias	4
2	Descripción general	5
3	Sección de instalación	5
4	Descripción funcional	5
4.1	Generación de ruido	6
4.2	Filtros lineales	8
4.3	Filtros de orden	9
4.4	Filtros adaptativos	10
4.5	Posclasificación	11

1. Introducción

1.1. Destinatarios

Este software está orientado al estudio de los resultados que se pueden obtener en una clasificación de imágenes de teledetección, aplicando previamente, o no, un filtro para reducir el ruido que esta tiene. El usuario de esta software debe tener amplios conocimientos de sobre tratamientos de imágenes de teledetección. También es recomendable que el usuario tenga conocimientos sobre la aplicación ENVI, ya que éste es un software adicional para dicha aplicación.

Por otro lado, este no pretende ser un manual sobre cómo usar ENVI, por lo que el usuario debería tener conocimientos previos sobre esta aplicación.

1.2. Aplicabilidad

Este manual de usuario es aplicable a la versión 1.0 del Software para filtrado y clasificación de imágenes de teledetección, para la generación de errores, filtrado y posclasificación.

1.3. Propósito

El propósito de este manual es explicar la utilización del Software para filtrado y clasificación de imágenes de teledetección para poder obtener los resultados esperados por el usuario. Como se dijo anteriormente, este manual no pretende explicar la utilización de ENVI, sino las funcionalidades que se le agregan a ésta.

El propósito del software es agregar funcionalidades a la aplicación ENVI con el objetivo de poder agregar diferentes tipo de ruidos a imágenes, utilizar diferentes filtros a imágenes con ruido y comparar dos clasificaciones tomando un como referencia una como la ideal.

1.4. Cómo usar este documento

Este documento tiene la intención de ser una guía para el usuario del Software para filtrado y clasificación de imágenes de teledetección, por lo que no se va a explicar cómo se debe utilizar ENVI. Tampoco se describen para qué sirve cada una de las funcionalidades sino, más bien, está orientado a mostrar cómo se ejecuta cada una de ellas.

En este documento se puede encontrar una descripción general del Software para filtrado y clasificación de imágenes de teledetección en la sección 2, que explica a grandes rasgos que funcionalidades pone a disposición este software. En la sección 3 se muestra los comandos que se deben ejecutar para instalar el software. Finalmente, la sección 4 detalla cómo se ejecuta cada una de las funcionalidades que se ofrecen.

1.5. Referencias

ID	Título	Versión	Fecha
RF.1	Documento de Requerimientos de Usuario (URD) - Software para filtrado y clasificación de imágenes de teledetección	1.1	10/08/2013
RF.2	Documento de Especificación de Software (SSD) - Software para filtrado y clasificación de imágenes de teledetección	1.1	10/08/2013
RF.3	Documento de Diseño Detallado (DDD) - Software para filtrado y clasificación de imágenes de teledetección	1.0	25/08/2013

Continúa en la página siguiente ...

Tabla 1.5.0.-a – Continuación

ID	Título	Versión	Fecha
RF.4	Guide to applying the ESA software engineering standards to small software projects	1.0	08/05/1996

Tabla 1.5.0.-a: Documentos de Referencia

2. Descripción general

El Software para filtrado y clasificación de imágenes de teledetección fue diseñado con el fin de hacer la aplicación ENVI una herramienta más específica para el estudio que se desea realizar. Es decir, este software fue diseñado para poder estudiar la factibilidad de realizar clasificaciones correctas a partir de imágenes con errores. Por lo que este software agrega a la aplicación ENVI, funcionalidades tales como generar sobre una imagen ruidos Gaussiano, uniforme e impulsivo; filtrar imágenes con filtros lineales (filtro de media aritmética, de media geométrica, de media armónica, de media contrarmónica y Gaussiano), filtros de orden (filtro de mediana, de máxima, de mínima, olímpico, de media alfa recortada, de media de los k valores más cercanos y de mediana de los k valores más cercanos) y filtros adaptativos (filtro de reducción de ruido local y de mediana adaptativo); y comparar dos clasificaciones tomando una como referencia como si fuese la real o "ideal". La comparación entre dos clasificaciones se realiza por medio de una matriz de confusión y su correspondiente coeficiente Kappa. La función de clasificación no están incluidas dentro del Software para filtrado y clasificación de imágenes de teledetección, ya que esto está provisto por la aplicación ENVI.

Estas funcionalidades, al quedar enmarcadas dentro de la aplicación ENVI, se ejecutan de manera gráfica, por lo que el usuario podrá ver el resultado inmediatamente e inclusive podrá combinarlos con otras funciones que vienen incluidas en la aplicación. En general, las funcionalidades que aporta el Software para filtrado y clasificación de imágenes de teledetección requiere de imágenes de teledetección (en cualquier formato soportado por ENVI) y sus correspondientes parámetros de entrada. La excepción es la comparación entre clasificaciones, que, como es de esperar, requiere de dos clasificaciones (en el formato establecido para la aplicación ENVI) para ser ejecutada.

3. Sección de instalación

Para instalar el Software para filtrado y clasificación de imágenes de teledetección se requiere que la aplicación ENVI esté instalada previamente. Una vez logrado esto se debe identificar el directorio donde ésta fue instalada y simplemente se debe ejecutar el script de instalación de la siguiente manera:

```
sh install.sh ENVI_instalation_path
```

donde *ENVI_instalation_path* es el path donde se encuentra instalada la aplicación ENVI. Esto genera un cambio en el menú principal de la aplicación ENVI, ofreciendo nuevas funcionalidades.

4. Descripción funcional

Como se mencionó anteriormente el Software para filtrado y clasificación de imágenes de teledetección es una batería de nuevas funcionalidades que se le agregan a ENVI. Entre estas funcionalidades se encuentran generadores de ruido sobre imágenes, filtros para imágenes y métodos para comparar dos clasificaciones tomando una de éstas como referencia.

Para poder utilizar cualquiera de estas funcionalidades, el usuario deberá ir a la sección *Soficlaite* del menú principal de ENVI que se muestra en la figura 4.0.0.-a.

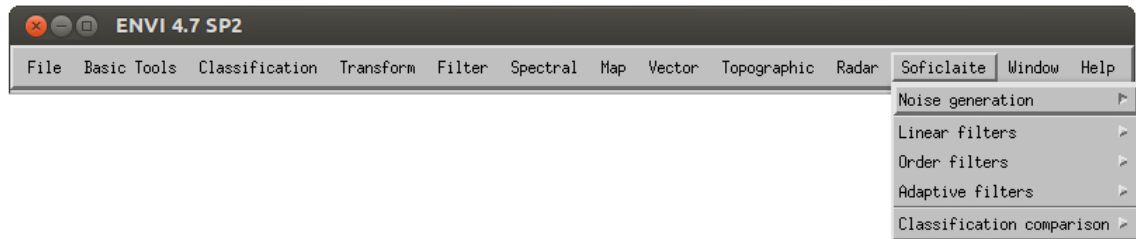


Figura 4.0.0.-a: Menú del Software para filtrado y clasificación de imágenes de teledetección dentro de ENVI.

Como se puede ver en esta figura, las funcionalidades están divididas en generación de ruido (noise generation), filtros lineales (linear filters), filtros de orden (order filters), filtros adaptativos (adaptive filters) y posclasificación (postclassification). A continuación se describen las funcionalidades contenidas en cada uno de estas secciones.

4.1. Generación de ruido

El Software para filtrado y clasificación de imágenes de teledetección permite generar tres tipos de ruido, como se muestra en la figura 4.1.0.-a: ruido Gaussiano (Gaussian noise), ruido uniforme (uniform noise) y ruido impulsivo (impulsive noise).

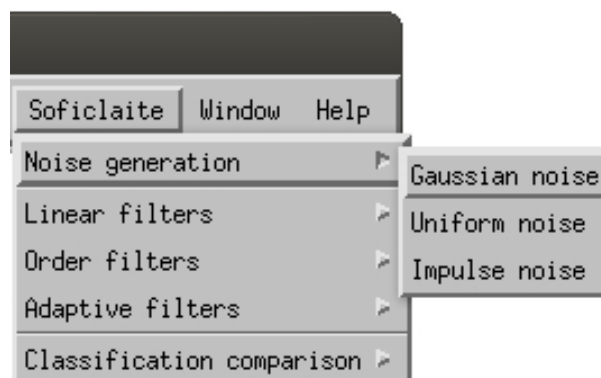


Figura 4.1.0.-a: Menú para la generación de ruido.

Al seleccionar alguno de estos tres tipos de ruido, el usuario debe elegir que imagen quiere alterar. Para esto se despliega un menú clásico de ENVI donde se permite utilizar una imagen que esté en memoria, una nueva imagen, una fracción de una imagen, bandas particulares de una imagen, etc. La figura 4.1.0.-b muestra este menú.

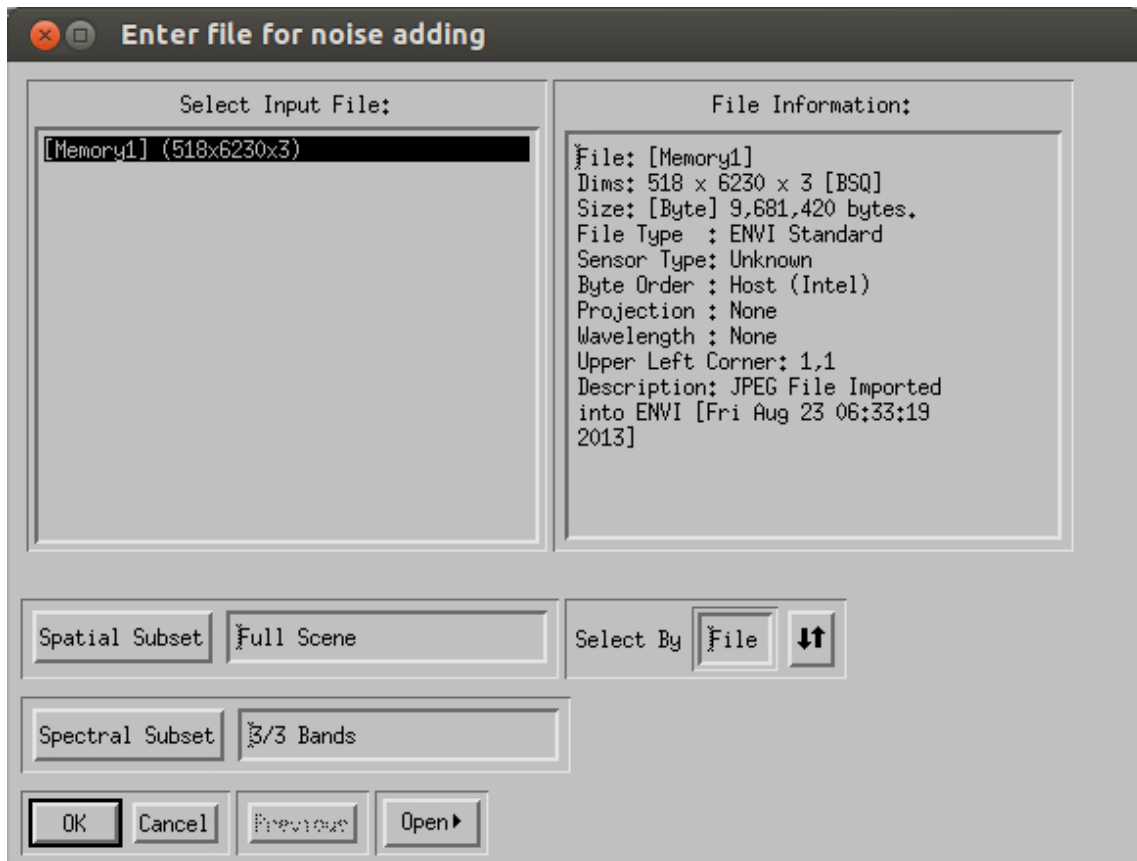


Figura 4.1.0.-b: Menú para abrir la imagen a modificar.

Una vez elegida la imagen que se quiere modificar, cada tipo de ruido tiene parámetros particulares que deben ser provistos, como se muestra en la figura 4.1.0.-c. El ruido Gaussiano requiere que se introduzca la media (mean) y la desviación estandar (standard deviation) del ruido a generar (figura 4.1.0.-c -izq.-). El ruido uniforme permite introducir el rango (range start y range end) en el que se encontrará en ruido agregado a la imagen (figura 4.1.0.-c -centro.-). Se debe tener en cuenta que el parámetro que determina el final del rango debe ser mayor o igual al comienzo; caso contrario la aplicación muestra un mensaje de error. Por último, el ruido impulsivo recibe como parámetros la probabilidad de que un píxel erróneo sea blanco -o “sal”- (white pixel probability), o negro -o “pimienta”- (black pixel probability). Para este tipo de ruido existen dos restricciones que si no se cumplen la aplicación muestra un error: ambas probabilidades deben estar en el rango entre 0 y 1, y la suma de ambas probabilidades no debe superar la unidad.

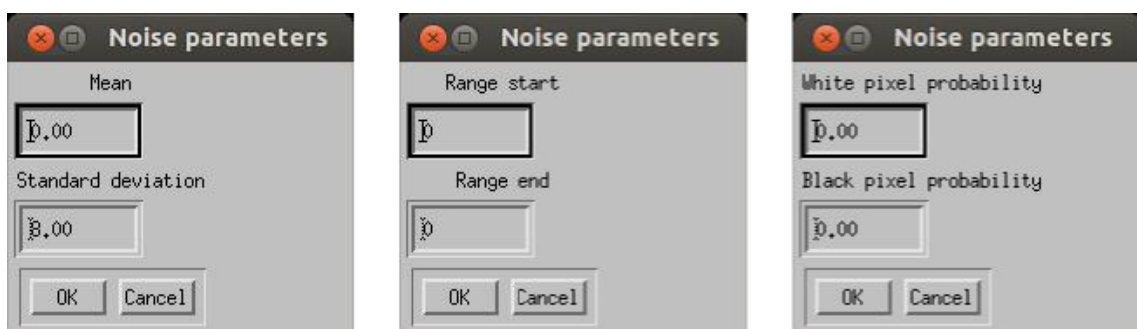


Figura 4.1.0.-c: Menú para introducir los parámetros para generar ruido Gaussiano (izquierda), uniforme (centro) e impulsivo (derecha).

Todas estas operaciones generan distintos tipos de alteraciones sobre la imagen elegida. Con el

ruido Gaussiano se puede apreciar una diferencia en la intensidad de los píxeles, algunos aparece más claros y otros más oscuros. Con el ruido uniforme todos los píxeles se ven más oscuros. Con el ruido impulsivo se observan píxeles blancos y negros distribuidos por toda la imagen.

4.2. Filtros lineales

Como se mostró en la figura 4.0.0.-a, en un segundo nivel del menu del Software para filtrado y clasificación de imágenes de teledetección se encuentran filtros lineales. Como se muestra en la figura 4.2.0.-a, los filtros lineales disponibles son: filtro de media aritmética, filtro de media geométrica, filtro de media armónica, filtro de media contrarmónica y filtro gaussiano.

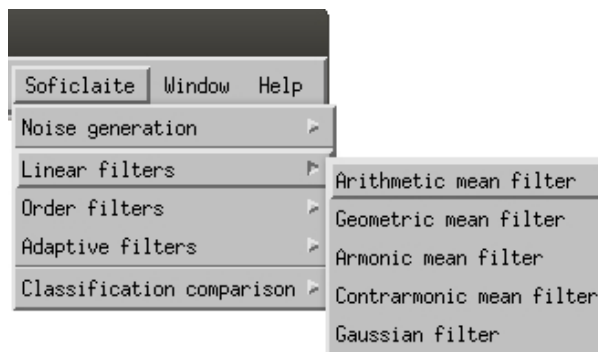


Figura 4.2.0.-a: Menú de los filtros lineales.

Como ocurre con los procesos de generación de ruido, un menú para la selección de una imagen surge cuando el usuario elige qué filtro quiere aplicar. El menú es similar al que se muestra en la figura 4.1.0.-b.

Una vez elegida la imagen, se deben establecer los parámetros particulares para cada filtro. Todos los filtros requieren el tamaño de la ventana (filter mask size) que se utilizará para el filtrado como se muestra en la figura 4.2.0.-b. Una aclaración que se debe hacer en este punto es que todas las ventanas de filtrado son cuadradas y su tamaño debe cumplir que sea positivo, menor al tamaño de la imagen e impar.

En la figura 4.2.0.-b se puede ver a la izquierda el menú que se muestra al elegir los filtros de media aritmética, geométrica y armónica, ya que no necesitan más parámetros para su ejecución que el tamaño de la ventana. Distintos son los casos de los filtros de media contrarmónica y filtro gaussiano que requieren el orden del filtro (filter order) y la varianza (filter variance), respectivamente. Como es de esperar, la varianza debe ser positiva.

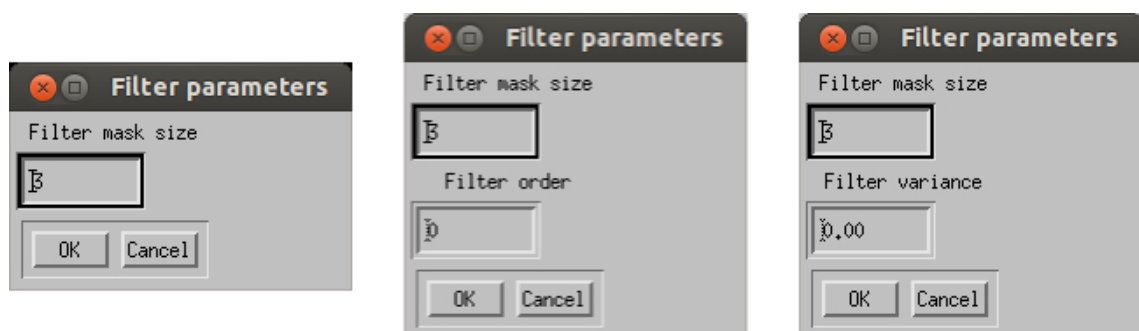


Figura 4.2.0.-b: Menú para introducir los parámetros para los filtros de media aritmética, filtro de media geométrica, filtro de media armónica (izquierda), filtro de media contrarmónica (centro) y filtro gaussiano (derecha).

Todos estos filtros tienden a suavizar las imágenes, siendo el filtro de media aritmética el más agresivo

a los bordes. En general, mientras más grande es la ventana de filtrado, mayor es la pérdida de detalle de los bordes.

4.3. Filtros de orden

La figura 4.3.0.-a muestra los filtros de orden disponibles con el Software para filtrado y clasificación de imágenes de teledetección. Estos filtros son: filtro de mediana (median filter), de mínima (minimum filter), de máxima (maximum filter), olímpico (olimpic filter), de media alfa recortada (alpha-trimmed mean filter), de media de los k valores más cercanos (k-nearest mean filter) y de mediana de los k valores más cercanos (k-nearest median filter).

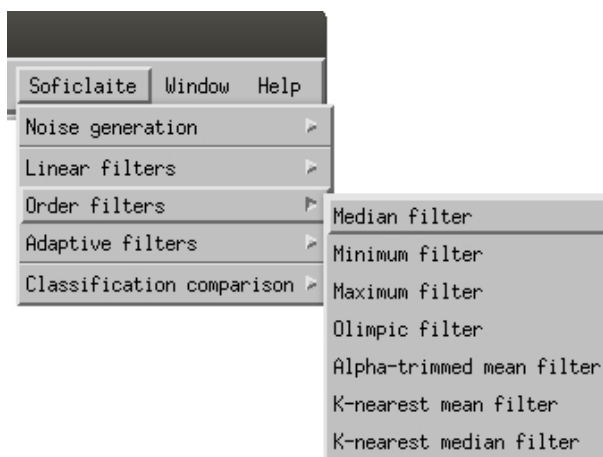


Figura 4.3.0.-a: Menú de los filtros de orden.

Al igual que los filtros lineales, los filtros de orden utilizan una ventana para realizar el filtrado, con las mismas restricciones de tamaño. La figura 4.3.0.-b muestra el menú para intertar parámetros para los diferentes tipos de filtros de orden. A la izquierda se muestra el menú para los filtros de mediana, de mínima, de máxima y olímpico, ya que estos solo requieren el tamaño de la ventana. Como se mencionó anteriormente, la ventana tiene las mismas características que las de los filtros lineales. Al centro se encuentra el menú del filtro de media alfa recortada y como es de esperar el parámetro extra que se debe establecer es el de *alfa*. Este valor debe estar en el rango entre 0 y el tamaño de la ventana. Finalmente, a la derecha de la figura 4.3.0.-b se muestra como se establecen los parámetros de los filtros de media y de mediana de los k valores más cercanos. Aquí, además del tamaño de la ventana, se debe proveer el valor de k que debe ser positivo y no puede ser mayor la cantidad de elementos que entran en la ventana.

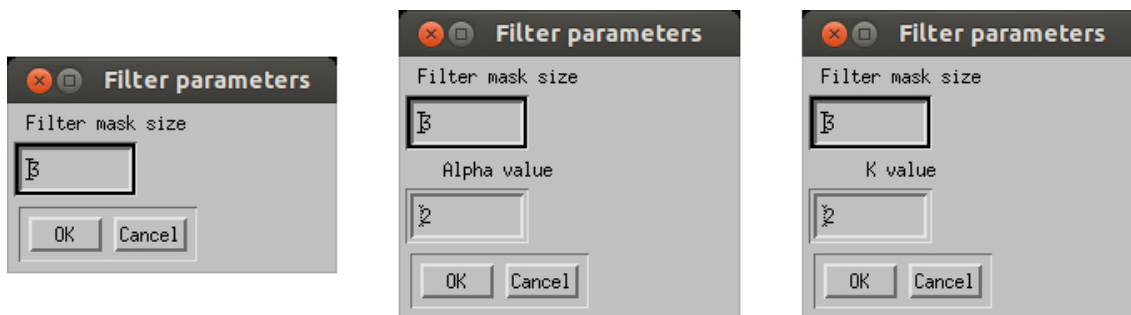


Figura 4.3.0.-b: Menú para introducir los parámetros para los filtros de mediana, de mínima, de máxima, olímpico (izquierda), de media alfa recortada (centro), de media de los k valores más cercanos y de mediana de los k valores más cercanos (derecha).

Estos filtros suelen ser menos agresivos que los filtros lineales, aunque se pueden perder detalles como puntos y líneas.

4.4. Filtros adaptativos

El último menú de filtros se muestra en la figura 4.4.0.-a y contiene dos tipos de filtros adaptativos: el filtro de reducción de ruido local (local noise reduction filter) y el filtro de mediana adaptativo (adaptive median filter).

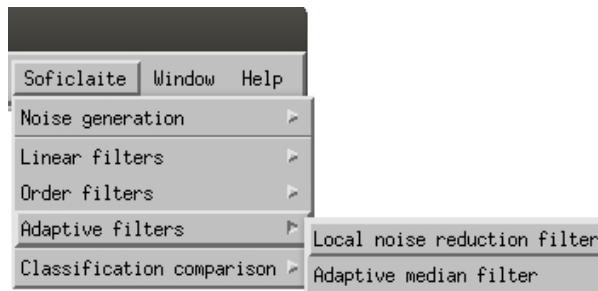


Figura 4.4.0.-a: Menú de los filtros adaptativos.

El filtro de reducción de ruido local requiere como parámetros de entrada el tamaño de la ventana (filter mask size), cuyas características son las mismas a los filtros antes mencionados, y una estimación de la varianza del ruido global (global noise variance). El menú para insertar estos parámetros se muestra en la figura 4.4.0.-b (izquierda) y aparece luego de que el usuario seleccione una imagen a filtrar mediante un menú del mismo estilo que el de la figura 4.1.0.-b. Como es de esperar, la varianza debe ser positiva.

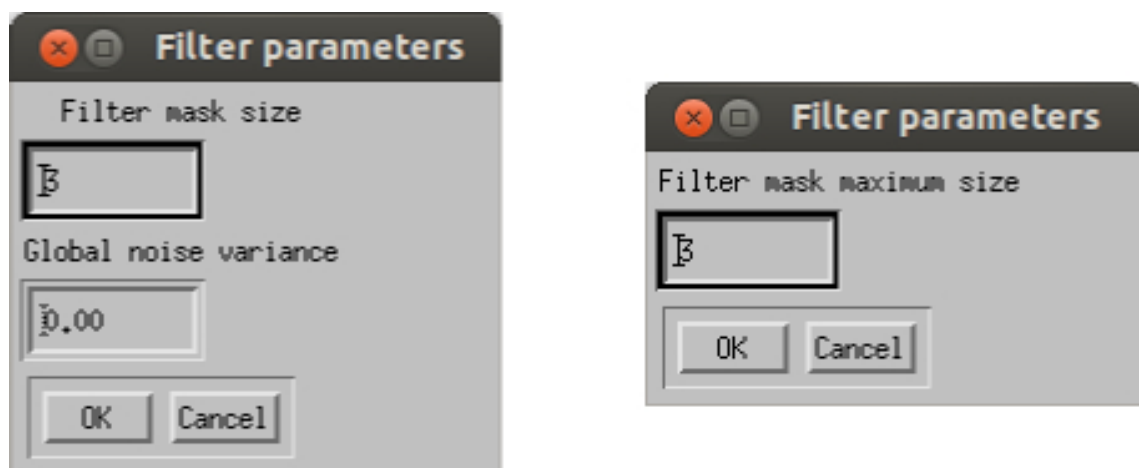


Figura 4.4.0.-b: Menú para introducir los parámetros para el filtro de reducción de ruido local (izquierda) y el filtro de mediana adaptativo (derecha).

A la derecha de la figura 4.4.0.-b se muestra la forma de establecer el parámetro que necesita el filtro de mediana adaptativo. A diferencia del resto de los filtros, éste requiere un tamaño máximo de ventana (filter mask maximum size), se debe tener en cuenta que este tamaño no debe superar el tamaño de la imagen.

Estos filtros intentan mejorar el rendimiento, perdiendo menos detalle.

4.5. Posclasificación

En el último menú (postclassification), se puede encontrar el procedimiento para calcular la matriz de confusión, como se ve en la figura 4.5.0.-a.

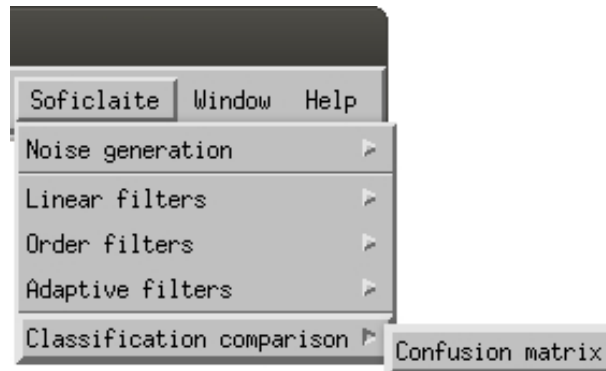


Figura 4.5.0.-a: Menú para funcionalidades posclasificación.

Esta funcionalidad requiere dos clasificaciones mediante dos ventanas similares a la que se muestra en la figura 4.1.0.-b. Al concluir este cálculo se muestra la matriz de confusión y el coeficiente kappa. Para que la ejecución sea correcta las dos clasificaciones elegidas deben tener el mismo tamaño.

APÉNDICE E _____

PLAN DE VERIFICACIÓN Y VALIDACIÓN DEL SOFTWARE

Plan de Verificación y Validación del Software (SVVP)

Software para filtrado y clasificación de imágenes de teledetección

Autor: Mariano E. Della Vedova

Fecha: 25 de Agosto de 2013

Versión: 1.0

HISTORIAL DE CAMBIOS

Versión del Documento	Fecha	Sección	Razón del cambio o revisión
1.0	25/08/2013	Todo	Primera revisión

TABLA DE CONTENIDOS

1	Introducción	5
1.1	Propósito de este documento	5
1.2	Definiciones, acrónimos y abreviaciones	5
1.3	Referencias	5
1.4	Resumen del documento	5
2	Plan de pruebas	5
2.1	Items a probar	5
2.2	Características a probar	6
2.3	Productos finales para las pruebas	6
2.4	Tareas previas a la ejecución de pruebas	6
2.5	Necesidades de ambiente	7
3	Especificaciones de casos de prueba	7
3.1	Caso de prueba SFCIT-001	7
3.1.1	Procedimiento de prueba	7
3.2	Caso de prueba SFCIT-002	7
3.2.1	Procedimiento de prueba	8
3.3	Caso de prueba SFCIT-003	8
3.3.1	Procedimiento de prueba	8
3.4	Caso de prueba SFCIT-004	8
3.4.1	Procedimiento de prueba	9
3.5	Caso de prueba SFCIT-005	9
3.5.1	Procedimiento de prueba	9
3.6	Caso de prueba SFCIT-006	9
3.6.1	Procedimiento de prueba	10
3.7	Caso de prueba SFCIT-007	10
3.7.1	Procedimiento de prueba	10
3.8	Caso de prueba SFCIT-008	10
3.8.1	Procedimiento de prueba	11
3.9	Caso de prueba SFCIT-009	11
3.9.1	Procedimiento de prueba	11
3.10	Caso de prueba SFCIT-010	11
3.10.1	Procedimiento de prueba	12
3.11	Caso de prueba SFCIT-011	12
3.11.1	Procedimiento de prueba	12
3.12	Caso de prueba SFCIT-012	12
3.12.1	Procedimiento de prueba	13
3.13	Caso de prueba SFCIT-013	13
3.13.1	Procedimiento de prueba	13
3.14	Caso de prueba SFCIT-014	13
3.14.1	Procedimiento de prueba	14
3.15	Caso de prueba SFCIT-015	14
3.15.1	Procedimiento de prueba	14
3.16	Caso de prueba SFCIT-017	14
3.16.1	Procedimiento de prueba	15
3.17	Caso de prueba SFCIT-018	15
3.17.1	Procedimiento de prueba	15
3.18	Caso de prueba SFCIT-019	15
3.18.1	Procedimiento de prueba	16

4	Reporte de ejecución de prueba	16
4.1	Plantilla de reporte de test	16
4.2	Registro de resultados	16
4.3	Métricas de calidad del Plan de Prueba	17
4.4	Métricas de calidad del Plan de Prueba	18

1. Introducción

1.1. Propósito de este documento

Este documento describe el plan de pruebas y los procedimientos definidos para realizar la verificación y validación del Software para filtrado y clasificación de imágenes de teledetección.

1.2. Definiciones, acrónimos y abreviaciones

Acrónimo	Descripción
SVVP	Software Verification and Validation Plan

Tabla 1.2.0.-a: Lista de definiciones, acrónimos y abreviaturas

1.3. Referencias

ID	Título	Versión	Fecha
RF.1	Documento de Requerimientos de Usuario (URD) - Software para filtrado y clasificación de imágenes de teledetección	1.1	10/08/2013
RF.2	Documento de Especificación de Software (SSD) - Software para filtrado y clasificación de imágenes de teledetección	1.1	10/08/2013
RF.3	Documento de Diseño Detallado (DDD) - Software para filtrado y clasificación de imágenes de teledetección	1.0	25/08/2013
RF.4	Manual de Usuario de Software (SUM) - Software para filtrado y clasificación de imágenes de teledetección	1.0	19/08/2013
RF.5	Guide to applying the ESA software engineering standards to small software projects	1.0	08/05/1996

Tabla 1.3.0.-a: Documentos de Referencia

1.4. Resumen del documento

2. Plan de pruebas

En esta sección se presenta una breve descripción del plan de pruebas correspondiente al Software para filtrado y clasificación de imágenes de teledetección.

2.1. Items a probar

Los items principales que deben ser probados son:

- Generación de ruido Gaussiano.
- Generación de ruido uniforme.
- Generación de ruido impulsivo.

- Filtro de media aritmética.
- Filtro de media geométrica.
- Filtro de media armónica.
- Filtro de media contrarmónica.
- Filtro Gaussiano.
- Filtro de mediana.
- Filtro de mínima.
- Filtro de máxima.
- Filtro olímpico.
- Filtro de media alfa recortada.
- Filtro de media de los k valores más cercanos.
- Filtro de mediana de los k valores más cercanos.
- Filtro de reducción de ruido local.
- Filtro de mediana adaptativo.
- Cálculo de la matriz de confusión.
- Cálculo del coeficiente kappa.

2.2. Características a probar

Las características que se deben probar incluyen:

- Obtención de imagen o de datos de clasificación según corresponda.
- Ingreso de parámetros.
- Visualización y guardado de resultados obtenidos.
- Ejecución del software en el sistema operativo linux.

2.3. Productos finales para las pruebas

Para comenzar con la etapa de prueba, se debe disponer del instalador del Software para filtrado y clasificación de imágenes de teledetección.

Como resultado esta etapa producirá un reporte de prueba basado en la plantilla que se describe en la sección 4.1. También se deben generar la métricas según se explica en la sección 4.3.

2.4. Tareas previas a la ejecución de pruebas

Para realizar la tarea de ejecución de pruebas se debe instalar previamente la aplicación ENVI y posteriormente el Software para filtrado y clasificación de imágenes de teledetección según se explica en el manual de usuario (RF.4).

2.5. Necesidades de ambiente

El único requisito de ambiente es que está instalado el sistema operativo Linux.

3. Especificaciones de casos de prueba

Esta sección describe los casos de prueba y sus correspondientes procedimientos, definidos con el objetivo de verificar y validar el cumplimiento de los requerimientos del trabajo comprobables mediante la ejecución de pruebas.

3.1. Caso de prueba SFCIT-001

ID	SFCIT-001
Items de prueba	Generación de ruido Gaussiano
Entradas	Imagen <i>bhtmref.img</i> , media igual a 0 y desviación estándar igual a 2
Salida	Imagen con ruido Gaussiano
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.1.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Generación de ruido Gaussiano	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Noise generation</i> y presionar sobre <i>Gaussian noise</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>bhtmref.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer la media en 0 y la desviación estándar en 2.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen original utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 0.

3.2. Caso de prueba SFCIT-002

ID	SFCIT-002
Items de prueba	Generación de ruido uniforme
Entradas	Imagen <i>bhtmref.img</i> y un rango desde 0 a 10
Salida	Imagen con ruido uniforme
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.2.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Generación de ruido uniforme	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Noise generation</i> y presionar sobre <i>Uniform noise</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>bhtmref.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el comienzo de rango en 0 y el fin del rango en 10.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen original utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 0.

3.3. Caso de prueba SFCIT-003

ID	SFCIT-003
Items de prueba	Generación de ruido impulsivo
Entradas	Imagen <i>bhtmref.img</i> y probabilidades de error 0.1 en ambos casos
Salida	Imagen con ruido impulsivo
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.3.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Generación de ruido impulsivo	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Noise generation</i> y presionar sobre <i>Impulse noise</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>bhtmref.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer las probabilidades de píxel blanco y de píxel negro en 0.1.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen original utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 0.

3.4. Caso de prueba SFCIT-004

ID	SFCIT-004
Items de prueba	Filtrado de imagen mediante un filtro de media aritméticas
Entradas	Imagen <i>uniform_noise.img</i> y tamaño de máscara igual a 3
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.4.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de media aritmética	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Linear filters</i> y presionar sobre <i>Arithmetic mean filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>uniform_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>arithmetic_mean_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.5. Caso de prueba SFCIT-005

ID	SFCIT-005
Items de prueba	Filtrado de imagen mediante un filtro de media geométrica
Entradas	Imagen <i>gaussian_noise.img</i> y tamaño de máscara igual a 3
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.5.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de media geométrica	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Linear filters</i> y presionar sobre <i>Geometric mean filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>geometric_mean_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.6. Caso de prueba SFCIT-006

ID	SFCIT-006
Items de prueba	Filtrado de imagen mediante un filtro de media armónica
Entradas	Imagen <i>gaussian_noise.img</i> y tamaño de máscara igual a 3
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.6.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de media armónica	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Linear filters</i> y presionar sobre <i>Armonic mean filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>armonic_mean_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.7. Caso de prueba SFCIT-007

ID	SFCIT-007
Items de prueba	Filtrado de imagen mediante un filtro de media contrarmónica
Entradas	Imagen <i>gaussian_noise.img</i> , tamaño de máscara igual a 3 y orden igual a -1
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.7.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de media contrarmónica	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Linear filters</i> y presionar sobre <i>Contrarmonic mean filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3 y el orden igual a -1.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>contrarmonic_mean_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.8. Caso de prueba SFCIT-008

ID	SFCIT-008
Items de prueba	Filtrado de imagen mediante un filtro Gaussiano
Entradas	Imagen <i>gaussian_noise.img</i> , tamaño de máscara igual a 3 y la varianza igual a 1.5
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.8.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro gaussiano	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Linear filters</i> y presionar sobre <i>Gaussian filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3 y la varianza igual a 1.5.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>gaussian_filter.img</i> utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 1.

3.9. Caso de prueba SFCIT-009

ID	SFCIT-009
Items de prueba	Filtrado de imagen mediante un filtro de mediana
Entradas	Imagen <i>impulse_noise.img</i> y tamaño de máscara igual a 3
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.9.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de mediana	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Order filters</i> y presionar sobre <i>Median filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>impulse_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>median_filter.img</i> utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 1.

3.10. Caso de prueba SFCIT-010

ID	SFCIT-010
Items de prueba	Filtrado de imagen mediante un filtro de mínima
Entradas	Imagen <i>gaussian_noise.img</i> y tamaño de máscara igual a 3
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.10.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de mínima	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Order filters</i> y presionar sobre <i>Minimum filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>minimum_filter.img</i> utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 1.

3.11. Caso de prueba SFCIT-011

ID	SFCIT-011
Items de prueba	Filtrado de imagen mediante un filtro de máxima
Entradas	Imagen <i>gaussian_noise.img</i> y tamaño de máscara igual a 3
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.11.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de máxima	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Order filters</i> y presionar sobre <i>Maximum filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>maximum_filter.img</i> utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 1.

3.12. Caso de prueba SFCIT-012

ID	SFCIT-012
Items de prueba	Filtrado de imagen mediante un filtro olímpico
Entradas	Imagen <i>gaussian_noise.img</i> y tamaño de máscara igual a 3
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.12.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro olímpico	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Order filters</i> y presionar sobre <i>Olimpic filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>olimpic_filter.img</i> utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 1.

3.13. Caso de prueba SFCIT-013

ID	SFCIT-013
Items de prueba	Filtrado de imagen mediante un filtro de media alfa recortada
Entradas	Imagen <i>impulse_noise.img</i> , tamaño de máscara igual a 3 y alfa igual a 6
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.13.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de media alfa recortada	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Order filters</i> y presionar sobre <i>Alpha-trimmed mean filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>impulse_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3 y alfa en 6.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>alpha-trimmed_mean_filter.img</i> utilizando la función <code>ARRAY_EQUAL</code> .	La función devuelve el valor 1.

3.14. Caso de prueba SFCIT-014

ID	SFCIT-014
Items de prueba	Filtrado de imagen mediante un filtro de media de los k valores más cercanos
Entradas	Imagen <i>uniform_noise.img</i> , tamaño de máscara igual a 3 y k igual a 6
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.14.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de media de los k valores más cercanos	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Order filters</i> y presionar sobre <i>K-nearest mean filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>uniform_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3 y k en 6.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>k-nearest_mean_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.15. Caso de prueba SFCIT-015

ID	SFCIT-015
Items de prueba	Filtrado de imagen mediante un filtro de mediana de los k valores más cercanos
Entradas	Imagen <i>impulse_noise.img</i> , tamaño de máscara igual a 3 y k igual a 6
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.15.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de mediana de los k valores más cercanos	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Order filters</i> y presionar sobre <i>K-nearest median filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>impulse_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3 y k en 6.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>k-nearest_median_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.16. Caso de prueba SFCIT-017

ID	SFCIT-017
Items de prueba	Filtrado de imagen mediante un filtro reducción de ruido local
Entradas	Imagen <i>gaussian_noise.img</i> , tamaño de máscara igual a 3 y varianza del ruido global igual a 4
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.16.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de reducción de ruido local	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Adaptive filters</i> y presionar sobre <i>Local noise reduction filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>gaussian_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño de la máscara en 3 y la varianza del ruido global en 4.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>local_noise_reduction_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.17. Caso de prueba SFCIT-018

ID	SFCIT-018
Items de prueba	Filtrado de imagen mediante un filtro de mediana adaptativo
Entradas	Imagen <i>impulse_noise.img</i> y tamaño máximo de máscara igual a 7
Salida	Imagen filtrada
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.17.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Filtrado utilizando filtro de mediana adaptativo	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Adaptive filters</i> y presionar sobre <i>Adaptive median filter</i> .	Se muestra un menú para la elección de la imagen.
A.2	Elegir la imagen <i>impulse_noise.img</i> .	Se despliega un menú para insertar los parámetros.
A.3	Establecer el tamaño máximo de la máscara en 7.	Se muestra un menú donde se muestra que la imagen generada se encuentra en memoria.
A.4	Comparar la imagen obtenida con la imagen <i>adaptive_median_filter.img</i> utilizando la función ARRAY_EQUAL.	La función devuelve el valor 1.

3.18. Caso de prueba SFCIT-019

ID	SFCIT-019
Items de prueba	Comparación entre dos clasificaciones mediante una matriz de confusión y el correspondiente coeficiente kappa
Entradas	Clasificaciones <i>original_classification</i> y <i>arithmetic_classification</i>
Salida	Matriz de confusión y coeficiente kappa
Requerimientos de ambiente	ENVI con el Software para filtrado y clasificación de imágenes de teledetección instalado en un sistema operativo Linux

3.18.1. Procedimiento de prueba

Paso	Descripción	Salida esperada
A	Detalle: Cálculo de matriz de confusión y coeficiente kappa	
A.1	Entrar al menú <i>Soficlaite</i> , luego a <i>Postclasificación</i> y presionar sobre <i>confusion matrix</i> .	Se muestra un menú para la elección de la clasificación.
A.2	Elegir la clasificación <i>original_classification</i> .	Se muestra un menú para la elección de la segunda clasificación.
A.3	Elegir la clasificación <i>arithmetic_classification</i> .	Se muestra una pantalla con los resultados de los cálculos de la matriz de confusión igual a [[48519,4254],[2534,204793]] y coeficiente kappa igual a 0.918324.

4. Reporte de ejecución de prueba

Este capítulo presenta una descripción de la forma en que se llevará el reporte de los resultados de las ejecuciones de los casos de test.

4.1. Plantilla de reporte de test

El procedimiento de pruebas es una lista cronológica de los pasos a ser ejecutados por cada test. Para reportar la ejecución de un procedimiento de prueba, se adopta la forma de plantilla que muestra la tabla 4.1.0.-a donde cada paso es especificado, como así también el resultado esperado, el resultado obtenido y comentarios que se quieran hacer al respecto

Test ID: Detalle:				
Paso	Descripción	Salida esperada	Salida obtenida	Comentarios
A.1				
A.2				
Fecha:	Firma del responsable de prueba:			

Tabla 4.1.0.-a: Plantilla de reporte de prueba.

4.2. Registro de resultados

Para facilitar el monitoreo de pruebas en curso y tener una clara evidencia de los resultados de las mismas, las plantillas de datos de prueba preparadas para detallar los pasos de procedimiento definirán el resultado esperado de cada paso. Cada prueba se llevará a cabo siguiendo los pasos de secuencia especificados por el procedimiento, y el resultado de cada paso se comparará con el resultado esperado.

En caso de discrepancia de los resultados obtenidos con respecto a lo previsto, una solución inmediata de problemas será realizada por el operador de prueba para verificar el motivo del problema ocurrido. El responsable de la prueba evaluará el impacto del problema y la oportunidad de continuar con la ejecución de la prueba o de suspenderla esperando un análisis más profundo o adecuado para las acciones de recuperación.

La siguiente política se adoptará para la gestión de cualquier discrepancia o no conformidad encontrada durante las pruebas de ejecución:

- Si la discrepancia no viola ningún requerimiento específico y no impone una suspensión de las actividades de prueba, la discrepancia será anotada en la hoja y la prueba se completará.
- De acuerdo con la naturaleza de la discrepancia, se llevará a cabo una investigación más a fondo para verificar cualquier posible acción y para decidir sobre la oportunidad de declarar el caso de prueba como **OK** o para programar su repetición.
- Si la discrepancia no es fácil de ser entendida y rápidamente solucionada y / o afecta a un requisito específico, un Reporte de Anomalía de Prueba (RAP) será llenado.
- El caso de prueba será entonces temporalmente declarado **FAILED** y se repetirá después de la acción de recuperación. Por el momento, si el problema no afecta a los subsiguientes casos de prueba, las actividades de prueba continuarán de acuerdo con el calendario nominal
- Si la discrepancia implica la imposibilidad de continuar las actividades de prueba, afecta a las actividades o el funcionamiento del próximo ciclo de casos de prueba, después de la emisión del Reporte de Anomalías de Prueba (RAP) para el caso de prueba **FAILED**, la operación de la prueba será suspendida temporalmente y se convocará lo más rápido posible a una reunión en la cual participen todos los actores para analizar el problema, identificar las acciones correctivas y el plan de intervención que se deberá realizar. Una vez que el problema se haya resuelto, la operación de prueba se reiniciará desde prueba fallida para seguir de acuerdo con la secuencia prevista.

El Reporte de Anomalía de Test será de la forma mostrada en la tabla 4.2.0.-a. Una declaración formal de Clausura de Anomalía (DCA) se publicará tan pronto como el problema de anomalía se haya resuelto. El formato se puede observar en la tabla 4.2.0.-b.

Reporte de Anomalía de Prueba				ID:
ID de prueba:				
Título de la anomalía:				
Nivel de criticidad:	Alta	Media	Baja	
Descripción de la anomalía:				
Acción recomendada:				
Fecha:		Firma:		

Tabla 4.2.0.-a: Plantilla de Reporte de Anomalía de Prueba.

Declaración de Clausura de Anomalía		ID:
ID de Reporte de Anomalía:		
Título de la anomalía:		
Acción realizada:		
Resolución de la anomalía:		
Fecha:	Firma:	

Tabla 4.2.0.-b: Plantilla de Declaración de Clausura de Anomalía.

4.3. Métricas de calidad del Plan de Prueba

En el reporte del plan de pruebas, se tiene que incluir: un resumen de la lista de Reportes de Anomalía de Prueba (RAPs) y Declaraciones de Clausura de Anomalía (DACs), y un conjunto de métricas obtenidas. Este resumen incluye:

- Identificador de la anomalía.

- Título de la anomalía.
- Fecha.
- Identificador de la prueba.
- Nivel de criticidad.
- Estado actual.
- Fecha de cierre.

De esta forma, la lista de RAP se construye como una tabla. Un ejemplo de esto es la tabla 4.3.0.-a.

ID de anomalía	Título de anomalía	Fecha	ID de prueba	Criticidad	Estado actual	Fecha de cierre

Tabla 4.3.0.-a: Plantilla de lista de Reporte de Anomalías.

4.4. Métricas de calidad del Plan de Prueba

Con el fin de establecer una forma de medir el grado de éxito de la ejecución del plan de pruebas y sus actividades, se introducen un conjunto de métricas. Para cada sesión de pruebas se tendrán que incluir los siguientes elementos:

- Cantidad de procedimientos de prueba ejecutados.
- Cantidad de procedimientos de prueba exitosos (esos que no producen RAPs).
- Cantidad de procedimientos de prueba completados (los completados que produjeron RAPs).
- Cantidad de procedimientos que no se pudieron completar.
- Cantidad de procedimientos de prueba con RAPs menores (es decir, cuya importancia es mínima).
- Cantidad de procedimientos de prueba con RAPs media (es decir, cuya importancia es media).
- Cantidad de procedimientos de prueba con RAPs críticos (es decir, cuya importancia es crítica).
- Versión del producto.

De acuerdo a esto, las métricas pueden ser como se muestra en la tabla 4.4.0.-a.

Proced. de prueba ejecutados	Proced. de prueba exitosos	Fecha	Proced. de prueba completados	Proced. de prueba no completados	Proced. de prueba con RAPs bajas	Proced. de prueba con RAPs media	Proced. de prueba con RAPs alta	Versión del producto

Tabla 4.4.0.-a: Plantilla de lista de Reporte de Anomalías.

BIBLIOGRAFÍA

- E. Chuvieco. *Teledetección ambiental: La observación de la tierra desde el espacio*. Ariel Ciencias. Editorial Ariel, 2010. ISBN 9788434434981.
- R. C. Gonzalez y R. E. Woods. *Digital image processing*. Prentice-Hall, 2. ed. edición, 2002. ISBN 0-201-18075-8, 0-20-118075-8.
- J.R. Jensen. *Introductory digital image processing: a remote sensing perspective*. Prentice-Hall Series in Geographic Information Science. Pearson Prentice Hall, 2005. ISBN 9780131453616.
- A. Kundu, S.K. Mitra, y P.P. Vaidyanathan. Application of two-dimensional generalized mean filtering for removal of impulse noises from images. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(3):600–609, 1984. ISSN 0096-3518.
- J.G. Liu y P. Mason. *Essential Image Processing and GIS for Remote Sensing*. Wiley, 2009. ISBN 9780470746042.
- M. Rabolli y A. Gulich. Conocimientos básicos sobre teleobservación. satélites noaa. Technical report, CONAE, 2005.
- J.A. Richards y X. Jia. *Remote Sensing Digital Image Analysis: An Introduction*. Springer, 2006. ISBN 9783540251286.
- R.A. Schowengerdt. *Remote Sensing: Models and Methods for Image Processing*. Elsevier Science, 2006. ISBN 9780080480589.
- J. H. Sossa Azuela y R. Rodriguez Morales. *Procesamiento y Análisis Digital de Imágenes*. RA-MA S.A. Editorial y Publicaciones, 2011. ISBN 9788499640778.