

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL MENDOZA
DIRECCIÓN DE POSGRADO

MAESTRÍA EN INSTRUMENTOS SATELITALES

TESIS:

“Estudio de sistemas de compresión de imágenes para
cámara satelital de alta resolución”

Tesista: Ing. Matías Leonel **MARTINI**

Director de Tesis: Dr. Sebastián **PAZOS**

Co-Directora de Tesis: Dra. Gabriela Fabiana **PUENTE**



5 de diciembre de 2017

ESTUDIO DE SISTEMAS DE COMPRESIÓN DE IMÁGENES PARA
CÁMARA SATELITAL DE ALTA RESOLUCIÓN

Tesista: Ing. Matías Leonel Martini

Presentado ante la Universidad Tecnológica Nacional, Facultad Regional
Mendoza, como parte de los requerimientos para la obtención del grado de:

MAGÍSTER EN INSTRUMENTOS SATELITALES

Director de Tesis: Dr. Sebastián Pazos

Co-Directora de Tesis: Dra. Gabriela Fabiana Puente

Miembros del Tribunal de Tesis

Dr. Oscar Humberto Bustos

Dr. Hugo Gustavo Marraco

Dr. Jorge Adrián Sánchez

5 de diciembre de 2017

A mis padres, Juan y Carmen.
A Eva, mi compañera de vida.

Agradecimientos

Un gran número de personas e instituciones hicieron posible la realización de la presente tesis de Maestría, en estas líneas trataré de retribuir a todas ellas.

En primer lugar, deseo agradecer a Sebastián Pazos, quien dirigió esta tesis, evacuó cada una de mis dudas en el proceso, corrigió el texto y sin dudas, hizo de este trabajo uno mejor.

Dentro de las instituciones, he de agradecer a la empresa INVAP por haberme aceptado para la realización de esta tesis en el marco de las actividades de Unidad de Desarrollo de la Maestría, particularmente al grupo de electro-óptica de dicha empresa que colaboró activamente aportando datos y bibliografía. Del mencionado grupo, agradezco especialmente a Ariel Drucaroff quien me transmitió una especial motivación para abordar cada uno de los objetivos y a Gabriela Puente quien co-dirigió esta tesis. De la misma empresa, también debo mencionar a Gustavo Wiman, Gabriel Platzcek y Jose Relloso, quienes colaboraron de diferentes maneras, el aporte de cada uno de ellos fue muy valioso para mi.

De la misma forma, agradezco a la Unidad de Formación Superior (UFS) de la Comisión Nacional de Actividades Espaciales (CONAE) y a la Facultad Regional Mendoza de la Universidad Tecnológica Nacional (UTN), instituciones que como socios académicos dictaron esta Maestría y me dieron la oportunidad de ser alumno de la misma. En particular, he de agradecer a Leonardo de Ferrariis, quien como director de la UFS y de la Maestría, siempre estuvo pendiente de los problemas que surgieron a lo largo del camino. A la CONAE, deseo agradecer además, por brindarme imágenes satelitales utilizadas en esta tesis, especialmente a Antonio Menescardi y Sandra Torrusio quienes me facilitaron el acceso a los productos de EROS-B.

También es importante mencionar a la UTN que me ha provisto de una beca durante dos años para la realización de esta carrera y a la subsecretaria de

posgrado de dicha institución, Alicia Roman, quien siempre estuvo a disposición en los aspectos referidos a la misma.

A mis compañeros de la Maestría, Nicolas, Carlos, Nahuel, Leandro, Emanuel, Ismael y Danilo, con los que compartí este apasionante camino, por la compañía, consejos, ayuda y amistad brindada a lo largo de estos dos años que siempre quedarán en mi memoria.

A mi estimada Facultad Regional de Villa María de la UTN, sin la formación impartida durante mi carrera de grado no habría logrado la actual meta. En ella he encontrado docentes, amigos y compañeros que me motivaron a iniciar esta Maestría y durante el camino me aconsejaron y devolvieron el aliento cuando lo había perdido.

A mis amigos, quienes fueron el principal cable a tierra y generadores de espacios de reflexión.

Finalmente, y no por eso menos importante, a Eva, a quien además de dedicar esta tesis, agradezco por su compañía y haber sabido entender mis ausencias.

Resumen

Las imágenes adquiridas por instrumentos satelitales ópticos de alta resolución, representan grandes volúmenes de datos y su compresión con pérdidas a bordo es una alternativa para mitigar las restricciones de memoria y transmisión de datos, a costa de una degradación de la calidad. El objetivo de este trabajo es presentar la selección de un sistema de compresión de imágenes para la Cámara de Alta Resolución (CAR) de la misión SARE de la CONAE teniendo en cuenta la relación de compromiso entre la tasa de compresión, la complejidad y la degradación de la calidad de imagen.

Se realizó una exploración del estado del arte en compresión de imágenes a bordo, contemplando sistemas basados en predicción, transformada discreta de coseno y transformada discreta de wavelet, que conduce a la preselección de cuatro sistemas de compresión que son evaluados.

Se presenta una revisión de las métricas de calidad de imagen que incluye las métricas clásicas de compresión de imagen y el NIIRS (*National Imagery Interpretability Rating Scale*) que es un indicador utilizado para imágenes de alta resolución. Para la estimación del NIIRS, y su degradación debido a la compresión, se desarrolló un algoritmo que utiliza la información extraída del análisis de bordes naturales presentes en imágenes satelitales. Se muestran las alternativas de procesamiento evaluadas y su desempeño.

Finalmente, se presentan los resultados de la evaluación del desempeño de los sistemas de compresión JPEG-Baselines, JPEG2000, JPEG-LS y CCSDS-IDC, y se muestra que los últimos dos presentan las mejores opciones considerando la relación de compromiso que se persigue en este trabajo.

Palabras Claves: Compresión de imágenes, calidad de imagen, sensorado remoto, alta resolución, CAR, JPEG-LS, JPEG-Baselines, JPEG2000, CCSDS-IDC, NIIRS.

Tabla de contenidos

Lista de figuras	XIII
Lista de tablas	XXIX
Lista de acrónimos	XXXIII
1. Introducción	1
1.1. Descripción del problema	1
1.2. El satélite SARE 2A y el instrumento CAR	3
1.2.1. Tasa de datos del instrumento	7
1.2.2. Tasa de compresión deseada	8
1.3. Objetivos del trabajo de tesis	9
1.4. Metodología empleada	9
1.5. Organización de la tesis	10
1.6. Consideraciones previas	11
2. Compresión de imágenes a bordo	15
2.1. Introducción a compresión de imágenes	15
2.1.1. Compresión basada en predicción	17
2.1.2. Compresión basada en DCT	17
2.1.3. Compresión basada en DWT	19
2.2. Estado actual de la compresión de imágenes en misiones espaciales	22

2.2.1.	Clasificación de acuerdo a la base teórica	23
2.2.2.	Sistemas de compresión con y sin pérdidas	27
2.2.3.	Sistemas de compresión estandarizados y no estandarizados	28
2.2.4.	Contexto nacional	29
2.2.5.	Proyección a futuro	29
2.3.	Preselección de sistemas de compresión	30
2.4.	Algoritmo JPEG-LS	30
2.5.	Algoritmo JPEG-Baselines	35
2.6.	Algoritmo JPEG2000	39
2.7.	Algoritmo CCSDS-IDC	43
3.	Calidad de imagen	49
3.1.	Métricas clásicas de degradación de la calidad de imagen	49
3.1.1.	Métricas basadas en la imagen error	50
3.1.2.	Índice de similitud estructural	52
3.2.	Métricas de calidad de imagen absolutas	55
3.2.1.	Métricas de respuesta en frecuencia de un sistema de imagen	55
3.2.2.	NIIRS	57
3.2.3.	Ecuación general de calidad de imagen	58
3.2.4.	Caracterización de la calidad de imagen de un instrumento en órbita	62
3.2.5.	Aplicación del método del borde objetivo para estimación del NIIRS	66
3.2.6.	Degradación del NIIRS por compresión	67
4.	Estimación del NIIRS de imágenes satelitales	69
4.1.	Determinación de la posición del borde	69
4.1.1.	Método 1: Ajuste de función sigmoide	70

4.1.2.	Método 2: Ajuste de función cúbica	74
4.1.3.	Método 3: Ajuste de función spline + preDN_Edge	75
4.1.4.	Método 4: Ajuste de función spline + máxima pendiente	76
4.1.5.	Método 5: Detector de bordes Canny + transformada de Hough	76
4.1.6.	Comparación y selección	77
4.2.	Reconstrucción de la ESF	78
4.2.1.	Alineación de las respuestas al borde submuestreadas de cada fila	79
4.2.2.	Corrección debido a la inclinación del borde	79
4.2.3.	Remuestreo y suavizado de la respuesta al borde	81
4.3.	Estimación de la SNR	86
4.4.	Estimación del NIIRS	88
4.5.	Algoritmo de estimación del NIIRS en imágenes satelitales	89
4.6.	Validación del algoritmo de estimación del NIIRS	93
4.7.	Algoritmo para cálculo de la degradación del NIIRS	97
5.	Evaluación de sistemas de compresión	99
5.1.	Selección de imágenes de prueba	99
5.2.	Esquema de simulación	105
5.3.	Criterios para la evaluación	107
5.4.	Evaluación de JPEG-LS	108
5.4.1.	Sobre la implementación de JPEG-LS	108
5.4.2.	Casos de prueba para JPEG-LS	108
5.4.3.	Resultados de JPEG-LS	109
5.5.	Evaluación de JPEG-Baselines	111
5.5.1.	Sobre la implementación de JPEG-Baselines	111

5.5.2.	Casos de prueba para JPEG-Baselines	113
5.5.3.	Resultados de JPEG-Baselines	114
5.6.	Evaluación de JPEG2000	117
5.6.1.	Sobre la implementación de JPEG2000	117
5.6.2.	Casos de prueba para JPEG2000	118
5.6.3.	Resultados de JPEG2000	119
5.7.	Evaluación de CCSDS-IDC	124
5.7.1.	Sobre la implementación de CCSDS-IDC	124
5.7.2.	Casos de prueba para CCSDS-IDC	125
5.7.3.	Resultados de CCSDS-IDC	126
5.8.	Análisis visual	130
5.9.	Comparación entre sistemas y selección	133
6.	Conclusiones y trabajos a futuro	139
6.1.	Conclusiones	139
6.2.	Trabajos a futuro	141
Anexos		143
A.	Cálculo del tiempo de línea	145
B.	Relevamiento de compresión de imágenes a bordo	147
C.	Metodología de simulación de un borde objetivo	153
D.	Desarrollo de (4.15)	161
E.	Histogramas de RER, H y SNR para imágenes de validación	163
F.	Estimación de NIIRS con método 2 para reconstrucción de ESF	167
G.	Resultados de evaluación de JPEG-LS	169
H.	Resultados de evaluación de JPEG-Baselines	175
I.	Resultados de evaluación de JPEG2000	185

J. Resultados de evaluación de CCSDS-IDC 199

Bibliografía **209**

Lista de figuras

1.1. Proceso completo compresión / descompresión.	2
1.2. GSD para banda PAN y banda MS.	4
1.3. Diagrama de bloques simplificado de la CAR.	5
1.4. Instrumento push broom [9].	5
1.5. Imagen digital.	12
2.1. Arquitectura típica de sistema de compresión y de descompresión.	17
2.2. Descorrelación espacial para sistemas basados en predicción. . . .	18
2.3. DWT 1D, (a) descomposición en sub-bandas y (b) reconstrucción.	20
2.4. DWT 2D.	21
2.5. Estructura de sub-bandas de la DWT 2D.	22
2.6. Distribución de los compresores de acuerdo al tipo de descorrelación espacial.	24
2.7. Distribución de los sistemas de compresión de acuerdo la pérdida de información.	27
2.8. Tasa de compresión para las misiones relevadas.	28
2.9. Distribución de los sistemas de compresión de acuerdo a la estandarización.	29
2.10. Diagrama de bloques del compresor JPEG-LS [27].	31
2.11. Cuantificación de gradientes.	32
2.12. Distribución geométrica bilateral desviada.	33

2.13. Diagrama de bloques del compresor JPEG-Baselines [26].	36
2.14. Preparación de coeficientes cuantificados para la codificación de entropía	38
2.15. Diagrama de bloques del descompresor JPEG-Baselines [26].	39
2.16. Diagrama de bloques del compresor JPEG2000.	40
2.17. Cuantificación escalar con zona muerta.	42
2.18. Procedimiento de codificación por plano de bits para datos de 6 bits de magnitud y un bit de signo.	43
2.19. Diagrama de bloques del compresor CCSDS-IDC.	44
2.20. Coeficientes de la DWT organizados en bloques para CCSDS-IDC [41].	45
2.21. Estructura de codificación de un plano de bits de un segmento [34].	47
3.1. Imágenes con igual MSE pero visualmente distintas.	52
3.2. Diagrama de flujo para el cálculo del SSIM [58].	53
3.3. Función impulso unitario correspondiente a la fuente puntual y PSF del sistema de imagen.	55
3.4. Medición de RER y H en ESF.	60
3.5. Borde objetivo ideal.	63
3.6. Borde objetivo artificial del SSC compuesto por lonas de $\rho_{low} = 3,5\%$ y $\rho_{high} = 52\%$ [74]. (a) Tendido del objetivo y (b) imagen satelital del mismo.	63
3.7. Bordos objetivos naturales.	63
3.8. Generación de la respuesta al borde.	64
3.9. Reconstrucción de la ESF a partir de respuestas al borde submuestreadas.	65
3.10. Objetivo de borde para caracterización (a) <i>cross-track</i> y (b) <i>along-track</i>	66
4.1. Borde objetivo y zonas de cálculo de <i>preDN_bright</i> y <i>preDN_Dark</i> .	71

4.2. Respuesta al borde de dos líneas y sus respectivas derivadas. . . .	71
4.3. Estimación de la posición del borde en cada fila por el método 1 (ajuste de función sigmoide) para imagen con borde inclinado 6°	73
4.4. Posición del flanco por derivación numérica, m_x , por resultado de ajuste en cada fila, P'_x , y recta de mínimos cuadrados, $y(x)$	74
4.5. Estimación de la posición del borde en cada fila por el método 2 (ajuste de función cúbica) para imagen con borde inclinado 6° . . .	75
4.6. Estimación de la posición del borde en cada fila por el método 3 (ajuste de función spline + preDN_Edge) para imagen con borde inclinado 6°	76
4.7. Estimación de la posición del borde por el método 5 (detector de bordes Canny + transformada de Hough) para imagen con borde inclinado 6°	77
4.8. Comparación de MSE en la estimación de la posición del flanco para los 5 métodos en función del ángulo de inclinación del borde. . . .	78
4.9. Alineación de respuestas al borde submuestreadas de cada fila para una imagen con un borde inclinado 6° y sin MTFC aplicada. . . .	80
4.10. Corrección de ESF por ángulo de inclinación para objetivo sin MTFC aplicada.	81
4.11. Ajuste de spline.	82
4.12. Reconstrucción de ESF por método 1 (ajuste de spline suavizada) para imagen con un borde inclinado 6° y sin MTFC aplicada. . . .	82
4.13. Reconstrucción de ESF por método 1 (ajuste de spline suavizada) para imagen con un borde inclinado 6° y con MTFC aplicada. . . .	83
4.14. Reconstrucción de ESF por método 2 (ajuste de función que modela un borde con sobrepicos) para imagen con un borde inclinado 6° y sin MTFC aplicada.	85
4.15. Reconstrucción de ESF por método 2 (ajuste de función que modela un borde con sobrepicos) para imagen con un borde inclinado 6° y con MTFC aplicada.	85

4.16. Error de estimación de (a) RER y (b) H, para imágenes sin MTFC aplicada, en función del ángulo de inclinación de borde.	86
4.17. Error de estimación de (a) RER y (b) H, para imágenes con MTFC aplicada, en función del ángulo de inclinación de borde.	86
4.18. Zonas de ρ_{low} y ρ_{high} para el cálculo de SNR en datos de respuesta al borde ordenada y corregida por inclinación.	87
4.19. Error de estimación de SNR en función del ángulo de inclinación de borde.	88
4.20. Error de estimación de NIIRS en función del ángulo de inclinación de borde.	89
4.21. Diagrama del algoritmo de estimación del NIIRS.	90
4.22. Bordes seleccionados manualmente.	91
4.23. Recortes de bordes, posición del borde y ESF reconstruida.	93
4.24. Resultados de estimación de NIIRS vs. NIIRS Real (método 1 para la reconstrucción de la ESF).	96
4.25. Diagrama del algoritmo de estimación del $\Delta NIIRS$ por compresión con pérdidas.	98
5.1. Diagrama de flujo para la evaluación de sistemas de compresión. .	106
5.2. Resultados de JPEG-LS con P_Urb. (a) CR y (b) TPP vs. <i>NEAR</i> . 110	
5.3. Resultados de JPEG-LS con P_Urb. $\Delta NIIRS$ vs. <i>NEAR</i>	110
5.4. Resultados de JPEG-LS con P_Urb. (a) SNR_C y (b) MSSIM vs. <i>NEAR</i>	111
5.5. Gráfico de $F(Quality)$	112
5.6. Comparación de tres tablas de cuantificación para JPEG-Baselines. 115	
5.7. Resultados de JPEG-Baselines con P_Urb. (a) CR y (b) TPP vs. <i>Quality</i>	116
5.8. Resultados de JPEG-Baselines con P_Urb. (a) $\Delta NIIRS_{RER}$, (b) $\Delta NIIRS_H$, (c) $\Delta NIIRS_{G/SNR}$ y (d) $\Delta NIIRS$ vs. <i>Quality</i>	116

5.9. Resultados de JPEG-Baselines con P_Urb. (a) SNR_C y (b) MSSIM vs. <i>Quality</i>	117
5.10. Comparación entre DWT entera y DWT flotante para JPEG2000.	120
5.11. CR y SNR_C para JPEG2000 en función de la cantidad de DWT 2D.	121
5.12. TPP para JPEG2000 en función de la cantidad de DWT 2D.	121
5.13. Desempeño JPEG2000 en función del tamaño de las teselas.	122
5.14. Comparación del desempeño de JPEG2000 según tamaño de tesela en función de CR.	122
5.15. Resultados de JPEG2000 con P_Urb. (a) TPP y (b) $\Delta NIIRS$ vs. <i>CR</i>	123
5.16. Resultados de JPEG2000 con P_Urb. (a) SNR_C y (b) MSSIM vs. <i>CR</i>	124
5.17. Comparación entre DWT entera y DWT flotante para CCSDS-IDC.	127
5.18. Desempeño CCSDS-IDC en función de del tamaño de subimagen.	128
5.19. Comparación del desempeño de CCSDS-IDC según tamaño de tesela en función de CR.	128
5.20. Resultados de CCSDS-IDC con P_Urb. (a) TPP y (b) $\Delta NIIRS$ vs. <i>CR</i>	129
5.21. Resultados de CCSDS-IDC con P_Urb. (a) SNR_C y (b) MSSIM vs. <i>CR</i>	129
5.22. Análisis visual de imágenes P_Urb comprimidas.	131
5.23. Análisis visual de imágenes P_Rur comprimidas.	132
5.24. Comparación de TPP vs. CR entre los cuatro sistemas de compresión.	134
5.25. Comparación de $\Delta NIIRS$ vs. CR entre los cuatro sistemas de compresión.	135
5.26. Comparación de SNR_C vs. CR entre los cuatro sistemas de compresión.	136

5.27. Comparación de MSSIM vs. CR entre los cuatro sistemas de compresión.	137
A.1. Órbita y velocidades.	145
C.1. Diagrama de bloques de simulación de borde objetivo.	154
C.2. Zonas para el cálculo de SNR real en al borde objetivo simulado. .	159
E.1. Estimación de NIIRS en imagen GE1-01-P, histograma de (a) RER, (b) H y (c) SNR.	163
E.2. Estimación de NIIRS en imagen GE1-01-M, histograma de (a) RER, (b) H y (c) SNR.	163
E.3. Estimación de NIIRS en imagen QB2-01-P, histograma de (a) RER, (b) H y (c) SNR.	164
E.4. Estimación de NIIRS en imagen QB2-01-M, histograma de (a) RER, (b) H y (c) SNR.	164
E.5. Estimación de NIIRS en imagen IK2-01-P, histograma de (a) RER, (b) H y (c) SNR.	164
E.6. Estimación de NIIRS en imagen IK2-02-P, histograma de (a) RER, (b) H y (c) SNR.	164
E.7. Estimación de NIIRS en imagen WV1-01-P, histograma de (a) RER, (b) H y (c) SNR.	165
E.8. Estimación de NIIRS en imagen WV2-01-P, histograma de (a) RER, (b) H y (c) SNR.	165
E.9. Estimación de NIIRS en imagen WV3-01-P, histograma de (a) RER, (b) H y (c) SNR.	165
E.10. Estimación de NIIRS en imagen WV3-01-M, histograma de (a) RER, (b) H y (c) SNR.	165
E.11. Estimación de NIIRS en imagen ERB-01-P, histograma de (a) RER, (b) H y (c) SNR.	166
F.1. Resultados de estimación de NIIRS vs. NIIRS Real (método 2 para la reconstrucción de la ESF).	167

G.1. TPP vs. <i>NEAR</i> para JLS_01.	169
G.2. CR vs. <i>NEAR</i> para JLS_01.	169
G.3. SNR_C vs. <i>NEAR</i> para JLS_01.	169
G.4. MSSIM vs. <i>NEAR</i> para JLS_01.	169
G.5. $\Delta NIIRS_{RER}$ vs. <i>NEAR</i> para JLS_01.	170
G.6. $\Delta NIIRS_H$ vs. <i>NEAR</i> para JLS_01.	170
G.7. $\Delta NIIRS_{SNR}$ vs. <i>NEAR</i> para JLS_01.	170
G.8. $\Delta NIIRS$ vs. <i>NEAR</i> para JLS_01.	170
G.9. TPP vs. <i>NEAR</i> para JLS_21.	170
G.10. CR vs. <i>NEAR</i> para JLS_21.	170
G.11. SNR_C vs. <i>NEAR</i> para JLS_21.	170
G.12. MSSIM vs. <i>NEAR</i> para JLS_21.	170
G.13. $\Delta NIIRS_{RER}$ vs. <i>NEAR</i> para JLS_21.	171
G.14. $\Delta NIIRS_H$ vs. <i>NEAR</i> para JLS_21.	171
G.15. $\Delta NIIRS_{SNR}$ vs. <i>NEAR</i> para JLS_21.	171
G.16. $\Delta NIIRS$ vs. <i>NEAR</i> para JLS_21.	171
G.17. TPP vs. <i>NEAR</i> para JLS_31.	171
G.18. CR vs. <i>NEAR</i> para JLS_31.	171
G.19. SNR_C vs. <i>NEAR</i> para JLS_31.	171
G.20. MSSIM vs. <i>NEAR</i> para JLS_31.	171
G.21. $\Delta NIIRS_{RER}$ vs. <i>NEAR</i> para JLS_31.	172
G.22. $\Delta NIIRS_H$ vs. <i>NEAR</i> para JLS_31.	172
G.23. $\Delta NIIRS_{SNR}$ vs. <i>NEAR</i> para JLS_31.	172
G.24. $\Delta NIIRS$ vs. <i>NEAR</i> para JLS_31.	172
G.25. TPP vs. <i>NEAR</i> para JLS_41.	172
G.26. CR vs. <i>NEAR</i> para JLS_41.	172

G.27.	SNR_C vs. $NEAR$ para JLS_41.	172
G.28.	MSSIM vs. $NEAR$ para JLS_41.	172
G.29.	$\Delta NIIRS_{RER}$ vs. $NEAR$ para JLS_41.	173
G.30.	$\Delta NIIRS_H$ vs. $NEAR$ para JLS_41.	173
G.31.	$\Delta NIIRS_{SNR}$ vs. $NEAR$ para JLS_41.	173
G.32.	$\Delta NIIRS$ vs. $NEAR$ para JLS_41.	173
H.1.	TPP vs. $Quality$ para JPG_01.	175
H.2.	CR vs. $Quality$ para JPG_01.	175
H.3.	SNR_C vs. $Quality$ para JPG_01.	175
H.4.	MSSIM vs. $Quality$ para JPG_01.	175
H.5.	$\Delta NIIRS_{RER}$ vs. $Quality$ para JPG_01.	176
H.6.	$\Delta NIIRS_H$ vs. $Quality$ para JPG_01.	176
H.7.	$\Delta NIIRS_{SNR}$ vs. $Quality$ para JPG_01.	176
H.8.	$\Delta NIIRS$ vs. $Quality$ para JPG_01.	176
H.9.	TPP vs. $Quality$ para JPG_02.	176
H.10.	CR vs. $Quality$ para JPG_02.	176
H.11.	SNR_C vs. $Quality$ para JPG_02.	176
H.12.	MSSIM vs. $Quality$ para JPG_02.	176
H.13.	$\Delta NIIRS_{RER}$ vs. $Quality$ para JPG_02.	177
H.14.	$\Delta NIIRS_H$ vs. $Quality$ para JPG_02.	177
H.15.	$\Delta NIIRS_{SNR}$ vs. $Quality$ para JPG_02.	177
H.16.	$\Delta NIIRS$ vs. $Quality$ para JPG_02.	177
H.17.	TPP vs. $Quality$ para JPG_03.	177
H.18.	CR vs. $Quality$ para JPG_03.	177
H.19.	SNR_C vs. $Quality$ para JPG_03.	177
H.20.	MSSIM vs. $Quality$ para JPG_03.	177

H.21. $\Delta NIIRS_{RER}$ vs. <i>Quality</i> para JPG_03.	178
H.22. $\Delta NIIRS_H$ vs. <i>Quality</i> para JPG_03.	178
H.23. $\Delta NIIRS_{SNR}$ vs. <i>Quality</i> para JPG_03.	178
H.24. $\Delta NIIRS$ vs. <i>Quality</i> para JPG_03.	178
H.25.TPP vs. <i>Quality</i> para JPG_21.	178
H.26.CR vs. <i>Quality</i> para JPG_21.	178
H.27. SNR_C vs. <i>Quality</i> para JPG_21.	178
H.28.MSSIM vs. <i>Quality</i> para JPG_21.	178
H.29. $\Delta NIIRS_{RER}$ vs. <i>Quality</i> para JPG_21.	179
H.30. $\Delta NIIRS_H$ vs. <i>Quality</i> para JPG_21.	179
H.31. $\Delta NIIRS_{SNR}$ vs. <i>Quality</i> para JPG_21.	179
H.32. $\Delta NIIRS$ vs. <i>Quality</i> para JPG_21.	179
H.33.TPP vs. <i>Quality</i> para JPG_22.	179
H.34.CR vs. <i>Quality</i> para JPG_22.	179
H.35. SNR_C vs. <i>Quality</i> para JPG_22.	179
H.36.MSSIM vs. <i>Quality</i> para JPG_22.	179
H.37. $\Delta NIIRS_{RER}$ vs. <i>Quality</i> para JPG_22.	180
H.38. $\Delta NIIRS_H$ vs. <i>Quality</i> para JPG_22.	180
H.39. $\Delta NIIRS_{SNR}$ vs. <i>Quality</i> para JPG_22.	180
H.40. $\Delta NIIRS$ vs. <i>Quality</i> para JPG_22.	180
H.41.TPP vs. <i>Quality</i> para JPG_23.	180
H.42.CR vs. <i>Quality</i> para JPG_23.	180
H.43. SNR_C vs. <i>Quality</i> para JPG_23.	180
H.44.MSSIM vs. <i>Quality</i> para JPG_23.	180
H.45. $\Delta NIIRS_{RER}$ vs. <i>Quality</i> para JPG_23.	181
H.46. $\Delta NIIRS_H$ vs. <i>Quality</i> para JPG_23.	181

H.47.	$\Delta NIIRS_{SNR}$ vs. <i>Quality</i> para JPG_23.	181
H.48.	$\Delta NIIRS$ vs. <i>Quality</i> para JPG_23.	181
H.49.	TPP vs. <i>Quality</i> para JPG_31.	181
H.50.	CR vs. <i>Quality</i> para JPG_31.	181
H.51.	SNR_C vs. <i>Quality</i> para JPG_31.	181
H.52.	MSSIM vs. <i>Quality</i> para JPG_31.	181
H.53.	$\Delta NIIRS_{RER}$ vs. <i>Quality</i> para JPG_31.	182
H.54.	$\Delta NIIRS_H$ vs. <i>Quality</i> para JPG_31.	182
H.55.	$\Delta NIIRS_{SNR}$ vs. <i>Quality</i> para JPG_31.	182
H.56.	$\Delta NIIRS$ vs. <i>Quality</i> para JPG_31.	182
H.57.	TPP vs. <i>Quality</i> para JPG_41.	182
H.58.	CR vs. <i>Quality</i> para JPG_41.	182
H.59.	SNR_C vs. <i>Quality</i> para JPG_41.	182
H.60.	MSSIM vs. <i>Quality</i> para JPG_41.	182
H.61.	$\Delta NIIRS_{RER}$ vs. <i>Quality</i> para JPG_41.	183
H.62.	$\Delta NIIRS_H$ vs. <i>Quality</i> para JPG_41.	183
H.63.	$\Delta NIIRS_{SNR}$ vs. <i>Quality</i> para JPG_41.	183
H.64.	$\Delta NIIRS$ vs. <i>Quality</i> para JPG_41.	183
I.1.	TPP vs. CR para JP2_01.	185
I.2.	SNR_C vs. CR para JP2_01.	185
I.3.	MSSIM vs. CR para JP2_01.	185
I.4.	$\Delta NIIRS_{RER}$ vs. CR para JP2_01.	186
I.5.	$\Delta NIIRS_H$ vs. CR para JP2_01.	186
I.6.	$\Delta NIIRS_{SNR}$ vs. CR para JP2_01.	186
I.7.	$\Delta NIIRS$ vs. CR para JP2_01.	186
I.8.	TPP vs. CR para JP2_02.	186

I.9. SNR_C vs. CR para JP2_02.	186
I.10. MSSIM vs. CR para JP2_02.	186
I.11. $\Delta NIIRS_{RER}$ vs. CR para JP2_02.	187
I.12. $\Delta NIIRS_H$ vs. CR para JP2_02.	187
I.13. $\Delta NIIRS_{SNR}$ vs. CR para JP2_02.	187
I.14. $\Delta NIIRS$ vs. CR para JP2_02.	187
I.15. TPP vs. N_{DWT} para JP2_03.	187
I.16. CR vs. N_{DWT} para JP2_03.	187
I.17. TPP vs. N_{DWT} para JP2_04.	188
I.18. SNR_C vs. N_{DWT} para JP2_04.	188
I.19. MSSIM vs. N_{DWT} para JP2_04.	188
I.20. $\Delta NIIRS$ vs. N_{DWT} para JP2_04.	188
I.21. TPP vs. M_T para JP2_05.	188
I.22. CR vs. M_T para JP2_05.	188
I.23. TPP vs. M_T para JP2_06.	189
I.24. SNR_C vs. M_T para JP2_06.	189
I.25. MSSIM vs. M_T para JP2_06.	189
I.26. $\Delta NIIRS$ vs. M_T para JP2_06.	189
I.27. TPP vs. CR para JP2_07.	189
I.28. SNR_C vs. CR para JP2_07.	189
I.29. MSSIM vs. CR para JP2_07.	190
I.30. $\Delta NIIRS_{RER}$ vs. CR para JP2_07.	190
I.31. $\Delta NIIRS_H$ vs. CR para JP2_07.	190
I.32. $\Delta NIIRS_{SNR}$ vs. CR para JP2_07.	190
I.33. $\Delta NIIRS$ vs. CR para JP2_07.	190
I.34. TPP vs. CR para JP2_08.	190

I.35. SNR_C vs. CR para JP2_08.	190
I.36. MSSIM vs. CR para JP2_08.	191
I.37. $\Delta NIIRS_{RER}$ vs. CR para JP2_08.	191
I.38. $\Delta NIIRS_H$ vs. CR para JP2_08.	191
I.39. $\Delta NIIRS_{SNR}$ vs. CR para JP2_08.	191
I.40. $\Delta NIIRS$ vs. CR para JP2_08.	191
I.41. TPP vs. CR para JP2_09.	191
I.42. SNR_C vs. CR para JP2_09.	191
I.43. MSSIM vs. CR para JP2_09.	192
I.44. $\Delta NIIRS_{RER}$ vs. CR para JP2_09.	192
I.45. $\Delta NIIRS_H$ vs. CR para JP2_09.	192
I.46. $\Delta NIIRS_{SNR}$ vs. CR para JP2_09.	192
I.47. $\Delta NIIRS$ vs. CR para JP2_09.	192
I.48. TPP vs. CR para JP2_21.	192
I.49. SNR_C vs. CR para JP2_21.	192
I.50. MSSIM vs. CR para JP2_21.	193
I.51. $\Delta NIIRS_{RER}$ vs. CR para JP2_21.	193
I.52. $\Delta NIIRS_H$ vs. CR para JP2_21.	193
I.53. $\Delta NIIRS_{SNR}$ vs. CR para JP2_21.	193
I.54. $\Delta NIIRS$ vs. CR para JP2_21.	193
I.55. TPP vs. N_{DWT} para JP2_23.	193
I.56. CR vs. N_{DWT} para JP2_23.	193
I.57. TPP vs. N_{DWT} para JP2_24.	194
I.58. SNR_C vs. N_{DWT} para JP2_24.	194
I.59. MSSIM vs. N_{DWT} para JP2_24.	194
I.60. $\Delta NIIRS$ vs. N_{DWT} para JP2_24.	194

I.61. TPP vs. M_T para JP2_25.	194
I.62. CR vs. M_T para JP2_25.	194
I.63. TPP vs. M_T para JP2_26.	195
I.64. SNR_C vs. M_T para JP2_26.	195
I.65. MSSIM vs. M_T para JP2_26.	195
I.66. $\Delta NIIRS$ vs. M_T para JP2_26.	195
I.67. TPP vs. CR para JP2_31.	195
I.68. SNR_C vs. CR para JP2_31.	195
I.69. MSSIM vs. CR para JP2_31.	196
I.70. $\Delta NIIRS_{RER}$ vs. CR para JP2_31.	196
I.71. $\Delta NIIRS_H$ vs. CR para JP2_31.	196
I.72. $\Delta NIIRS_{SNR}$ vs. CR para JP2_31.	196
I.73. $\Delta NIIRS$ vs. CR para JP2_31.	196
I.74. TPP vs. CR para JP2_41.	196
I.75. SNR_C vs. CR para JP2_41.	196
I.76. MSSIM vs. CR para JP2_41.	197
I.77. $\Delta NIIRS_{RER}$ vs. CR para JP2_41.	197
I.78. $\Delta NIIRS_H$ vs. CR para JP2_41.	197
I.79. $\Delta NIIRS_{SNR}$ vs. CR para JP2_41.	197
I.80. $\Delta NIIRS$ vs. CR para JP2_41.	197
J.1. TPP vs. CR para IDC_01.	199
J.2. SNR_C vs. CR para IDC_01.	199
J.3. MSSIM vs. CR para IDC_01.	199
J.4. $\Delta NIIRS_{RER}$ vs. CR para IDC_01.	200
J.5. $\Delta NIIRS_H$ vs. CR para IDC_01.	200
J.6. $\Delta NIIRS_{SNR}$ vs. CR para IDC_01.	200

J.7. $\Delta NIIRS$ vs. CR para IDC_01.	200
J.8. TPP vs. CR para IDC_02.	200
J.9. SNR_C vs. CR para IDC_02.	200
J.10. MSSIM vs. CR para IDC_02.	200
J.11. $\Delta NIIRS_{RER}$ vs. CR para IDC_02.	201
J.12. $\Delta NIIRS_H$ vs. CR para IDC_02.	201
J.13. $\Delta NIIRS_{SNR}$ vs. CR para IDC_02.	201
J.14. $\Delta NIIRS$ vs. CR para IDC_02.	201
J.15. TPP vs. M_{si} para IDC_03.	201
J.16. CR vs. M_{si} para IDC_03.	201
J.17. TPP vs. M_{si} para IDC_04.	202
J.18. SNR_C vs. M_{si} para IDC_04.	202
J.19. MSSIM vs. M_{si} para IDC_04.	202
J.20. $\Delta NIIRS$ vs. M_{si} para IDC_04.	202
J.21. TPP vs. CR para IDC_05.	202
J.22. SNR_C vs. CR para IDC_05.	202
J.23. MSSIM vs. CR para IDC_05.	203
J.24. $\Delta NIIRS_{RER}$ vs. CR para IDC_05.	203
J.25. $\Delta NIIRS_H$ vs. CR para IDC_05.	203
J.26. $\Delta NIIRS_{SNR}$ vs. CR para IDC_05.	203
J.27. $\Delta NIIRS$ vs. CR para IDC_05.	203
J.28. TPP vs. CR para IDC_06.	203
J.29. SNR_C vs. CR para IDC_06.	203
J.30. MSSIM vs. CR para IDC_06.	204
J.31. $\Delta NIIRS_{RER}$ vs. CR para IDC_06.	204
J.32. $\Delta NIIRS_H$ vs. CR para IDC_06.	204

J.33. $\Delta NIIRS_{SNR}$ vs. CR para IDC_06.	204
J.34. $\Delta NIIRS$ vs. CR para IDC_06.	204
J.35. TPP vs. CR para IDC_21.	204
J.36. SNR_C vs. CR para IDC_21.	204
J.37. MSSIM vs. CR para IDC_21.	205
J.38. $\Delta NIIRS_{RER}$ vs. CR para IDC_21.	205
J.39. $\Delta NIIRS_H$ vs. CR para IDC_21.	205
J.40. $\Delta NIIRS_{SNR}$ vs. CR para IDC_21.	205
J.41. $\Delta NIIRS$ vs. CR para IDC_21.	205
J.42. TPP vs. M_{si} para IDC_23.	205
J.43. CR vs. M_{si} para IDC_23.	205
J.44. TPP vs. M_{si} para IDC_24.	206
J.45. SNR_C vs. M_{si} para IDC_24.	206
J.46. MSSIM vs. M_{si} para IDC_24.	206
J.47. $\Delta NIIRS$ vs. M_{si} para IDC_24.	206
J.48. TPP vs. CR para IDC_31.	206
J.49. SNR_C vs. CR para IDC_31.	206
J.50. MSSIM vs. CR para IDC_31.	207
J.51. $\Delta NIIRS_{RER}$ vs. CR para IDC_31.	207
J.52. $\Delta NIIRS_H$ vs. CR para IDC_31.	207
J.53. $\Delta NIIRS_{SNR}$ vs. CR para IDC_31.	207
J.54. $\Delta NIIRS$ vs. CR para IDC_31.	207
J.55. TPP vs. CR para IDC_41.	207
J.56. SNR_C vs. CR para IDC_41.	207
J.57. MSSIM vs. CR para IDC_41.	208
J.58. $\Delta NIIRS_{RER}$ vs. CR para IDC_41.	208

J.59. $\Delta NIIRS_H$ vs. CR para IDC_41.	208
J.60. $\Delta NIIRS_{SNR}$ vs. CR para IDC_41.	208
J.61. $\Delta NIIRS$ vs. CR para IDC_41.	208

Lista de tablas

1.1. Características del detector.	6
1.2. Resumen del cálculo de la tasa de datos.	7
2.1. Cantidad de implementaciones de cada algoritmo.	24
2.2. Tasa de compresión para algoritmos sin pérdidas.	28
2.3. Factores de peso para cada sub-banda (aplica con DWT entera).	44
3.1. Escalas de NIIRS [63].	58
3.2. Rango de valores de validación de la GIQE 4.0 [66].	59
3.3. Coeficientes de la GIQE según la versión.	59
4.1. Error y tiempos de ejecución promedio de cada método de estimación de la posición de borde.	78
4.2. Tiempo de ejecución promedio de cada método de reconstrucción de la ESF.	86
4.3. Características de las imágenes usadas para la validación de NIIRS.	94
4.4. Resultados de validación de NIIRS (método 1 para la reconstrucción de la ESF).	95
4.5. Comparación de estimación de NIIRS de imágenes de NIIRS conocido entre método 1 y método 2 para la reconstrucción de la ESF.	96
5.1. Opciones de imágenes de prueba.	101

5.2. Información del producto y de los recortes usados como imágenes de prueba.	104
5.3. Imágenes de prueba PAN de zonas urbanas (P_Urb).	104
5.4. Imágenes de prueba PAN de zonas rurales (P_Rur).	104
5.5. Imágenes de prueba MS de zonas urbanas (M_Urb).	104
5.6. Imágenes de prueba MS de zonas rurales (M_Rur).	104
5.7. Cálculo de media de la métrica SNR_C para cada valor del parámetro de configuración variable, P_k	107
5.8. Casos de prueba para JPEG-LS.	109
5.9. Valores por defecto de $[T_1 T_2 T_3]$ para cada valor de $NEAR$ utilizado, con $MAXVAL = 1023$	109
5.10. Casos de prueba para JPEG-Baselines.	114
5.11. Casos de prueba para JPEG2000.	119
5.12. Casos de prueba para CCSDS-IDC.	126
5.13. Grados de calidad subjetiva para el análisis visual de imágenes comprimidas.	130
5.14. Cantidad de filas de la imagen que deben almacenarse en la memoria de entrada del compresor para cada algoritmo evaluado.	134
B.1. Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (1/5).	148
B.2. Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (2/5).	149
B.3. Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (3/5).	150
B.4. Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (4/5).	151
B.5. Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (5/5).	152

F.1. Resultados de validación de NIIRS (método 2 para la reconstrucción de la ESF).	167
--	-----

Lista de acrónimos

ADPCM	<i>Adaptive DPCM</i> (DPCM adaptativo)
ASIC	<i>Application-Specific Integrated Circuit</i> (circuito integrado para aplicaciones específicas)
B	<i>Blue</i> (azul)
BPE	<i>Bit-Plane Encoder</i> (codificador de plano de bits)
BTC	<i>Block Truncation Coding</i> (codificación de truncado por bloque)
CAR	Cámara de Alta Resolución
CCSDS	<i>Consultative Committee for Space Data Systems</i>
CNES	<i>Centre National d'Études Spatiales</i> (Agencia Espacial de Francia)
CONAE	Comisión Nacional de Actividades Espaciales
CORECI	<i>COmpression REcording and CIphering</i>
CR	<i>Compression Ratio</i> (tasa de compresión)
DCT	<i>Discrete Cosine Transform</i> (transformada discreta de coseno)
DE	Diseñado Específicamente
DFT	<i>Discrete Fourier Transform</i> (transformada discreta de Fourier)

DLR	<i>Deutsches zentrum für Luft-und Raumfahrt e.V.</i> (Agencia Espacial de Alemania)
DMCii	<i>Disaster Monitoring Constellation International Imaging</i>
DN	<i>Digital Number</i> (número digital)
DoE	<i>Department of Energy</i> (Departamento de Energía de Estados Unidos)
DPCM	<i>Differential Pulse Code Modulation</i> (modulación por impulsos codificados diferencial)
DWHT	<i>Discrete Walsh-Hadamard Transform</i> (transformada discreta de Walsh-Hadamard)
DWT	<i>Discrete Wavelet Transform</i> (transformada discreta de wavelet)
EBCOT	<i>Embedded Block Coding With Optimized Truncation</i>
EO	<i>Earth Observation</i> (observación terrestre)
ESA	<i>European Space Agency</i> (Agencia Espacial Europea)
ESF	<i>Edge Spread Function</i> (función de dispersión de borde)
FIR	<i>Finite Impulse Response</i> (respuesta al impulso finita)
FPA	Filtro Pasa Alto
FPB	Filtro Pasa Bajo
FPGA	<i>Field Programmable Gate Array</i> (arreglo programable en campo)
G	<i>Green</i> (verde)
GIQE	<i>General Image-Quality Equation</i> (ecuación general de calidad de imagen)
GSD	<i>Ground Sample Distance</i> (tamaño de píxel del suelo)

GSFC	<i>Goddard Space Flight Center</i> (Centro de Vuelo Espacial Goddard)
GWG	Geospatial intelligence standards Working Group
ICM	<i>Image Compression Module</i>
IDC	<i>Image Data Compression</i>
IGN	Instituto Geográfico Nacional
IJG	<i>Independent JPEG Group</i>
IR	<i>InfraRed</i> (infrarrojo)
ISO	<i>International Organization for Standardization</i> (Organización Internacional de Normalización)
ISRO	<i>Indian Space Research Organisation</i> (Agencia India de Investigación Espacial)
ITU	<i>International Telecommunication Union</i> (Unión Internacional de Telecomunicaciones)
JAXA	<i>Japan Aerospace Exploration Agency</i> (Agencia Japonesa de Exploración Aeroespacial)
JPEG	<i>Joint Photographic Experts Group</i> (Grupo Conjunto de Expertos en Fotografía)
JPL	<i>Jet Propulsion Laboratory</i> (Laboratorio de Propulsión a Reacción)
KARI	<i>Korea Aerospace Research Institute</i> (Instituto de Investigación Aeroespacial de Corea)
KLT	<i>Karhunen-Loève Transform</i> (transformada de Karhunen-Loève)
LDC	<i>Lossless Data Compression</i>
LOCO-I	<i>LOW COMplexity LOSSless COmpression for Images</i>

LSF	<i>Line Spread Function</i> (función de dispersión de línea)
LUT	<i>LookUp Table</i> (tabla de consulta)
MAD	<i>Maximum Absolute Difference</i> (máxima diferencia absoluta)
MED	<i>Median Edge Detector</i>
MHDC	<i>Multispectral and Hyperspectral Data Compression</i>
MMRS	<i>Multispectral Medium Resolution Scanner</i> (cámara multiespectral de resolución media)
MoST	<i>Ministry of Science and Technology</i> (Ministerio de Ciencia y Tecnología de China)
MRCPB	<i>Multi-Résolution par Codage de Plans Binaires</i>
MS	<i>MultiSpectral</i> (multiespectral)
MSE	<i>Mean Square Error</i> (error cuadrático medio)
MSSIM	<i>Mean Structural SIMilarity index</i> (media del índice de similitud estructural)
MTF	<i>Modulation Transfer Function</i> (función de transferencia de modulación)
MTFC	<i>MTFC Compensation</i> (compensación de la MTF)
NASA	<i>National Aeronautics and Space Administration</i> (Administración Nacional de la Aeronáutica y del Espacio de Estados Unidos)
NIIRS	<i>National Imagery Interpretability Rating Scale</i>
NIR	<i>Near InfraRed</i> (infrarrojo cercano)
NSPO	<i>National Space Program Office</i> (Agencia Espacial de Taiwán)

OTF	<i>Optical Transfer Function</i> (función de transferencia óptica)
PAN	<i>PANchromatic</i> (pancromática)
PSF	<i>Point Spread Function</i> (función de dispersión de punto)
PSNR	<i>Peak Signal to Noise Ratio</i> (relación señal a ruido de pico)
R	<i>Red</i> (rojo)
RD	Rango Dinámico
RER	<i>Relative Edge Response</i> (respuesta al borde relativa)
RLE	<i>Run Length Encoding</i>
RM	ReMuestreo
RMSE	<i>Root Mean Square Error</i> (raíz del error cuadrático medio)
SAICS	<i>Satellite Adaptive Image Compression System</i>
SNR	<i>Signal to Noise Ratio</i> (relación señal a ruido)
SPIHT	<i>Set Partitioning In Hierarchical Trees</i>
SSC	<i>Stennis Space Center</i> (Centro Espacial Stenis)
SSIM	<i>Structural SIMilarity index</i> (índice de similitud estructural)
SWIR	<i>Short Wave InfraRed</i> (infrarrojo de onda corta)
TPP	Tiempo de ejecución medio Por Píxel
URSS	Unión de Repúblicas Socialistas Soviéticas
USGS	<i>United States Geological Survey</i> (Servicio Geológico de los Estados Unidos)

Introducción

1.1. Descripción del problema

Los satélites de observación terrestre (EO, *Earth Observation*) orbitan alrededor de la tierra y llevan distintos tipos de instrumentos para adquirir datos. Un tipo de estos, son los instrumentos ópticos de alta resolución, capaces de producir imágenes del espectro visible con una resolución tal que un píxel equivale a una porción de la tierra menor o igual a $10m$, incluso, actualmente existen instrumentos con resoluciones del orden de las decenas de centímetros. Las aplicaciones de sus imágenes abarcan seguridad, defensa, gestión de emergencias naturales, agricultura de alta precisión y estudios de crecimiento y planeamiento urbano, entre otros.

Estos instrumentos generan grandes volúmenes de información que deben ser almacenados a bordo y transmitidos a la estación terrena. Por otro lado, debido a las limitaciones de masa y potencia, y al alto costo del equipamiento de uso espacial, es importante reducir al mínimo la capacidad de almacenamiento y la tasa de transmisión. En este escenario, la cantidad de imágenes que puede tomar el instrumento es limitada [1]. Una alternativa para mitigar esta restricción es la compresión de imágenes a bordo, que es el mecanismo que permite reducir la cantidad de datos necesarios para representar una imagen.

Una arquitectura típica de un sistema de adquisición de imágenes que incluye un sistema de compresión, y contempla tanto los procesos a bordo como los procesos en tierra, se esquematiza en la Fig. 1.1. Cuando el instrumento opera bajo el modo “almacenamiento y envío”, los datos son generados por el cabezal electro óptico, se comprimen, se almacenan temporalmente y durante la ventana de visibilidad de la estación terrena se transmiten. Si el instrumento opera en modo “tiempo real”, durante el período de visibilidad, el instrumento adquiere

datos, los comprime y los envía directamente al transmisor para su descarga. En la estación terrena, se reciben los datos y se descomprimen para recuperar la información y obtener los datos de ciencia [2].

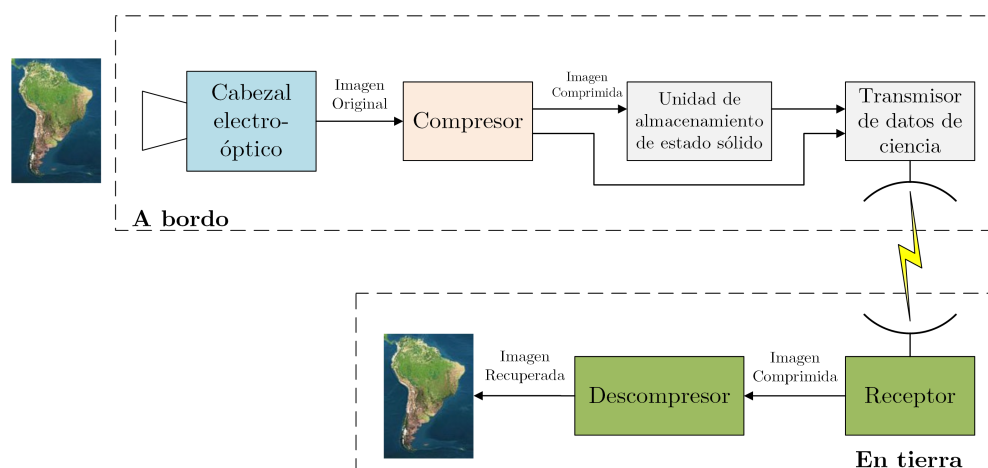


Fig. 1.1. Proceso completo compresión / descompresión.

Al aplicar la compresión a una imagen original se obtiene un grupo de datos que representa a la imagen y ocupa una menor cantidad de espacio de memoria. La relación entre el tamaño de la imagen comprimida y la imagen original es la tasa de compresión del algoritmo. Esta compresión puede ser sin pérdidas, lo cual, luego de un proceso de descompresión en tierra, permite recuperar la imagen exactamente igual a la original; o con pérdidas, que implica alguna pérdida de información y consecuentemente una degradación de la calidad de la imagen con el beneficio de lograr una mayor reducción del espacio ocupado por ella [3]. Cabe destacar que, cuando se realiza compresión con pérdidas, existe una relación inversa entre la tasa de compresión y la calidad de imagen. Por lo tanto, es esencial para un diseño óptimo del sistema, encontrar un equilibrio entre estos parámetros.

Otra variable a considerar es la complejidad del sistema de compresión, ya que el hardware disponible para el espacio no tiene grandes prestaciones de memoria, velocidad, etc. Si bien existen algoritmos con una alta tasa de compresión, son demasiado complejos para ser implementados en hardware tolerante al ambiente espacial [2].

Cuando se evalúa la pérdida de información por compresión normalmente se lo hace basándose en métricas derivadas de la diferencia entre la imagen comprimida y la imagen original [1–6]. Estas métricas clásicas, si bien son simples de calcular, no indican para qué es útil la imagen. Por otro lado, en imágenes de alta resolución se utiliza el índice *National Imagery Interpretability Rating Scale*

(NIIRS) que mide la calidad de acuerdo a qué información puede interpretarse o extraerse de la imagen. La predicción de esta métrica durante el diseño del instrumento sólo contempla la degradación producida por el detector, la óptica y otros factores propios del instrumento y el satélite. Sin embargo, es importante tener en cuenta todos los factores de degradación, incluida la compresión, para poder evaluar la utilidad final de la imagen.

En el marco del Plan Espacial Nacional de la Comisión Nacional de Actividades Espaciales (CONAE) [7], se desarrolla la misión SARE cuyo primer satélite con carga útil óptica será el SARE 2A con un instrumento denominado Cámara de Alta Resolución (CAR) desarrollado por la empresa INVAP. En esta tesis se realizó un estudio que propone la selección de un sistema de compresión por la CAR, que presenta una relación de compromiso entre la tasa de compresión, complejidad y degradación de la calidad de imagen donde ésta se evaluó tanto con métricas clásicas como por la degradación del NIIRS.

1.2. El satélite SARE 2A y el instrumento CAR

La misión SARE está formada por satélites livianos, que serán puestos en órbita por los lanzadores argentinos Tronador II/III. Van a conformar misiones satelitales de observación de la tierra y formarán parte del conjunto de satélites de arquitectura segmentada a medida que dicha tecnología se concrete. Dichos satélites se dividirán en dos grupos, SARE con carga útil óptica y SARE con carga útil de microondas. En lo que hace a la serie SARE óptico, la primera misión será una constelación de satélites con sensores de alta resolución espacial. Cada satélite llevará a bordo una cámara pancromática (PAN, *PANchromatic*) y una cámara multispectral (MS, *MultiSpectral*) con cuatro a siete bandas distribuidas en el espectro visible e infrarrojo. La misión estará dedicada principalmente a los aspectos urbanos, cartografía, transporte y seguridad, y los datos que proveerá también serán de utilidad para otras temáticas, como ser agricultura, hidrología, uso y cobertura terrestre, detección de cambios, análisis costeros y emergencias, entre otras [8].

Actualmente se encuentra en fase de diseño preliminar el primer satélite de la misión SARE con carga útil óptica, el SARE 2A, que llevará a bordo un instrumento denominado Cámara de Alta Resolución (CAR) desarrollado por la empresa INVAP. Si bien para el SARE 2A se requiere aplicar técnicas de compresión sin pérdidas, en este trabajo se propone la selección de un sistema de

compresión de imágenes con pérdidas para ser considerado en futuros satélites SARE u otros que lleven a bordo la CAR.

La CAR tiene capacidad de detección PAN y MS. El canal PAN tiene una única banda que incluye todo el espectro visible y el infrarrojo cercano. El canal MS tiene 4 bandas: rojo (R, *Red*), verde (G, *Green*), azul (B, *Blue*) e infrarrojo cercano (NIR, *Near InfraRed*). Para la banda PAN el instrumento tiene un tamaño de píxel del suelo (GSD, *Ground Sample Distance*) de 1 m y para las bandas MS de 4 m, es decir, el área cubierta por un píxel de una de las bandas MS es cubierta por 16 píxeles de la banda PAN como se ve en la Fig. 1.2.

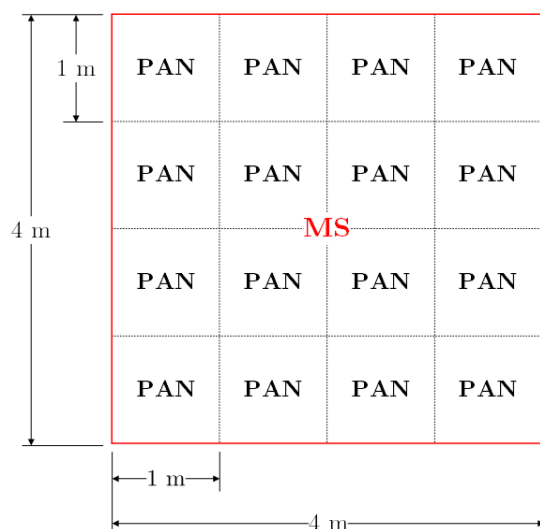


Fig. 1.2. GSD para banda PAN y banda MS.

En la Fig. 1.3 puede verse un diagrama de bloques simplificado de la CAR. Aquí pueden observarse 5 partes principales: el sistema óptico, el detector óptico, la electrónica de proximidad, el compresor de imágenes y el generador de la trama de datos de ciencia. A continuación se hace una breve descripción de cada una de las partes.

El sistema óptico es el encargado de coleccionar la radiación que llega desde el espacio objeto, en este caso la superficie terrestre, y concentrarla de manera de formar una imagen en el plano donde se encuentra el detector óptico.

El detector óptico es el dispositivo encargado de convertir la energía radiante que llega al detector en carga eléctrica que luego es sensada en la electrónica de proximidad. La CAR utiliza un detector de línea, que captura una fila de píxeles, de cada banda, cada un tiempo denominado tiempo de línea, T_l . Conforme el satélite se desplaza en su órbita, el instrumento adquiere información de líneas de la superficie terrestre consecutivas que finalmente

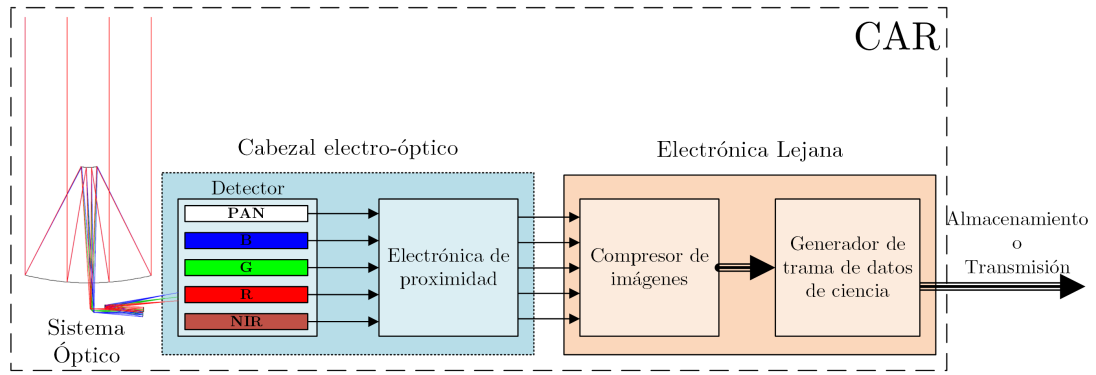


Fig. 1.3. Diagrama de bloques simplificado de la CAR.

componen una imagen. Este principio de funcionamiento, denominado *push broom*, se esquematiza en la Fig. 1.4 y es análogo a la forma en que adquiere la imagen una fotocopiadora.

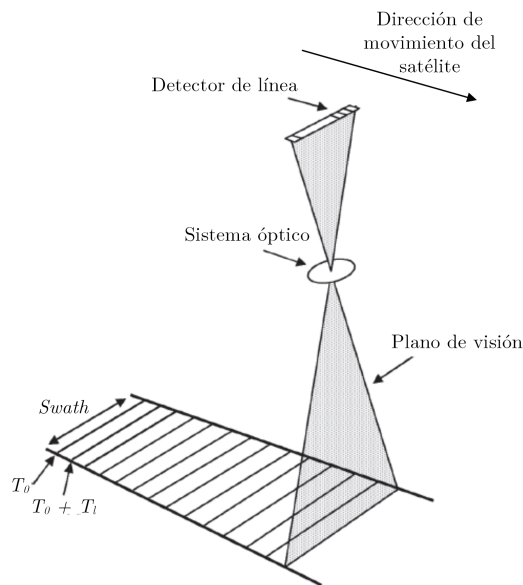


Fig. 1.4. Instrumento *push broom* [9].

El detector que utiliza la CAR cuenta con 5 líneas de píxeles, una para cada banda. La línea de que corresponde a la banda PAN tiene 12288 píxeles de ancho y las líneas que corresponden a las bandas del canal MS tienen 3072 píxeles de ancho, pero sus píxeles son 4 veces más grandes por lo tanto cubren el mismo ancho del terreno (*swath*). En Tabla 1.1 se resumen las características del detector, donde entre otros parámetros se indican las longitudes de onda que abarca cada banda.

El bloque de electrónica de proximidad contiene los circuitos necesarios para manejar el sensor, obtener las señales adquiridas por éste, acondicionarlas y

Tabla 1.1 – *Características del detector.*

Banda	Longitudes de onda [nm]	Tamaño de píxel [μm]	Ancho de la línea [píxeles]
PAN	450 - 900	7×7	12288
MS	B	28×28	3072
	G	28×28	3072
	R	28×28	3072
	NIR	28×28	3072

realizar la conversión analógica a digital. De esta manera para cada píxel se tiene un número digital (DN, *Digital Number*) que es proporcional a la energía capturada por éste. A la salida se tienen 5 canales de comunicación digital, uno para cada banda, por los cuales se envían los DN de cada píxel al compresor.

En el estado actual del diseño de la CAR, se considera cuantificar cada píxel con una cantidad de *bits* $B \leq 14$, pero no está definido exactamente con que valor. Junto con el grupo de electro-óptica de INVAP, quien tiene a cargo el diseño de la CAR, se consensuó adoptar el valor de 10bits para el desarrollo de esta tesis.

Adicionalmente, aunque no está contemplado en la arquitectura actual de la CAR, se considera que en la electrónica de proximidad los datos crudos son corregidos debido a la respuesta no uniforme de los píxeles y viñeteo. Esta técnica fue empleada por diversos instrumentos satelitales de alta resolución que realizan compresión a bordo [10–13] y sólo agrega la complejidad de un producto y una suma por cada píxel de datos generado. Se asume ésto ya que el análisis de qué ocurre al comprimir con o sin éste tipo de corrección escapa al alcance de esta tesis y a que no se dispone de imágenes de alta resolución crudas sin dicha corrección.

El compresor de imágenes recibe la imagen original y realiza la compresión para que ésta ocupe un menor espacio de memoria. Debido al principio de funcionamiento *push broom*, el compresor recibe la imagen fila a fila, para cada una de las bandas, y debe acumular la cantidad de filas necesarias en una memoria de entrada para luego aplicar a éstas el algoritmo de compresión. En el próximo capítulo se describirán sistemas que necesitan almacenar sólo una fila o varias de ellas.

El último bloque de la Fig. 1.3 recibe la imagen comprimida y se encarga de generar la trama de datos de ciencia agregando a la imagen sus correspondientes identificadores, estampa de tiempo y datos auxiliares.

Finalmente, envía los datos de ciencia a la unidad de almacenamiento de estado sólido del satélite para su almacenamiento y posterior transmisión a tierra; o directamente al transmisor para su descarga en tiempo real.

1.2.1. Tasa de datos del instrumento

La tasa de datos depende de la cantidad de píxeles en una línea de la imagen, N , el tiempo en el que se genera una nueva línea de la imagen, T_l , y la cantidad de *bits* con los que se cuantifica cada píxel, B . El tiempo de línea, depende de la altura de la órbita, H , y el GSD, entre otros factores. Para el presente caso de estudio los cálculos del tiempo de línea para cada banda pueden encontrarse en el Anexo A, mientras que los resultados se muestran en la tercera columna de la Tabla 1.2. Conociendo el tiempo de línea y la cantidad de píxeles de cada línea, la tasa de generación de píxeles de cada banda está dada por

$$Tasa\ de\ Píxeles = \frac{N}{T_l}. \quad (1.1)$$

Finalmente, dada la cantidad de *bits* con la que se cuantifica la señal de cada píxel, la tasa de datos está dada por

$$Tasa\ de\ Datos = Tasa\ de\ Píxeles \times B. \quad (1.2)$$

En la Tabla 1.2 se resumen el cálculo de la tasa de datos de cada una de las bandas y la tasa de datos total del instrumento.

Tabla 1.2 – Resumen del cálculo de la tasa de datos.

Banda	N [píxeles]	T_l [μs]	$Tasa\ de\ Píxeles$ [Mpíxel/s]	B [<i>bits</i>]	$Tasa\ de\ Datos$ [Mbps]
PAN	12288	143,16	85,8429	10	858,429
B	3072	572,58	5,3652	10	53,652
G	3072	572,58	5,3652	10	53,652
R	3072	572,58	5,3652	10	53,652
NIR	3072	572,58	5,3652	10	53,652
Total CAR	-	-	107,3036	-	1073,036

1.2.2. Tasa de compresión deseada

Actualmente no existen requerimientos de tasa de compresión para la CAR. No obstante pueden plantearse algunos criterios basados en las características actuales de la plataforma de servicio del SARE 2A considerando que el espacio de almacenamiento de a bordo es de 256Gbit y la tasa de datos de su transmisor de datos de ciencia es de 320Mbps .

Operación en modo almacenamiento y envío

A falta de conocimiento de los escenarios de operación de la CAR, se plantea un escenario hipotético donde el satélite realiza una pasada de Norte a Sur de Argentina recorriendo la máxima longitud posible que es de aproximadamente 3600 Km . El tiempo durante el cual el instrumento se encuentra adquiriendo es

$$T_{ad} = \frac{3600\text{Km}}{v_{st}} = 515,3163\text{ s}, \quad (1.3)$$

donde v_{st} es la velocidad con la que la línea de píxeles se desplaza por la superficie terrestre, calculada en el Anexo A. El volumen de datos generados en este tiempo resulta

$$D_{ad} = \text{Tasa de Datos}_{CAR} \times T_{ad} = 552,96\text{ Gbits}. \quad (1.4)$$

Si durante esta pasada se desea almacenar todo el volumen de datos en el espacio de almacenamiento de a bordo, debe reducirse el volumen de datos en un factor

$$CR = \frac{D_{ad}}{256\text{ Gbits}} = 2,16. \quad (1.5)$$

Operación en modo tiempo real

Acorde a la arquitectura de la Fig. 1.1, la segunda alternativa es transmitir los datos conforme estos son generados, sin almacenarlos temporalmente. Por lo tanto, la tasa de generación de datos debe disminuirse hasta alcanzar la tasa de datos del transmisor de datos de ciencia, lo que implica una reducción en un factor

$$CR = \frac{\text{Tasa de Datos}_{CAR}}{320\text{Mbps}} = 3,35 \quad (1.6)$$

Cabe considerar que ambos criterios son aproximaciones de primer orden, no obstante brindan una idea de cuanto debe reducir el tamaño de las imágenes el sistema de compresión. Se tomará el mayor de estos valores, que por conveniencia se redondea a $3,5$, como un valor propuesto para algunos análisis que se realizan

en el Capítulo 5. Esto último sin dejar de lado el objetivo de esta tesis de encontrar el sistema de compresión que presenta una relación de compromiso entre la tasa de compresión, la degradación de la calidad de imagen y la complejidad.

1.3. Objetivos del trabajo de tesis

Objetivo general

- Realizar un estudio sobre sistemas de compresión de imágenes que permita seleccionar un compresor para la CAR, que presente una relación de compromiso entre la tasa de compresión, la complejidad y la degradación de la calidad de la imagen.

Objetivos específicos

- Tomar conocimiento del estado del arte en compresión de imágenes a bordo de misiones satelitales y estudiar los sistemas de compresión de imágenes que sean factibles de implementar en la CAR.
- Estudiar las métricas para evaluar la degradación de la calidad de imagen.
- Desarrollar un algoritmo para la caracterización de la degradación de la calidad de imagen en términos de la degradación del NIIRS debido a la compresión.
- Evaluar el desempeño de los distintos sistemas de compresión de imágenes en términos de: tasa de compresión, complejidad y degradación de la calidad de imagen.

1.4. Metodología empleada

Para tomar conocimiento del estado del arte en compresión de imágenes a bordo se realizó una revisión bibliográfica de las bases teóricas y se relevaron los casos de la aplicación de compresión de imágenes en misiones espaciales. Con esta información se preseleccionaron cuatro sistemas de compresión potencialmente aptos para la CAR.

En segundo lugar se estudiaron las métricas de degradación de la calidad de imagen por compresión y las distintas alternativas de procesamiento para la estimación del NIIRS.

La evaluación de la degradación del NIIRS debida a la compresión con pérdidas se hizo a través de un algoritmo, desarrollado en MATLAB, que mide el

NIIRS antes y después de comprimir. El algoritmo estima el NIIRS extrayendo la información contenida en bordes entre una zona clara y una zona oscura de la imagen, como la sombra de un edificio y campos con diferentes cultivos. Durante el desarrollo del algoritmo se evaluaron distintos métodos de procesamiento. Para seleccionar el método que estima el NIIRS con menor error se estimó el NIIRS de imágenes simuladas con efectos de degradación conocidos. Finalmente el algoritmo se validó con imágenes satelitales reales de NIIRS conocido.

Dados los sistemas de compresión preseleccionados, se realizaron simulaciones para evaluar el desempeño de los mismos. Como imágenes de prueba se utilizaron imágenes satelitales de alta resolución de características similares a la CAR, del canal PAN y MS, de áreas urbanas y rurales. Para evaluar cada uno de los sistemas de compresión estos se implementaron en MATLAB utilizando códigos existentes en C/C++. El desempeño de los sistemas de compresión se evaluó en términos de tres variables: su tasa de compresión, complejidad y degradación de la calidad de imagen. La tasa de compresión se calculó por simple cómputo del cociente entre la cantidad de *bits* necesarios para la representación de la imagen antes y después de la compresión. Como indicador de la complejidad se utilizó el tiempo que le toma al algoritmo comprimir una imagen y la cantidad de filas de la misma que deben almacenarse en la memoria de entrada del compresor. La degradación de la calidad de imagen se evaluó a través de métricas basadas en el error píxel a píxel, la estructura percibida por el sistema de visión humano y la degradación del NIIRS. De manera complementaria se realizó un análisis visual para verificar la presencia de distorsiones notables. Para cada algoritmo de compresión se plantearon casos de prueba en los que se ajustaron sus parámetros de configuración para obtener un desempeño óptimo del mismo. Finalmente se seleccionó un sistema de compresión que presenta una relación de compromiso entre las tres variables.

1.5. Organización de la tesis

En el Capítulo 2 se introducen los aspectos generales sobre compresión de imágenes y tres bases teóricas aplicadas, se muestran los resultados de una revisión sobre la aplicación de compresión a bordo de misiones satelitales y se realiza la preselección de cuatro sistemas de compresión aptos para la CAR que finalmente son explicados con mayor profundidad.

En el Capítulo 3 se introducen los aspectos teóricos sobre las métricas de degradación de la calidad de imagen clásicas y las métricas de calidad de imagen absolutas junto con métodos para su cálculo.

En el Capítulo 4 se presentan y evalúan distintas alternativas de procesamiento para la estimación del NIIRS, se describe el algoritmo desarrollado para la estimación del NIIRS, se muestran los resultados de la validación del mismo y finalmente se describe el algoritmo para evaluar la degradación del NIIRS por compresión.

En el Capítulo 5 se describen las características de las imágenes de prueba, se presentan los resultados obtenidos de las simulaciones mediante las cuales se evaluó el desempeño de los cuatro sistemas de compresión preseleccionados y por último se realiza una comparación entre ellos.

Finalmente, en el Capítulo 6, se presentan las conclusiones extraídas del trabajo y se indican las posibles líneas de investigación para futuros desarrollos vinculados al tema de esta tesis.

1.6. Consideraciones previas

Dada una imagen digital de una única componente, por ejemplo banda PAN, en esta tesis se utiliza la letra M para indicar la cantidad de filas y N para la cantidad de columnas, como se ilustra en la Fig. 1.5. Matemáticamente se utiliza tanto la representación matricial,

$$\mathbf{I} = \begin{bmatrix} \mathbf{I}_{0,0} & \mathbf{I}_{0,1} & \cdots & \mathbf{I}_{0,N-1} \\ \mathbf{I}_{1,0} & \mathbf{I}_{1,1} & \cdots & \mathbf{I}_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{M-1,0} & \mathbf{I}_{M-1,1} & \cdots & \mathbf{I}_{M-1,N-1} \end{bmatrix}, \quad (1.7)$$

como la representación funcional,

$$I[x, y] = \begin{bmatrix} I[0, 0] & I[0, 1] & \cdots & I[0, N-1] \\ I[1, 0] & I[1, 1] & \cdots & I[1, N-1] \\ \vdots & \vdots & \ddots & \vdots \\ I[M-1, 0] & I[M-1, 1] & \cdots & I[M-1, N-1] \end{bmatrix}, \quad (1.8)$$

según sea conveniente de acuerdo al tema a desarrollar. Para la primera representación i indica el número de fila y j el número de columna, de manera análoga se tienen x e y para la segunda representación. La intensidad o nivel de gris de los elementos de cada representación guardan la relación de igualdad $I_{ij} = I[i, j]$. En el caso de imágenes de K componentes espectrales, donde $K > 1$, se agrega una tercera dimension, k , o variable independiente, z , que indica la banda.

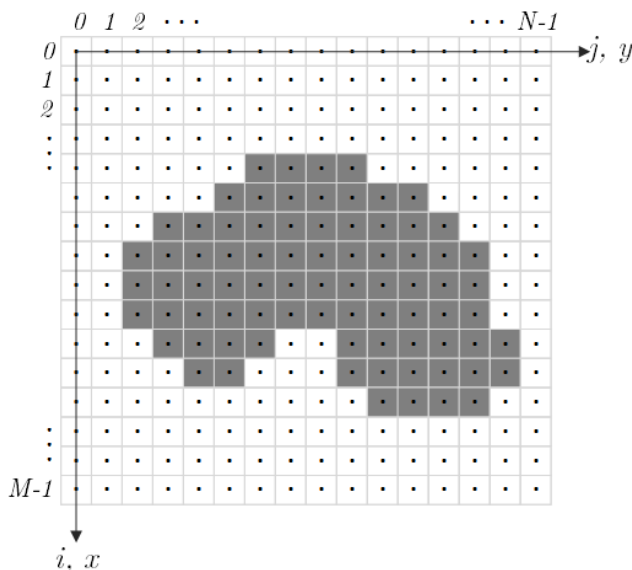


Fig. 1.5. *Imagen digital.*

La cantidad de niveles de cuantificación utilizados para representar el nivel de gris de un píxel se simboliza por la letra B y se utiliza la unidad bits por píxel, bpp , o simplemente *bits* de manera equivalente.

La tasa de compresión (CR , *Compression Ratio*) de un algoritmo se define como el cociente entre la cantidad de *bits* necesarios para representar la imagen original y el tamaño en *bits* del archivo comprimido, T_2 , es decir

$$CR = \frac{M \cdot N \cdot K \cdot B}{T_2}. \quad (1.9)$$

Alternativamente, en algunos estándares la tasa de compresión se especifica como la cantidad de bpp promedio necesarios para representar la imagen comprimida. Es decir, si una imagen de 1024×1024 píxeles se comprime en un archivo de $2Mbits$ se dice que la tasa de compresión es de $CR(bpp) = 2bpp$. El valor de

$CR(bpp)$ y su relación con CR está dado por

$$CR(bpp) = \frac{T_2}{M \cdot N \cdot K} = \frac{B}{CR}. \quad (1.10)$$

Cabe destacar que en algunos formatos de imagen utilizados típicamente por las computadoras, si la imagen es de B no múltiplo de 8, cada píxel es almacenado en una palabra digital de n -bytes, así por ejemplo, imágenes de $B = 12bpp$ ocupan $2bytes$ por cada píxel y el tamaño de la imagen es mayor. No obstante en este trabajo, B , y por lo tanto CR , siempre hacen referencia a la cantidad de *bits* útiles.

A lo largo de esta tesis, se utilizan las siguientes notaciones con sus correspondientes significados:

- $\lfloor x \rfloor$ implica redondear x al entero inmediato inferior,
- $\lceil x \rceil$ implica redondear x al entero inmediato superior,
- $round(x)$ implica redondear x al entero más cercano.

Compresión de imágenes a bordo

2.1. Introducción a compresión de imágenes

La compresión de imágenes es utilizada para reducir la cantidad de datos requeridos para representar una imagen digital previo al almacenamiento o transmisión de las mismas en aplicaciones tales como: videoconferencia, televisión, imágenes médicas, FAX, control de vehículos militares pilotados a distancia y sensado remoto [14].

En una imagen de una componente hay tres tipos de redundancias que pueden ser reducidas o eliminadas para lograr la compresión [14]. En primer lugar, la redundancia espacial (o inter-píxel) involucra elementos que están duplicados en la imagen e información de cada píxel que puede deducirse de sus vecinos. Para removerla se aplican métodos de descorrelación espacial basados en predicción y transformaciones. En segundo lugar, la redundancia de codificación (o estadística) se basa en la probabilidad de ocurrencia de los símbolos (en el caso de una imagen, un determinado nivel de gris). Para mitigarla, la idea básica es asignar palabras de código cortas a símbolos con alta probabilidad de ocurrencia y palabras de código largas a símbolos con baja probabilidad. Los dos métodos más populares son la codificación de Huffman y codificación aritmética, generalmente llamados codificación de entropía. Finalmente, el último tipo de redundancia es la información irrelevante, caracterizada por la existencia de información sensada que es ajena a la utilización prevista de la imagen o que el sistema de visión humano no percibe. Este tipo de redundancia se explota en la compresión con pérdidas por medio de una cuantificación.

Los sistemas de compresión de imágenes pueden clasificarse, de acuerdo a la pérdida de información en la imagen que se recupera luego del proceso de compresión, en tres categorías:

- **Sin pérdidas (*Lossless*):** La imagen puede ser recuperada exactamente igual a la original pero se tiene una tasa de compresión limitada por la entropía (o cantidad de información) de la imagen.
- **Con pérdidas (*Lossy*):** La imagen recuperada difiere de la original y se obtienen mayores tasas que en el caso sin pérdidas. La tasa de compresión es inversamente proporcional a la calidad de la imagen.
- **Casi sin pérdidas (*Near-Lossless*):** Es un caso particular de la compresión con pérdidas donde el error píxel a píxel entre la imagen original y la recuperada se mantiene debajo de un nivel definido. Principalmente están basados en predicción, la idea básica es incluir el decodificador en el compresor. Existen pocos algoritmos estándar y son mayormente diseñados ad hoc [5].

Otra manera de clasificar los sistemas de compresión es según cómo estos procesan los datos. Dada una imagen de M filas y N columnas, algunos procesan la información por franjas, mientras que otros lo hacen por área. Los sistemas que procesan la imagen por franja, sólo necesitan una o unas pocas líneas de la imagen con sus N columnas para procesar. Su virtud es que requieren un hardware sencillo con reducida memoria. Por otro lado existen sistemas que procesan la imagen por área, es decir necesitan almacenar la imagen de M filas por N columnas para procesar, consecuentemente requieren mayor cantidad de memoria.

Dada una imagen de entrada de tamaño fijo, los sistemas de compresión pueden clasificarse respecto a si el archivo comprimido varía su tamaño dependiendo de las características de la entrada o si su tamaño es siempre igual. El primer caso corresponde a los sistemas de CR variable, que incluye a todos los sin pérdidas y algunos con pérdidas. El segundo caso corresponde a los compresores de CR fija que contempla al resto de los sistemas de compresión con pérdidas.

En la Fig. 2.1(a) se ilustra una arquitectura típica de compresión de imágenes que consiste en tres etapas: descorrelación espacial, cuantificación (sólo en los casos con pérdidas) y codificación de entropía. Análogamente, el descompresor ilustrado en Fig. 2.1(b), presenta las operaciones opuestas: decodificación de entropía, reconstrucción de datos cuantificados (sólo en los casos con pérdidas) y reconstrucción espacial. Dependiendo de la técnica utilizada para eliminar la correlación espacial, los sistemas pueden ser clasificados de acuerdo a

su base teórica como: basados en predicción, basados en transformada discreta de coseno (DCT, *Discrete Cosine Transform*) y basados en transformada discreta de wavelet (DWT, *Discrete Wavelet Transform*) [3]. A continuación se describen brevemente estas tres bases teóricas aplicadas a compresión de imágenes.

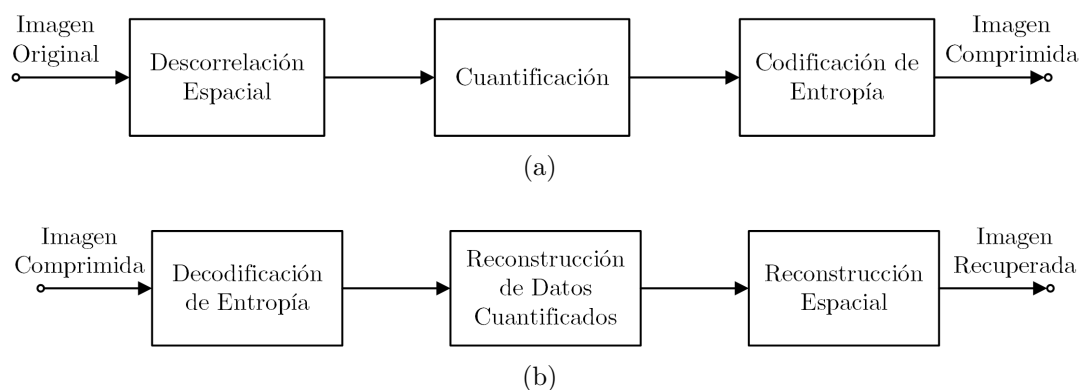


Fig. 2.1. Arquitectura típica de (a) sistema de compresión y (b) sistema de descompresión.

2.1.1. Compresión basada en predicción

Los sistemas de compresión basados en predicción son simples en términos computacionales permitiendo compresión sin, con y casi sin pérdidas. Eliminan la redundancia de los píxeles cercanos prediciendo el valor de un píxel en base al de sus vecinos y transmitiendo el error de predicción, como se indica en la Fig. 2.2(a). El predictor utilizado depende del compresor. Al realizar la predicción se pasa de una representación de n -bits a una representación de $(n + 1)$ -bits con signo, no obstante ahora los datos tienen una distribución geométrica bilateral con menor entropía que puede ser codificada más eficientemente [14]. Para ilustrar esto, en la Fig. 2.2(b) se muestra un ejemplo donde se utilizó como predicción del píxel x el valor del píxel $x - 1$ (predictor de retardo unitario), la entropía en la imagen original es $H_1 = 7,44 \text{ bits/píxel}$ mientras que en la imagen del error de predicción es $H_2 = 1 \text{ bit/píxel}$.

2.1.2. Compresión basada en DCT

Dada la naturaleza bidimensional de una imagen, las transformadas que aquí se mencionan son 2D y además son separables, lo que implica que primero puede hacerse la transformada a lo largo de las filas y luego a lo largo de las columnas. Entonces, si \mathbf{I} es una imagen de $n \times n$ píxeles su transformada \mathbf{T}

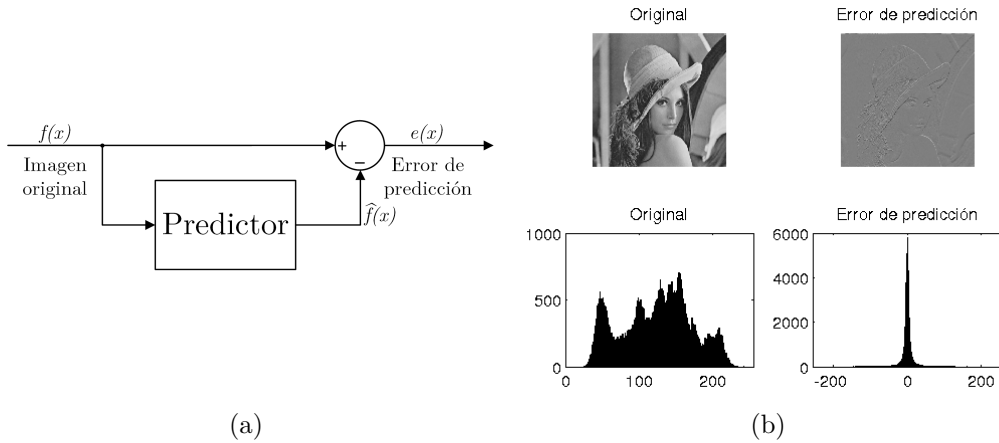


Fig. 2.2. Descorrelación espacial para sistemas basados en predicción. (a) esquema del bloque de decorrelación espacial y (b) imagen original e imagen del error de predicción con sus histogramas.

puede expresarse por el producto de matrices

$$\mathbf{T} = \mathbf{A} \mathbf{I} \mathbf{A}^t \quad (2.1)$$

donde \mathbf{A} es la matriz o base de la transformación, de tamaño $n \times n$, y \mathbf{A}^t su respectiva transpuesta. Además, considerando que las transformadas tratadas son ortonormales, la inversa está dada por

$$\mathbf{I} = \mathbf{A}^t \mathbf{T} \mathbf{A}. \quad (2.2)$$

En la bibliografía se analiza el desempeño de distintas transformaciones para la compresión de datos [14–16], entre ellas se encuentran la transformada de Karhunen-Loève (KLT, *Karhunen-Loève Transform*) que tiene la mayor capacidad de compactación, es decir, decorrelacionar la información y retenerla en unos pocos coeficientes. El problema de ésta, es que las bases de la transformación no son siempre las mismas, sino que depende de la imagen a transformar y deben calcularse para cada caso, lo que implica un alto costo computacional y que la matriz debe transmitirse junto con la imagen comprimida. Transformaciones tales como transformada discreta de Fourier (DFT, *Discrete Fourier Transform*), transformada discreta de Walsh-Hadamard (DWHT, *Discrete Walsh-Hadamard Transform*) y transformada discreta de coseno (DCT, *Discrete Cosine Transform*); son de matriz de transformación conocida e independiente de la entrada. Entre ellas la DCT es la que tiene la

mejor capacidad de compactar información y es muy cercana a la de la KLT, por lo tanto a sido utilizada en diversos sistemas de compresión. Su matriz de transformación está dada por

$$\mathbf{A}_{ij} = \begin{cases} \sqrt{\frac{1}{N}} \cdot \cos\left[\frac{(2j+1)i\pi}{2n}\right] & i = 0, \\ \sqrt{\frac{2}{N}} \cdot \cos\left[\frac{(2j+1)i\pi}{2n}\right] & i > 0. \end{cases} \quad (2.3)$$

Como puede verse la DCT involucra una transformación donde los elementos de la matriz de transformación son números irracionales, esto significa que los coeficientes de la transformada no pueden ser representados exactamente usando cualquier número finito de bits, por lo tanto la DCT es inherentemente una operación con pérdidas [15].

Considerando una imagen de M filas y N columnas, los sistemas de compresión basados en DCT en primer lugar dividen a ésta en subimágenes de $n \times n$ píxeles no solapadas, y en caso de que M o N no sean múltiplos enteros de n , generalmente se hace un relleno extendiendo simétricamente. Luego a cada subimagen se aplica la DCT 2D, donde cada coeficiente resultante es la amplitud de la correspondiente componente de frecuencia 2D, el coeficiente \mathbf{T}_{00} se denomina coeficiente DC y el resto coeficientes AC.

Por último estos coeficientes son cuantificados y codificados. Para realizar la cuantificación, se utiliza una tabla de cuantificación, dada por un matriz de la forma

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{0,0} & \mathbf{Q}_{0,1} & \cdots & \mathbf{Q}_{0,n-1} \\ \mathbf{Q}_{1,0} & \mathbf{Q}_{1,1} & \cdots & \mathbf{Q}_{1,n-1} \\ \dots & \vdots & \ddots & \vdots \\ \mathbf{Q}_{n-1,0} & \mathbf{Q}_{n-1,1} & \cdots & \mathbf{Q}_{n-1,n-1} \end{bmatrix}, \quad (2.4)$$

y los coeficientes cuantificados resultan de la operación

$$\mathbf{T}_{\mathbf{Q}ij} = \text{round}\left(\frac{\mathbf{T}_{ij}}{\mathbf{Q}_{ij}}\right). \quad (2.5)$$

2.1.3. Compresión basada en DWT

La transformada discreta de wavelet (DWT, *Discrete Wavelet Transform*) es una transformación que descompone una imagen en coeficientes que muestran la aproximación y detalles de la imagen en menores niveles de resolución y

también descorrelaciona la información de los píxeles de manera que estos pueden ser codificados de una forma más eficiente para lograr compresión. Además, si la transformación es capaz de concentrar la información más importante en un pequeño número de coeficientes, los restantes pueden ser cuantificados groseramente o truncados a cero con una pequeña distorsión en la imagen [14].

La DWT permite una descomposición de la señal en sub-bandas compuestas por los coeficientes resultantes de la transformada. Así la DWT 1D de una señal $x[n]$ puede calcularse mediante el esquema de banco de filtros mostrado en la Fig. 2.3(a). La señal de entrada $x[n]$ pasa a través de un Filtro Pasa Bajo (FPB) y un Filtro Pasa Alto (FPA), dominados filtros de descomposición o análisis, con respuestas al impulso $h_{dL}[n]$ y $h_{dH}[n]$ respectivamente. Las salidas de los filtros, $x'_L[n]$ y $x'_H[n]$, son submuestreadas por un factor de dos y resultan las sub-bandas de aproximación y detalle, es decir $a_L[n] = x'_L[2n]$ y $a_H[n] = x'_H[2n]$. Luego, es posible recuperar la señal de entrada original de acuerdo al esquema de la Fig. 2.3(b). Las sub-bandas son sobremuestreadas, se aplican los filtros de reconstrucción (o síntesis), $h_{rL}[n]$ y $h_{rH}[n]$, y se suman los resultados.

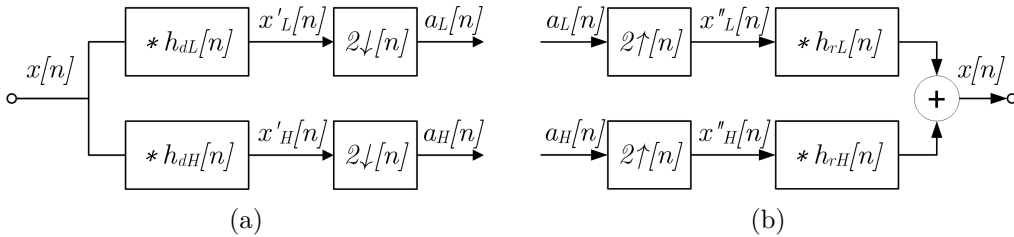


Fig. 2.3. DWT 1D, (a) descomposición en sub-bandas y (b) reconstrucción.

Los filtros $h_{dL}[n]$ y $h_{dH}[n]$ son diseñados de tal manera que luego del proceso de submuestreo la señal pueda ser recuperada de las muestras restantes y son denominados filtros de reconstrucción perfecta. Los filtros usados en sistemas de compresión son de soporte compacto, es decir son filtros de respuesta al impulso finita (FIR, *Finite Impulse Response*), tienen longitud impar y son simétricos alrededor del origen. Por ejemplo el estándar JPEG2000 especifica el par de filtros de descomposición

$$h_{dL}[n] = \begin{cases} 0,75 & \text{si } n = 0 \\ 0,25 & \text{si } n = \pm 1 \\ -0,125 & \text{si } n = \pm 2 \end{cases} \quad h_{dH}[n] = \begin{cases} 0,5 & \text{si } n = 0 \\ -0,25 & \text{si } n = \pm 1 \end{cases} \quad (2.6)$$

denominado “biortogonal 5/3” (5 coeficientes en el FPB y 3 en el FPA) [17].

Debido a la simetría de los filtros es necesario realizar un relleno en los bordes de la señal, de $\frac{L-1}{2}$ muestras, donde L es la longitud del filtro. Para evitar la discontinuidad de borde en los sistemas de compresión se usa la expansión simétrica que consiste en replicar en orden inverso las $\frac{L-1}{2}$ muestras cercanas al borde [14].

Dada la naturaleza bidimensional de las imágenes se aplica la DWT 2D que consiste en aplicar primero la DWT 1D a las filas y luego a las columnas. En la Fig. 2.4 se muestra como una imagen puede ser descompuesta en sub-bandas. Dada una imagen de $M \times N$ píxeles, se filtra y submuestra cada fila para obtener dos imágenes intermedias de $M \times N/2$, una horizontalmente filtrada por un FPB y otra horizontalmente filtrada por un FPA. Luego se aplica el mismo filtrado y submuestreo a las columnas para obtener 4 subimágenes de $M/2 \times N/2$. De las cuatro subimágenes, la que fue filtrada por un FPB en sus filas y columnas se denomina sub-banda LL (o aproximación); la que fue obtenida por aplicación de un FPB a las filas y FPA a las columnas se denomina sub-banda LH (o detalle vertical); la que fue obtenida por aplicación de un FPA a las filas y FPB a las columnas se denomina sub-banda HL (o detalle horizontal); y la imagen obtenida por un filtrado FPA a las filas y columnas se denomina HH (o detalle diagonal). Esta descomposición se representa como se muestra en la figura Fig. 2.5(a). A cada una una de las sub-bandas obtenidas puede aplicarse nuevamente la DWT 2D para obtener 4 sub bandas más hasta alcanzar la estructura deseada.

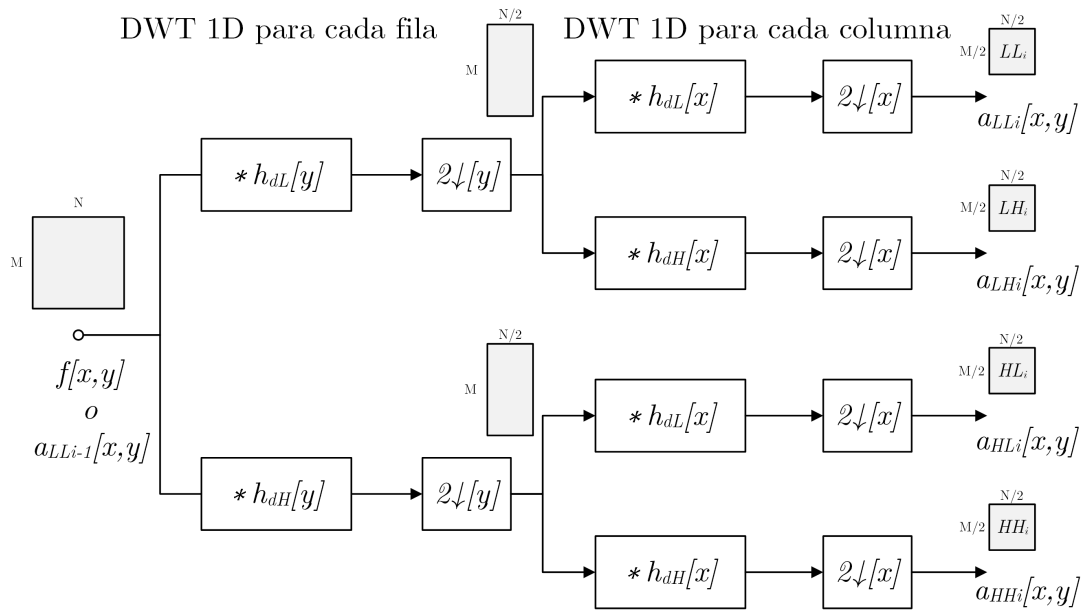


Fig. 2.4. DWT 2D.

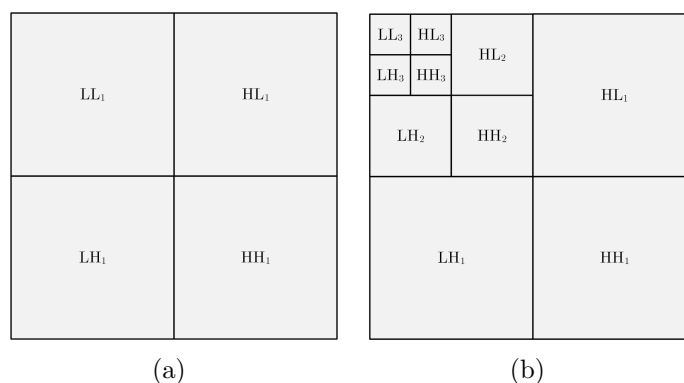


Fig. 2.5. Estructura de sub-bandas de la DWT 2D. (a) Descomposición de 1 nivel y (b) descomposición de 3 niveles.

La estructura comúnmente utilizada en compresión de imágenes, consiste en descomponer sucesivamente la banda LL en sub-bandas, así se obtienen 4 sub-bandas de cada nivel. En la Fig. 2.5(b) se ve una descomposición wavelet de 3 niveles donde: LH_1 , HL_1 y HH_1 representan los detalles de nivel 1 (resultantes de la primera descomposición); LH_2 , HL_2 y HH_2 representan los detalles de nivel 2 (resultantes de descomponer la sub-banda LL_1); LH_3 , HL_3 y HH_3 representan los detalles de nivel 3 (resultantes de descomponer la sub-banda LL_2) y LL_3 representa la aproximación de nivel 3.

Los sistemas de compresión habitualmente utilizan una implementación alternativa para obtener la DWT, denominada esquema de *lifting*. Esta implementación permite una significativa reducción en la memoria necesaria y la complejidad computacional, con resultados idénticos a los obtenidos por el banco de filtros. Los detalles acerca de esta implementación pueden verse en [18].

Luego de la descomposición wavelet, en la cantidad de niveles correspondientes, cada sistema aplica técnicas diferentes para la cuantificación y codificación de los coeficientes.

2.2. Estado actual de la compresión de imágenes en misiones espaciales

Una exhaustiva revisión bibliográfica fue realizada para tomar conocimiento del estado del arte de los sistemas de compresión de imágenes a bordo. Como resultado se identificaron más de 70 misiones que utilizaron compresión en instrumentos ópticos que pueden verse en las Tablas B.1 a B.5 del Anexo B. En éstas tablas, para cada misión relevada, se indica: el sistema

de compresión utilizado, su base teórica, si es un sistemas con o sin pérdidas, si es un sistema estandarizado o no y la tasa de compresión (en los casos que se encontró publicada). Además, se muestran otras características de la misión como propietario, año de lanzamiento y su uso.

En las siguientes secciones se presenta un resumen de lo relevado que muestra el estado del arte de la compresión de imágenes a bordo haciendo una clasificación de acuerdo a: su base teórica, sistemas con pérdidas y sin pérdidas, y sistemas estandarizados y no estandarizados. Finalmente se comenta sobre la aplicación de sistemas de compresión a bordo en el contexto nacional y las proyecciones a futuro de la temática.

2.2.1. Clasificación de acuerdo a la base teórica

La Fig. 2.6(a) muestra que, acorde a lo relevado, los sistemas de compresión más utilizados son los basados en predicción seguidos por los basados en transformación de coseno y wavelet respectivamente. Por último, existe una pequeña cantidad de misiones en las que se utilizó codificación de truncado por bloque (BTC, *Block Truncation Coding*) [19] o técnicas de diccionario que son contabilizadas bajo la etiqueta “otros”.

En la Fig. 2.6(b) puede verse que las primeras técnicas aplicadas fueron las basadas en predicción, por su sencillez, y DCT por su capacidad de lograr mayores tasas de compresión. Estas técnicas fueron prácticamente las únicas durante las décadas del 80 y 90. La aplicación de técnicas basadas en DWT apareció luego del año 2000 y se incrementó con el pasar de los años debido a su excelente desempeño a altas CR. La principal razón para la disminución del uso de las técnicas basadas en DCT es el artefacto de compresión introducido por su método de procesamiento en bloques. Las técnicas basadas en predicción se mantienen populares en tiempos actuales debido a que es el método más efectivo de lograr compresión sin pérdidas [3].

Para las tres bases teóricas más importantes, en la Tabla 2.1 puede verse cuántas implementaciones se relevaron para cada algoritmo. En las secciones siguientes se presenta una breve reseña histórica de la aplicación de estos algoritmos en misiones espaciales.

2.2.1.1. Basados en predicción

La primera aplicación de compresión de imágenes a bordo de satélites fue en el SPOT 1 (lanzado en 1986) de la Agencia Espacial de Francia (CNES,

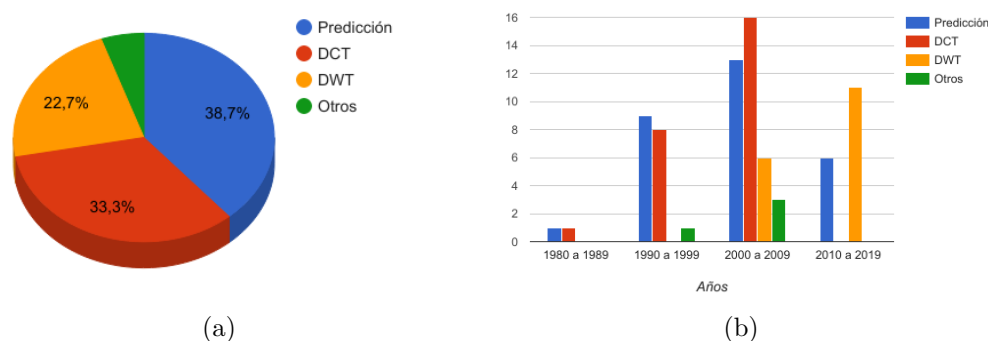


Fig. 2.6. Distribución de los compresores de acuerdo al tipo de decorrelación espacial. (a) Distribución general y (b) cantidad por década.

Tabla 2.1 – Cantidad de implementaciones de cada algoritmo.

Predicción		DCT		DWT	
DPCM	5	DCT-DE ^a	2	DWT-DE ^a	2
ADPCM	3	ICM	5	JPEG2000	6
LOCO-I	2	JPEG-Baselines	18	CCSDS-IDC	3
JPEG-LS	3			MRCPB-CORECI	4
SAICS	1			ICER	2
JPEG <i>Lossless</i>	3				
CCSDS-LDC	12				
CCSDS-MHDC	0				

^a Diseñado Específicamente (DE).

Centre National d'Études Spatiales) que utilizó un compresor de modulación por impulsos codificados diferencial (DPCM, *Differential Pulse Code Modulation*) con pérdidas y se continuó con éste hasta el SPOT 4 (1998). Fue seleccionado por su simplicidad, acorde al hardware disponible en la época. Un píxel de cada tres, se considera como un píxel de referencia y se transmite en 8 bits, mientras que la predicción de los otros dos píxeles se calcula como el promedio de los píxeles de referencia adyacentes. Finalmente el error se cuantifica no-uniformemente y se codifica con 5 bits, por lo tanto opera con una tasa de compresión fija de 1,33 [2].

El problema de los sistemas DPCM es que si el rango dinámico del cuantificador es bajo se produce un artefacto de compresión (*ringing artifact*) en zonas de alto contraste, como bordes, y si el rango dinámico es alto se degradan las zona de bajo contraste. Por lo tanto, se necesitan técnicas donde el rango dinámico del cuantificador se ajuste de acuerdo al contexto, esto conduce a los sistemas DPCM adaptativo (ADPCM, *Adaptive DPCM*). Eastman Kodak desarrolló un circuito integrado para aplicaciones específicas (ASIC, *Application-Specific Integrated Circuit*) que comprime con un sistema ADPCM y

fue utilizado por los satélites con instrumentos de alta resolución: Ikonos (1999), QuickBird (2001) y WorldView-1 (2007) [11]. Otras técnicas DPCM y ADPCM usadas en satélites pueden verse en [20].

En la década del 90, el *Consultative Committee for Space Data Systems* (CCSDS), integrado por las principales agencias espaciales, publicó una recomendación para un algoritmo de compresión de datos sin pérdidas, denominado *CCSDS-Lossless Data Compression* (CCSDS-LDC). Si bien no logra grandes tasa de compresión, es de baja complejidad. Está formado por dos bloques principales: un preprocesador y un codificador de entropía que aplica el algoritmo Rice. El preprocesador está formado un predictor de retardo unitario y luego un bloque que mapea el error de predicción a enteros no negativos [21]. Este sistema se aplicó en más de 10 misiones, entre las que se encuentran LANDSAT-8 (2013) [22] y EO-1 [23]. Para este sistema de compresión se desarrolló el ASIC USES(JC06) [24] y el *IP core* USES-32C para arreglo programable en campo (FPGA, *Field Programmable Gate Array*) [25].

El estándar de compresión de imágenes publicado por primera vez en 1991 por el Grupo Conjunto de Expertos en Fotografía (JPEG, *Joint Photographic Experts Group*) incluye dos modos: con pérdidas (denominado JPEG-Baselines) basado en DCT y sin pérdidas (denominado JPEG *Lossless*) basado en predicción [26]. El modo sin pérdidas está compuesto por un predictor seleccionable y codificación de Huffman o aritmética.

JPEG *Lossless* es ineficiente y existen pocas implementaciones, como resultado el JPEG decidió desarrollar un nuevo estándar denominado JPEG-LS [27] basado en el algoritmo *LOW COMplexity LOSSless COMpression for Images* (LOCO-I) [28], ambos aplicados en misiones espaciales [29, 30]. JPEG-LS es un sistema de baja complejidad que permite compresión de imágenes sin pérdidas y casi sin pérdidas, en la Sección 2.4 se profundiza sobre este algoritmo. Para el satélite australiano FedSat (2001) se desarrolló una técnica de compresión con pérdidas denominada *Satellite Adaptive Image Compression System* (SAICS) basada en JPEG-LS con algunas modificaciones [31].

2.2.1.2. Basados en DCT

Dentro de los algoritmos estandarizados basados en DCT, se encuentra el JPEG-Baselines que fue aplicado en más de 10 misiones entre 1998 y 2009. En primer lugar, el algoritmo realiza la DCT y cuantifica los coeficientes, como se explicó en la Sección 2.1.2. Luego éstos se escanean en zigzag y se codifican

con códigos de longitud variable (códigos de Huffman) de acuerdo a una tabla predefinida. En la Sección 2.5 se explica con mayor detalle dicho algoritmo.

El comienzo de la aplicación de algoritmos de compresión con pérdidas basados en DCT lo dio la URSS junto a Francia en la sonda interplanetaria PHOBOS (1988) con un algoritmo específicamente diseñado. Éste divide la imagen en bloques de 16×16 píxeles, les realiza la DCT, luego los coeficientes son cuantificados uniformemente y se conservan sólo los n de mayor valor absoluto, donde n depende de la CR seleccionada. Por último los coeficientes se codifican con una palabra de longitud fija de 4 bits y la posición de los no nulos se codifica usando una representación de árbol cuádruple [32].

Para el SPOT 5 (2002) se diseñó un sistema de compresión similar al JPEG-Baselines que procesa franjas de 8 filas por 12000 columnas. Se modificó la tabla de cuantificación y se agregó un lazo de control que mantiene la CR fija a 2,81 para todas las franjas [4]. Otro caso de algoritmos basados en DCT similares al JPEG-Baselines es el implementado en el *Image Compression Module* (ICM) que es un módulo calificado para el espacio que utiliza un ASIC tolerante a radiación y fue utilizado en varios satélites artificiales de la tierra y de otros cuerpos celestes [1].

2.2.1.3. Basados en DWT

La aplicación de técnicas de compresión de imágenes basadas en DWT a bordo de satélites tuvo lugar luego del año 2000. La técnica más utilizada es el estándar JPEG2000. Ésta consiste en una descomposición DWT en L niveles definidos por el usuario, cuantificación uniforme con zona muerta y codificación basada en códigos aritméticos. El estándar soporta compresión con pérdidas, sin pérdidas y operación tanto por área como por franja. En la Sección 2.6 se profundiza sobre este algoritmo.

Luego de la publicación del estándar CCSDS-LDC, el grupo de trabajo de compresión de datos continuó trabajando en un estándar de compresión de imágenes apto para el espacio. El resultado fue la recomendación CCSDS-*Image Data Compression* (CCSDS-IDC) publicada en 2005 [33,34]. Esta recomendación puede ser usada tanto para compresión sin pérdidas como con pérdidas y operar por área o por franjas. Se asemeja bastante al estándar JPEG2000 aunque es más sencillo computacionalmente y presenta algunas ventajas para su uso en el espacio [3]. Este algoritmo realiza una descomposición wavelet en 3 niveles y la cuantificación se realiza junto con la codificación, la cual aplica codificación por

plano de bits. Este algoritmo se utilizó en los satélites de observación terrestre Prova-V (2013) [35,36], Formosat-5 (2017) [37] y existe una implementación más, a bordo del satélite EnMAP, que se espera sea lanzado en los próximos años [38].

Aparte de las dos técnicas estandarizadas antes mencionadas se utilizaron sistemas de compresión específicamente diseñados. Para los satélites Pleiades-HR la CNES desarrolló una técnica similar a JPEG2000 que toma bloques de 16 filas por 1496 columnas de píxeles y les realiza la descomposición DWT en dos niveles [2, 39]. Otra técnica similar al JPEG2000 es el algoritmo MRCPB implementado en el módulo CORECI de Airbus Defense & Space que además incluye memoria masiva y encriptación [40] y se utilizó en los satélites SPOT 6 (2012), SPOT 7 (2013) y KazEOsat-1 (2014). Por último puede mencionarse el algoritmo ICER, desarrollado para los vehículos de exploración marciana *Opportunity* y *Spirit* (2003), que realiza la descomposición DWT en 4 niveles y permite compresión con pérdidas y sin pérdidas, no obstante, dichos vehículos utilizan este algoritmo sólo para compresión con pérdidas y LOCO-I para compresión sin pérdidas [29].

2.2.2. Sistemas de compresión con y sin pérdidas

Desde otro punto de vista, como se muestra en la Fig. 2.7, más de la mitad de las implementaciones relevadas son de sistemas de compresión con pérdidas, debido a la amplia utilización de los sistemas predictivos con pérdidas y basados en DCT. El resto de los casos relevados presentan una distribución similar entre compresión sin pérdidas y sistemas donde existe la opción de comprimir tanto sin pérdidas como con pérdidas.

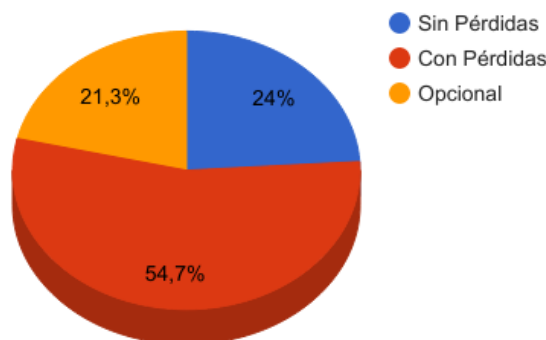


Fig. 2.7. Distribución de los sistemas de compresión de acuerdo a la pérdida de información.

En la Fig. 2.8 se muestra como se distribuyen los valores de tasas de compresión que se utilizaron en aquellas misiones en las que se encontró este dato publicado. Claramente puede observarse que los sistemas de compresión sin

pérdidas presentan tasas menores en comparación con los sistemas con pérdidas. Además puede verse que en ningún caso se excede $CR = 10$ y que la mayoría de los casos se encuentra por debajo de $CR = 5$.

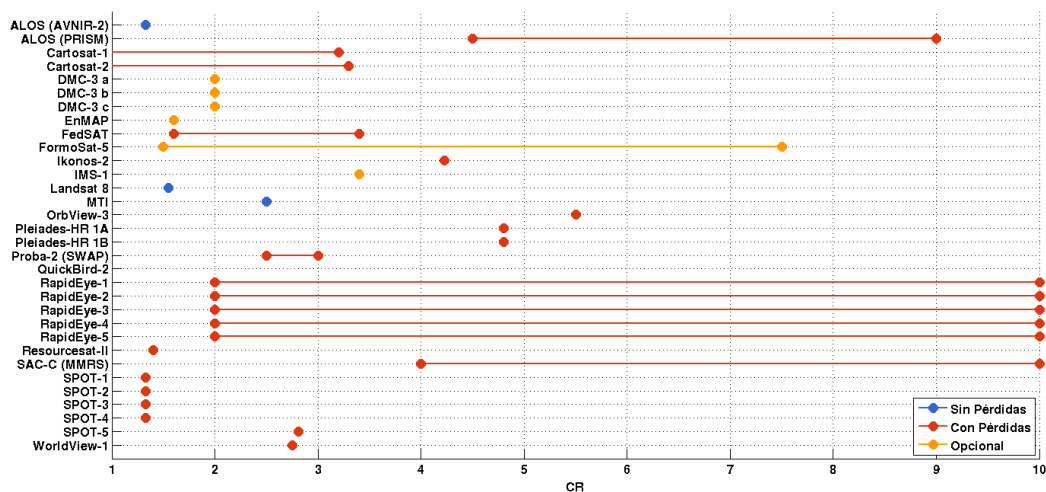


Fig. 2.8. Tasa de compresión para las misiones relevadas.

Considerando el estudio hecho por el CCSDS [41] con más de 30 imágenes espaciales (de 8, 10, 12 y 16 *bpp*), que se resume en la Tabla 2.2, puede verse que los sistemas de compresión sin pérdidas, tanto los basados en franja como los basados en área, obtienen tasas de compresión de aproximadamente 2. Además, el sistema de compresión JPEG-LS es considerado el estado del arte en compresión sin pérdidas ya que generalmente logra la mejor tasa de compresión. Sólo es superado por ICER pero a costa de un gran incremento en complejidad.

Tabla 2.2 – Tasa de compresión para algoritmos sin pérdidas.

B [<i>bpp</i>]	operación por franja				operación por área			
	CCSDS IDC	CCSDS IDC	JPEG LS	JPEG 2000	CCSDS IDC	JPEG 2000	SPIHT	ICER
8	2,09	1,96	2,26	2,18	2,09	2,22	2,23	2,27
10	2,13	1,87	2,24	2,16	2,13	2,18	2,18	2,24
12	2,1	1,91	2,15	2,13	2,1	2,14	2,31	2,21
16	1,45	1,39	1,45	1,42	1,45	1,42	–	1,4

2.2.3. Sistemas de compresión estandarizados y no estandarizados

En las misiones espaciales se utilizaron tanto sistemas de compresión diseñados específicamente, no estandarizados, como sistemas estandarizados. Como se ilustra en la Fig. 2.9, el último grupo presenta la mayoría entre

los sistemas relevados. Entre los sistemas estandarizados pueden mencionarse los desarrollados por el JPEG, formado por la Organización Internacional de Normalización (ISO, *International Organization for Standardization*) y la Unión Internacional de Telecomunicaciones (ITU, *International Telecommunication Union*) y los sistemas desarrollados específicamente para su uso en el espacio por el CCSDS, organización formada por las principales agencias espaciales.

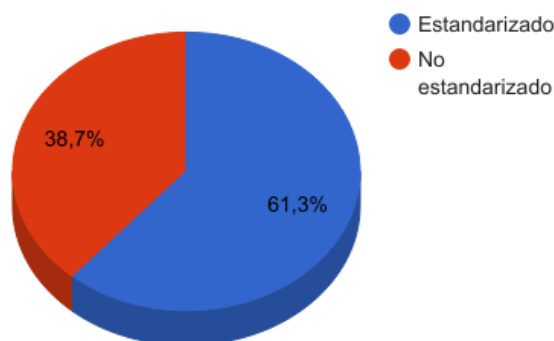


Fig. 2.9. Distribución de los sistemas de compresión de acuerdo a la estandarización.

2.2.4. Contexto nacional

Respecto a la compresión de imágenes a bordo en el contexto nacional, en Argentina el único antecedente fue en la cámara multiespectral de resolución media (MMRS, *Multispectral Medium Resolution Scanner*) del satélite SAC-C (2000) de la CONAE, que utilizó JPEG-Baselines, implementado en un circuito integrado comercial [42]. De esta manera se lograba una tasa de compresión configurable en 10 ó 4 [43]. No obstante durante la operación normal del satélite no se realizó la compresión a bordo y las imágenes se bajaron en su formato original.

2.2.5. Proyección a futuro

Los algoritmos antes mencionados son de compresión mono-espectral donde, en imágenes multiespectrales, cada una de las bandas se comprime individualmente sin explotar la correlación espectral. Las proyecciones a futuro apuntan a la aplicación de técnicas de descorrelación espectral en imágenes multiespectrales e hiperspectrales [44, 45]. El CCSDS publicó en 2012 la recomendación CCSDS-*Multispectral and Hyperspectral Data Compression* (CCSDS-MHDC), un algoritmo sin pérdidas basado en predicción para compresión de imágenes multiespectrales e hiperspectrales [46–48]. Una

comparación entre esta recomendación y una técnica que aplica KLT para la descorrelación espectral y luego JPEG2000 puede encontrarse en [5].

Otra tendencia es la compresión selectiva, que consiste en usar diferentes tasas de compresión según la porción de la imagen sea de interés o no (por ej. comprimir más las áreas cubiertas por nubes) [49, 50].

En el presente trabajo no se relevaron implementaciones de sistemas de compresión con descorrelación espectral y compresión selectiva en misiones espaciales.

2.3. Preselección de sistemas de compresión

Entre todos los algoritmos relevados se seleccionó un grupo de sistemas de compresión para ser evaluados. De la lista de algoritmos de la Tabla 2.1 se descartaron todos aquellos que sólo permiten compresión sin pérdidas (LOCO-I, JPEG *Lossless*, CCSDS-LDC) debido a que son ajenos al estudio propuesto y no alcanzan la tasa de compresión deseada para la CAR. Luego se descartaron todos los sistemas de compresión que fueron diseñados específicamente (DPCM, ADPCM, SAICS, DCT-DE, ICM, DWT-DE, MRCPB, ICER) debido a la ausencia de información completa para su implementación, además es conveniente utilizar sistemas de compresión estandarizados para aprovechar la documentación e implementaciones existentes.

Finalmente los algoritmos preseleccionados son: CCSDS-IDC y JPEG2000 basados en DWT, JPEG-LS basado en predicción y JPEG-Baselines basado en DCT. A continuación se realiza una breve descripción de cada uno de ellos.

2.4. Algoritmo JPEG-LS

Como se ha mencionado en la Sección 2.2.1.1, el modo sin pérdidas de JPEG es ineficiente y como resultado, la ISO junto a la ITU desarrollaron un nuevo estándar para la compresión sin pérdidas y casi sin pérdidas basado en el algoritmo LOCO-I. En esta sección se presenta un resumen sobre el funcionamiento del algoritmo JPEG-LS mientras que las bases teóricas pueden verse en [27, 28] y los detalles para su implementación en [51].

JPEG-LS es un método de baja complejidad que provee compresión sin pérdidas y casi sin pérdidas. Básicamente, JPEG-LS consiste en dos etapas independientes llamadas modelado y codificación. En el modo casi sin pérdidas

el máximo error absoluto entre la imagen original y la comprimida no será mayor al valor del parámetro de configuración *NEAR*. En primer lugar se describe la operación del compresor en su modo sin pérdidas y al final de la sección se mencionan las modificaciones para su operación en el modo casi sin pérdidas.

Un diagrama de bloques simplificado se muestra en la Fig. 2.10. Dada una imagen, JPEG-LS la escanea iniciando por el primer píxel de la primera fila, avanza por la fila hasta su fin y luego continúa con la fila inmediata inferior hasta el fin de la misma. Este modo de lectura se denomina *raster scan*. La etapa de modelado se basa en el contexto de un píxel que, para un dado píxel x , depende de los vecinos a , b , c y d , previamente codificados. Esto indica que, el compresor sólo necesita almacenar una fila de la imagen.

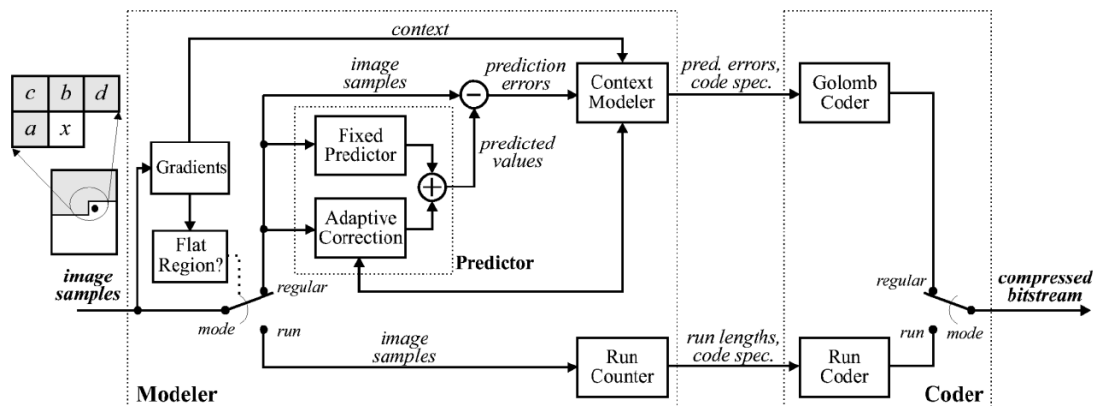


Fig. 2.10. Diagrama de bloques del compresor JPEG-LS [27].

El primer paso para determinar el contexto es calcular las diferencias dadas por

$$g_1 = d - b; \quad g_2 = b - c; \quad g_3 = c - a; \quad (2.7)$$

que representan los gradientes y muestran si la zona que rodea al píxel presenta una textura suave o de bordes, la cual gobierna estadísticamente la predicción del error. Luego se cuantifican los gradientes según 3 niveles de umbral predefinidos, T_1 , T_2 , T_3 . Para cada g_i , donde $i = \{1, 2, 3\}$, como resultado se obtiene el valor del gradiente cuantificado, q_i , que puede tomar los valores $\{-4, -3, -1, 0, 1, 2, 3, 4\}$, como se ilustra en la Fig. 2.11. Así el contexto queda determinado por la terna $C_t = [q_1, q_2, q_3]$. Si el primero de los q_i no nulo es negativo, debe invertirse el signo de todos los elementos de C_t y la bandera de signo, $SIGN$, toma el valor -1 , de lo contrario $SIGN = +1$. Finalmente C_t se mapea a un entero sin signo Q . La cuantificación mencionada no tiene la finalidad de introducir pérdida sino de reducir la cantidad de contextos considerados diferentes.

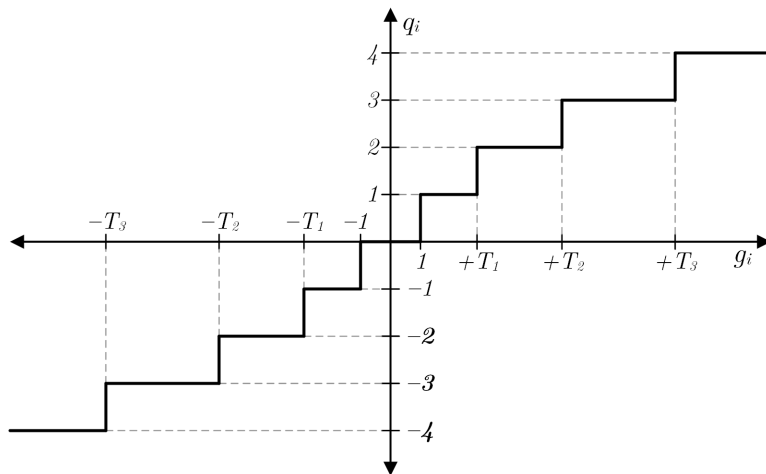


Fig. 2.11. Cuantificación de gradientes.

Considerando nuevamente los gradientes (2.7), si los tres valores son cero el codificador selecciona el modo *Run Length Encoding* (RLE), de lo contrario selecciona el modo regular.

Modo Regular

Está basado en una predicción adaptativa y codificación de entropía con códigos de Golomb-Rice. La predicción adaptativa está compuesta por dos partes, una predicción fija y una corrección adaptativa dependiente del contexto.

Para la predicción fija del píxel x se emplea el predictor *Median Edge Detector* (MED) dado por

$$\hat{x}_{MED} = \begin{cases} \min(a, b) & \text{si } c \geq \max(a, b) \\ \max(a, b) & \text{si } c \leq \min(a, b) \\ a + b - c & \text{otro caso.} \end{cases} \quad (2.8)$$

En caso de de un borde vertical se escoge b como valor de la predicción, en caso de un borde horizontal se elige a y si no existe un borde se utiliza el valor dado por $a + b - c$.

Para una dada secuencia de píxeles, el error con la predicción fija, $\epsilon' = x - \hat{x}_{MED}$, si bien presenta una distribución geométrica bilateral, se encuentra desplazada del origen una cantidad μ . Esta desviación es debida a la limitación de los valores enteros y derivas en el predictor. El caso más general, que se muestra en la Fig. 2.12, es considerar que la desviación está compuesta por una parte entera R y una decimal ρ . La parte adaptativa del predictor, calcula un valor de corrección para cada contexto, $C[Q]$, que cancela R y corrige la distribución

de los errores de predicción. Como resultado se obtiene una distribución con una desviación con $0 \leq \rho < 1$ que puede ser eficientemente codificada por códigos Golomb-Rice. El valor de corrección para cada contexto, $C[Q]$, podría calcularse como la media de los errores de predicción de los píxeles de un dado contexto, aunque esto atenta contra el requisito de diseño del algoritmo de ser de baja complejidad. En su lugar se calcula la corrección de una manera alternativa, a través de unas pocas operaciones de sustracción y adición, que se basa en acumular ϵ' para cada contexto y la cantidad de veces que ocurrió el contexto Q . Estos acumuladores se dividen a la mitad periódicamente cada una cantidad de muestras especificada por el parámetro de configuración *RESET*, por defecto $RESET = 64$.

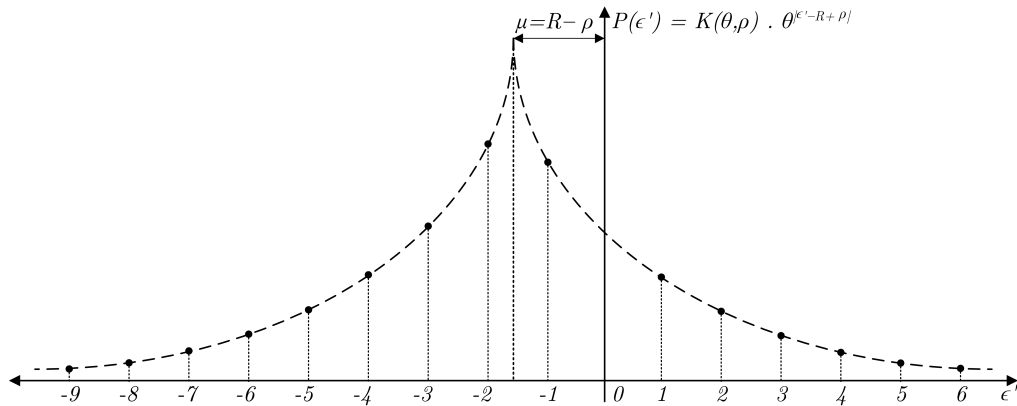


Fig. 2.12. Distribución geométrica bilateral desviada.

Entonces el valor de la predicción adaptativa, que contempla la predicción fija y la corrección adaptativa, queda dado por

$$\hat{x} = \begin{cases} \hat{x}_{MED} + C[Q] & \text{si } SING = +1 \\ \hat{x}_{MED} - C[Q] & \text{si } SING = -1. \end{cases} \quad (2.9)$$

Finalmente el error de predicción se calcula haciendo la diferencia entre el valor del píxel y su predicción, y si $SIGN = -1$ se invierte el signo del error de predicción, es decir

$$\epsilon = \begin{cases} (x - \hat{x}) & \text{si } SING = +1 \\ -(x - \hat{x}) & \text{si } SING = -1. \end{cases} \quad (2.10)$$

Considerando que el mínimo valor posible de x es 0 y el máximo $MAXVAL = \alpha - 1 = 2^B - 1$, para un dado \hat{x} , ϵ toma valores en el rango $-\hat{x} \leq \epsilon < (\alpha - 1) - \hat{x}$. Como \hat{x} es conocido en el decodificador (se predice basado

en los píxeles antes decodificados) es posible reducir la representación de ϵ al rango $-\lceil \alpha/2 \rceil \leq \epsilon < \lceil \alpha/2 \rceil - 1$.

La codificación de Golomb-Rice sirve para codificar muestras enteras sin signo con una distribución geométrica unilateral, por lo tanto es necesario modificar los valores del error de predicción que son enteros con signo con una distribución geométrica bilateral. Para esto se aplica el mapeo

$$\epsilon_m = M(\epsilon) = \begin{cases} 2 \cdot \epsilon & \text{si } \epsilon \geq 0 \\ 2 \cdot |\epsilon| - 1 & \text{si } \epsilon < 0 \end{cases} \quad (2.11)$$

y los valores de ϵ_m quedan en el rango $0 \leq \epsilon_m \leq \alpha - 1$.

Dado un número n , su representación en código de Golomb-Rice, $G(n, k)$ está dada por la representación unaria de $q = \lfloor n/2^k \rfloor$ (q ceros seguidos de un uno) concatenada con los k -bits menos significativos de n . Se tendrán distintos códigos de Golomb-Rice acorde al parámetro k , donde dependiendo de las características de la distribución estadística, existirá un valor de k que permite codificar de manera óptima. Recordando que el contexto Q caracteriza la estadística del píxel a codificar, el valor de k óptimo para codificar ϵ_m se calcula a partir del contexto.

El proceso de codificación antes descrito puede producir una significativa expansión para una simple muestra, por ejemplo, suponiendo una imagen de $B = 8bpp$, un $\epsilon = -128 \Rightarrow \epsilon_m = 255$ con $k = 0$ es codificado con una palabra de $256bits$, una expansión de 32 veces. Si bien este efecto no produce un gran impacto en la tasa de compresión de una imagen, ya que ocurre con poca frecuencia, causa problemas de implementación. Por lo tanto JPEG-LS implementa una versión de la codificación de Golomb-Rice de longitud limitada, $LG(n, k, L_{Max})$, donde la longitud de un código se limita a $L_{Max} = 4 \cdot 2^B$. Dado $q_{max} = L_{Max} - \lceil \log_2 \alpha \rceil - 1$, si $q < q_{max}$ el código de Golomb-Rice de longitud limitada es $LG(n, k, L_{Max}) = G(n, k)$. De otra manera el código será la representación unaria de q_{max} seguida por los $(\lceil \log_2 \alpha \rceil)$ -bits menos significativos de $n - 1$.

Modo RLE

Si los valores de los tres gradientes de (2.7) son cero se ingresa al modo RLE. En este modo, el codificador empieza por el píxel x y cuenta la cantidad de píxeles consecutivos que son iguales a a , hasta llegar al final de la fila o encontrar un píxel que no cumpla la condición. Ya que todos los píxeles contados son iguales

a a , y considerando que a ya fue codificado, sólo se necesita codificar el largo de la secuencia.

Compresión casi sin pérdidas

Si se ha seleccionado el modo casi sin pérdidas, luego de calcular el error de predicción y antes de la reducción de rango, tiene lugar la cuantificación del error de predicción, dada por

$$\epsilon = \text{sign}(\epsilon) \cdot \left\lfloor \frac{|\epsilon| + NEAR}{2 \cdot NEAR + 1} \right\rfloor. \quad (2.12)$$

Esta forma de cuantificar garantiza que cuando en el decodificador reconstruya el valor de píxel x , el valor de la reconstrucción, Rx , no diferirá del valor original en un valor absoluto mayor a $NEAR$. Si $NEAR = 0$ no se realiza la cuantificación y se tiene el modo sin pérdidas.

Además los siguientes aspectos del proceso de codificación son afectados por la cuantificación. El modelado del contexto y predicción son realizados con los valores de los píxeles reconstruidos, Ra , Rb , Rc y Rd . El valor de un dado píxel se reconstruye según

$$Rx = \hat{x} + SIGN \cdot \epsilon \cdot (2 \cdot NEAR + 1). \quad (2.13)$$

El valor de α se cambia por el resultado de

$$\alpha' = \left\lfloor \frac{\alpha + 4 \cdot NEAR}{2 \cdot NEAR + 1} \right\rfloor. \quad (2.14)$$

La condición para entrar en el modo RLE se relaja a que los valores g_i satisfagan $|g_i| \leq NEAR$ para $i = \{1, 2, 3\}$. Una vez dentro del modo RLE el codificador considera como iguales a los píxeles que difieren de Ra en un valor absoluto menor o igual a $NEAR$.

2.5. Algoritmo JPEG-Baselines

JPEG fue el primer estándar internacional de compresión de imágenes de nivel de gris y de color, fue publicado por la ISO e ITU por primera vez en 1991, y tiene cuatro modos de operación: codificación sin pérdidas, codificación progresiva, codificación jerárquica y codificación secuencial. Adicionalmente en cada modo puede utilizarse codificación de Huffman o codificación aritmética. El modo sin pérdidas (JPEG *Lossless*) está basado en predicción, mientras que los

tres modos restantes son con pérdidas y están basados en DCT, particularmente el último modo, con codificación de Huffman, es el denominado JPEG-Baselines que es el método de compresión más utilizado para imágenes en internet. El estándar JPEG especifica compresión para imágenes de $8bpp$ y $12bpp$ solamente, mientras que formatos distintos pueden acomodarse dentro de estos formatos pero es responsabilidad del usuario. En [26] puede encontrarse una descripción general sobre el estándar JPEG que aborda los cuatro modos, mientras que en el documento del estándar publicado por la ISO [51] se encuentran todos los detalles para su implementación, a continuación se presenta un resumen sobre el modo JPEG-Baselines.

Aunque no aplica para el presente caso de estudio es menester mencionar, que para imágenes RGB, antes de iniciar la compresión se realiza una transformación al espacio de color de luminancia y crominancia [14]. Este paradigma, orientado a cómo un humano ve las imágenes, supone que la información de color (crominancia) no es importante y por lo tanto se comprime más que la información de luminancia. En este trabajo se aplica la compresión independiente a cada banda y cada una de éstas es tratada como una componente de luminancia.

En la Fig. 2.13 se muestra un diagrama de bloques del sistemas de compresión JPEG-Baselines. Previo a la transformación se realizan dos operaciones. En primer lugar, dada una imagen se forman subimágenes de 8×8 píxeles y luego cada bloque es procesado independientemente. En caso de que la imagen no tenga una cantidad de filas o columnas múltiplo de 8, se repiten las filas o columnas adyacentes tantas veces como sea necesario. En segundo lugar, se resta a cada muestra la cantidad 2^{B-1} para trasladar las muestras del rango de enteros sin signo $[0, 2^B - 1]$ al rango de entero con signo $[-2^{B-1}, 2^{B-1} - 1]$.

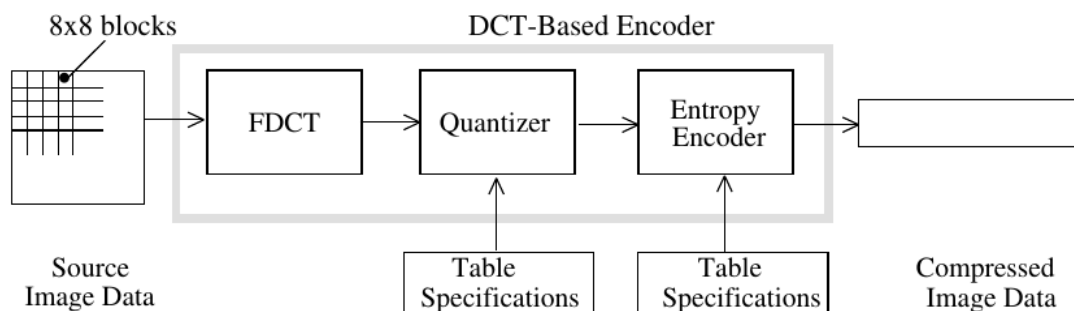


Fig. 2.13. Diagrama de bloques del compresor JPEG-Baselines [26].

Luego se realiza la DCT 2D, acorde a lo mencionado en la Sección 2.1.2, aunque no necesariamente se realiza el producto de matrices, existen otros

métodos que requieren menor cantidad de operaciones [52]. Como resultado se obtienen 64 coeficientes que pueden ser entendidos como la cantidad relativa de frecuencia espacial 2D contenida en en la subimagen de entrada. El coeficiente con frecuencia cero en ambas dimensiones, \mathbf{T}_{00} , es llamado “coeficiente DC” y los restantes 63 coeficientes son llamados “coeficientes AC”. Debido a que los valores de las muestras típicamente varían lentamente de un punto a otro a través de la imagen, la mayoría de la información queda concentrada en los coeficientes de baja frecuencia mientras que los coeficientes de alta frecuencia tienen valores cercanos o igual a cero.

Luego de la DCT, cada uno de los 64 coeficientes es uniformemente cuantificado dividiendo cada uno de ellos por el correspondiente paso de cuantificación acorde a (2.5). Puede utilizarse la tabla que especifica el estándar por defecto,

$$\mathbf{Q}_b = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}, \quad (2.15)$$

o definir nuevas tablas acorde a la aplicación. La tabla de cuantificación debe estar compuesta por valores enteros en el rango $[1, 2^8]$ o $[1, 2^{12}]$ según corresponda. Una tabla de cuantificación especificada por el usuario puede obtenerse a partir de la expresión

$$\mathbf{Q}_{ij} = 1 + (i + j) \times R, \quad (2.16)$$

donde R toma valores enteros positivos [15].

Para obtener distintas tasas de compresión, y consecuentemente calidad de imagen, puede escalarse la matriz de cuantificación predefinida.

Previo a la codificación de entropía tienen lugar dos operaciones preparatorias. En primer lugar la matriz de coeficientes se ordena de acuerdo a un escaneo en zigzag iniciando desde el coeficiente DC y terminando en el coeficiente \mathbf{T}_{77} , como se muestra en la Fig. 2.14(a). De esta manera la matriz se transforma en un vector. Una vez encontrado el último coeficiente no nulo el próximo elemento toma el valor *EOB* que indica el final del bloque y luego es codificado de manera especial indicando que el resto de los coeficientes son cero.

A modo de ejemplo, dada la matriz de coeficientes cuantificados,

$$\mathbf{T}_Q = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.17)$$

se obtiene el vector

$$[-26, -3, 1, -3, -2, -6, 2, -4, 1, -4, 1, 1, 5, 0, 2, 0, 0, -1, 2, \dots \\ \dots 0, 0, 0, 0, 0, -1, -1, EOB]. \quad (2.18)$$

En segundo lugar, el coeficiente DC de todos los bloques, excepto el del primer bloque, es transmitido diferencialmente, para ello al valor del coeficiente DC del bloque actual se le resta el valor de coeficiente DC del bloque anterior, ver Fig. 2.14(b). En el ejemplo, suponiendo que el coeficiente DC del bloque anterior es -17 el vector a codificar es

$$[-9, -3, 1, -3, -2, -6, 2, -4, 1, -4, 1, 1, 5, 0, 2, 0, 0, -1, 2, \dots \\ \dots 0, 0, 0, 0, 0, -1, -1, EOB]. \quad (2.19)$$

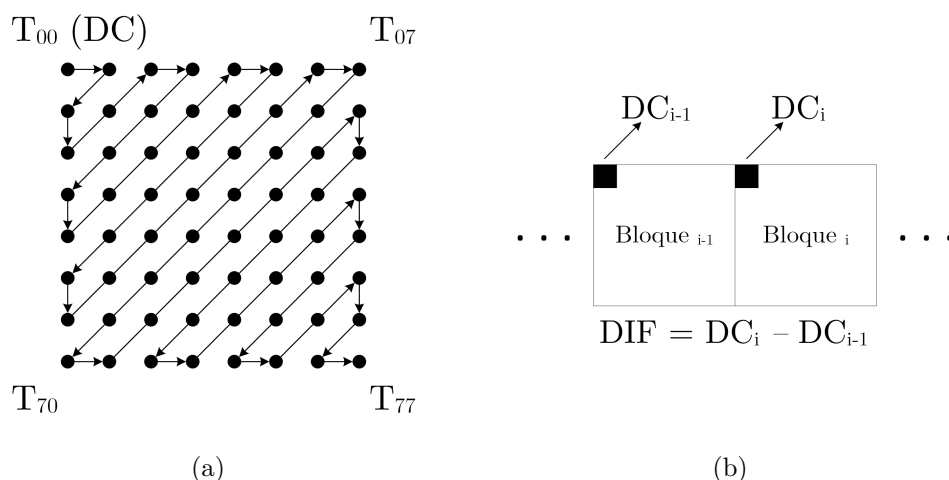


Fig. 2.14. Preparación de coeficientes cuantificados para la codificación de entropía. (a) Escaneo en zigzag y (b) codificación diferencial del coeficiente DC.

Finalmente tiene lugar la codificación de entropía, los coeficientes DC y AC son codificados de distinta manera. Los coeficientes DC son codificados utilizando una tabla códigos de Huffman predefinidos que vincula el valor del

coeficiente con un número binario de longitud variable. Los coeficientes AC son codificados usando una combinación de RLE y codificación de Huffman. Para esto, se utiliza una tabla de doble entrada que de acuerdo a la cantidad de coeficientes nulos que anteceden al coeficiente a codificar y el valor de éste, se determina un código de longitud variable. Las tablas de codificación pueden encontrarse en [53].

El proceso de descompresión involucra las operaciones inversas y puede verse en la Fig. 2.15. Debido a que el código de Huffman por sí solo es instantánea y únicamente decodificable este paso puede lograrse con una tabla de consulta (LUT, *LookUp Table*). Luego los coeficientes son reordenados en una matriz. El proceso inverso a la cuantificación, no reversible, se lleva a cabo por la multiplicación

$$\hat{\mathbf{T}}_{ij} = \mathbf{T}_{\mathbf{Q}}_{ij} \times \mathbf{Q}_{ij}. \quad (2.20)$$

Finalmente se realiza la transformación DWT inversa, se desplazan las muestras sumando a cada una la cantidad 2^{B-1} y se reordenan las subimágenes para componer la imagen.

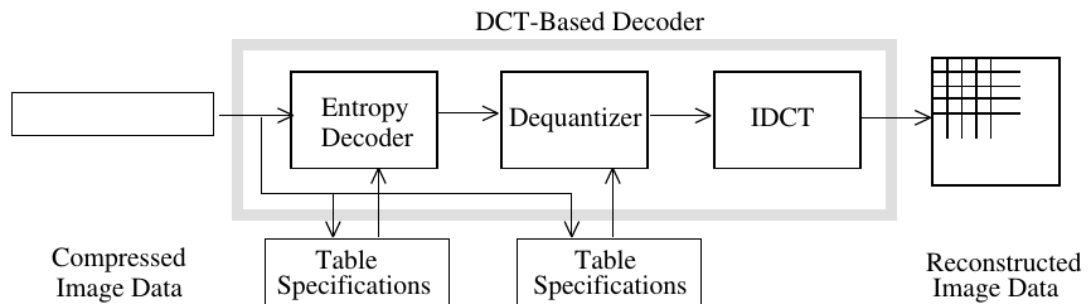


Fig. 2.15. Diagrama de bloques del descompresor JPEG-Baselines [26].

2.6. Algoritmo JPEG2000

Una revisión general sobre el estándar de compresión puede encontrarse en [17], mientras que en la publicación de la ISO [54] pueden encontrarse los detalles para la implementación de un sistema de compresión compatible. Adicionalmente, fueron publicados dos libros específicos del estándar que contienen las bases teóricas sobre el mismo [18, 55]. A continuación se presenta una breve descripción del sistema de compresión, extraída de dichas fuentes.

Uno de los nuevos e importantes enfoques introducidos por la compresión JPEG2000 es el paradigma de *escalabilidad*. Una representación comprimida se dice que es escalable en resolución si contiene subconjuntos identificables

que representan sucesivamente la imagen fuente a menores resoluciones y es escalable en distorsión si contiene subconjuntos identificables que representan la imagen a la máxima resolución pero con sucesivamente menores calidades (más distorsión). El concepto de escalabilidad puede resumirse en la frase “comprimir una vez, descomprimir de muchas maneras”. En el codificador, la imagen se comprime a una resolución y calidad determinada, y en el decodificador la imagen se descomprime en cualquier resolución y calidad igual o inferior a la que se comprimió. En JPEG2000 la escalabilidad en resolución está basada en la DWT y la escalabilidad en distorsión está basada en la cuantificación y codificación integrada (usando EBCOT [56]).

Además JPEG2000 permite descomprimir sólo partes de la imagen, ampliar zonas de la imagen y descomprimir con mayor resolución sólo las porciones que se están observando, y rotar, reflejar o recortar la imagen sin descomprimirla.

Aunque no aplica para el presente caso de estudio, es menester mencionar, que para imágenes RGB, antes de iniciar la compresión se realiza una transformación al espacio de color de luminancia y crominancia. El estándar especifica dos tipos de transformación: una reversible, que debe ser utilizada para compresión sin pérdidas; y una irreversible, que puede ser utilizada para la compresión con pérdidas.

Un diagrama del sistema JPEG2000 puede verse en la Fig. 2.16. Previo al proceso de descorrelación espacial, la imagen puede ser dividida en regiones rectangulares llamadas teselas (del inglés *tiles*) y cada una de éstas se comprime independientemente de la otra. Las teselas pueden tener cualquier tamaño inferior o igual al de la imagen completa. El motivo de estas subdivisiones es que se puedan descomprimir partes de una imagen. Un uso alternativo es para limitar los requerimientos de memoria del sistema de compresión.

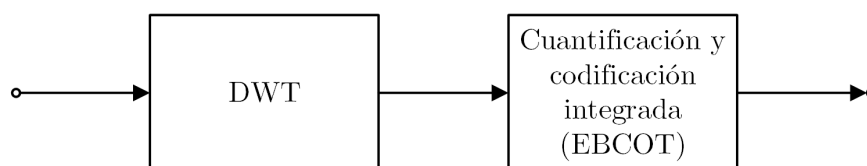


Fig. 2.16. Diagrama de bloques del compresor JPEG2000.

JPEG2000 utiliza la DWT tanto para explotar la redundancia espacial como para dar escalabilidad en resolución. Se realiza la descomposición wavelet de cada tesela, acorde a lo mencionado en la Sección 2.1.3, en una cantidad de

niveles especificada por el usuario. Por defecto, el estándar realiza la DWT 2D 5 veces, lo que implica una estructura de 5 niveles con 16 sub-bandas. Además, se especifican dos opciones de filtros para la DWT:

- La wavelet Daubechies 9/7 de punto flotante, que produce una transformación irreversible y fue seleccionada por su desempeño superior para compresión con pérdidas.
- La wavelet biortogonal 5/3 entera que permite compresión sin pérdidas.

El último bloque en la Fig. 2.16 es la cuantificación y codificación integrada mediante la cual JPEG2000 logra escalabilidad en distorsión. Cada sub-banda de la DWT se divide en bloques rectangulares denominados bloques de código (*code-blocks*) que típicamente tienen un tamaño de 64×64 ó 32×32 coeficientes. Además, estos se agrupan en estructuras espaciales denominadas recintos (*precincts*) que están compuestas por bloques de código de todas las sub-bandas que corresponden a una determinada zona de la imagen. Estos agrupamientos permiten acceder a porciones de la imagen sin descomprimirla en su totalidad. Por defecto se utilizan recintos de gran tamaño ($2^{15} \times 2^{15}$ píxeles) que no benefician la accesibilidad por zonas pero son eficientes respecto a la CR obtenida.

Cada bloque de código es codificado independientemente en un flujo de bits (*bitstream*). Truncar el flujo de bits asociado a un bloque de código tiene el efecto de cuantificar los coeficientes de la DWT. Mientras más larga la cadena de datos menor cuantificación es aplicada.

Debido a que la cadena de bits debe corresponder sucesivamente a cuantificaciones más finas de las muestras de un bloque, la codificación embebida necesariamente está asociada con una familia de cuantificadores. JPEG2000 utiliza cuantificación escalar con zona muerta, donde la zona central, que corresponde a aquellas muestras que son cuantificadas a 0, se denomina zona muerta y es dos veces más grande que el resto de las zonas. Reduciendo sucesivamente a la mitad el paso de cuantificación se obtienen cuantificaciones más finas y de esta manera se da origen a la familia de de cuantificadores que se muestra en la Fig. 2.17.

La estructura de cuantificación puede ser convenientemente asociada con los planos de bit en una representación signo-magnitud de las muestras de cada bloque. Dado un determinado coeficiente, $a_b[u, v]$ de una determinada sub-banda b dentro de un bloque de código; $q_b^{(0)}[u, v]$ es el coeficiente cuantificado asociado

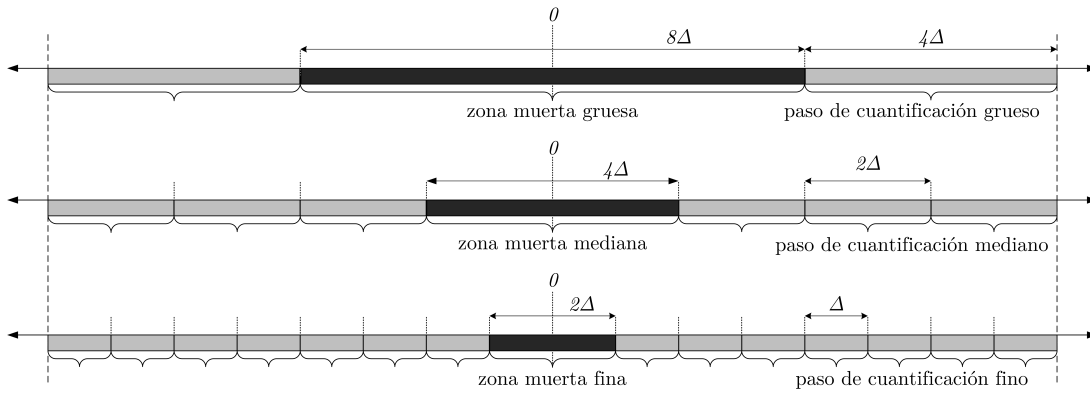


Fig. 2.17. *Cuantificación escalar con zona muerta.*

con el cuantificador de zona muerta más fina, con un paso de cuantificación Δ_b y está dado por

$$q_b^{(0)}[u, v] = \text{sign}(a_b[u, v]) \cdot \left\lfloor \frac{|a_b[u, v]|}{\Delta_b} \right\rfloor. \quad (2.21)$$

Ahora, considerando que $q_b^{(p)}[u, v]$ denota el coeficiente cuantificado de manera más gruesa con un tamaño del paso de cuantificación $2^p \cdot \Delta_b$, éste está dado por

$$q_b^{(p)}[u, v] = \text{sign}(a_b[u, v]) \cdot \left\lfloor \frac{|a_b[u, v]|}{2^p \cdot \Delta_b} \right\rfloor = \text{sign}(q_b^{(0)}[u, v]) \cdot \left\lfloor \frac{|q_b^{(0)}[u, v]|}{2^p} \right\rfloor. \quad (2.22)$$

Como muestra la última expresión, los coeficientes cuantificados de manera más gruesa, $q_b^{(p)}[u, v]$, son obtenidos simplemente descartando los bits menos significativos de la representación binaria de los coeficientes cuantificados de manera más fina.

Basándose en esta observación, un flujo de bits puede ser construido de la manera sugerida por la Fig. 2.18. Asumiendo una representación de K -bits de magnitud y un bit de signo, los coeficientes con la cuantificación más gruesa, $q_b^{(K-1)}[u, v]$, están representados por el bit de magnitud más significativo de cada muestra junto con el bit de signo de aquellas muestras cuyas magnitudes no son cuantificadas a cero. El codificador de plano de bits (BPE, *Bit-Plane Encoder*) avanza a través de cada muestra, codificando esos bits. Si el flujo de bits es truncado a este punto, el decodificador recibe los coeficientes cuantificados de la manera más gruesa, $q_b^{(K-1)}[u, v]$. Luego el BPE se mueve al próximo plano de bits de magnitud, codificando el signo y magnitud de cualquier muestra cuya magnitud sea por primera vez no cero en este plano de bits y los bits del presente plano de aquellas muestras que han sido codificadas en pasos anteriores. Si el flujo de bits es truncado luego de este punto, el decodificador recibe los coeficientes

cuantificados de manera más fina, $q_b^{(K-2)}[u, v]$, y así sucesivamente. Finalmente los bits son codificados con una estrategia de codificación aritmética adaptativa conocida como *MQ coder* [55].

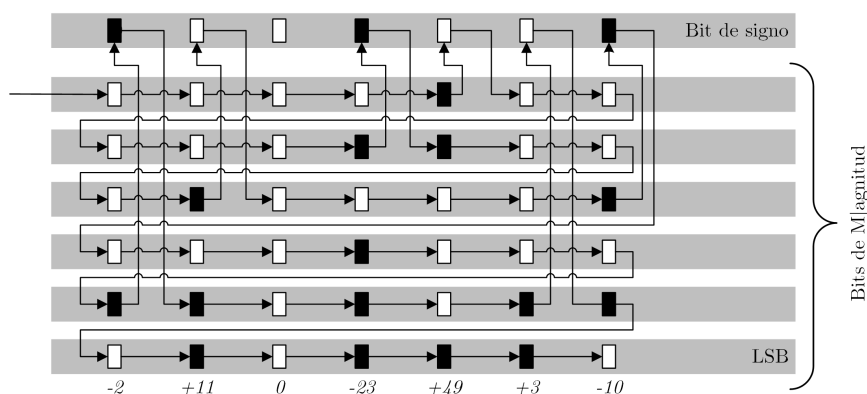


Fig. 2.18. Procedimiento de codificación por plano de bits para datos de 6 bits de magnitud y un bit de signo. Los rectángulos pintados de negro equivalen a un 1 lógico y los rectángulos pintados de blanco equivalen a un 0 lógico.

2.7. Algoritmo CCSDS-IDC

Desde 1998, el grupo de trabajo de compresión de datos del CCSDS comenzó a trabajar en un sistema de compresión de imágenes apto para ser utilizado en misiones espaciales, como resultado en 2005 se publicó la recomendación CCSDS-IDC (o CCSDS 122.0). Esta técnica de compresión puede ser utilizada para lograra compresión de imágenes con pérdidas y sin pérdidas, soporta tanto la operación basado en área como en franja, y es apto para imágenes desde 4 hasta 16bpp.

El compresor está compuesto por dos módulos funcionales, el módulo de DWT, que realiza la decorrelación de las muestras, y un BPE que codifica los datos decorrelacionados, como se ilustra en la Fig. 2.19. Esta arquitectura es similar a la de JPEG2000, pero difiere en que es un algoritmo de baja complejidad especialmente dirigido a instrumentos de alta tasa de datos y puede lograra un desempeño similar al del JPEG2000 pero con una complejidad computacional reducida en un factor de dos. A continuación se realiza una breve descripción funcional del algoritmo, mientras que una introducción puede encontrarse en [33], los detalles para su implementación en [34] y las fundamentaciones teóricas subyacentes son parcialmente tratadas en [41] donde también se muestran resultados del desempeño del algoritmo.

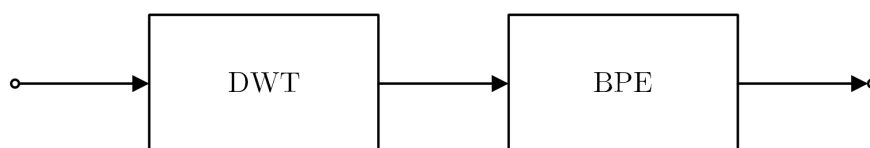


Fig. 2.19. Diagrama de bloques del compresor CCSDS-IDC.

El primer paso para la compresión es la descomposición DWT. Debido a la estructura de bloques que utiliza CCSDS-IDC, previo a realizar la DWT, si la imagen no tiene un número de filas o columnas múltiplo de 8 se rellena, después de la última fila o columna según corresponda, replicando las columnas adyacentes. Las filas o columnas de relleno son eliminadas en el decodificador. Por otro lado, aunque el compresor puede operar con imágenes pequeñas, no se recomienda que las imágenes tengan un ancho o alto menor a 32 píxeles.

Este estándar usa una descomposición DWT 2D de tres niveles y se especifican dos tipos: la DWT bi-ortogonal 9/7, referida como DWT flotante, y una aproximación entera no lineal de ésta, referida como DWT entera o de punto fijo. Mientras que la DWT flotante generalmente muestra una eficiencia de compresión superior en compresión con pérdidas, sólo la DWT entera soporta compresión sin pérdidas. Para la DWT flotante los coeficientes resultantes de la transformación son redondeados al entero más cercano antes del BPE y almacenados en una palabra de $(B+5)$ -bits, para una imagen de entrada de B -bits. En el caso de la DWT entera, antes del BPE, los coeficientes son multiplicados por pesos definidos para cada sub-banda y almacenados en palabras de $(B+4)$ -bits, aunque para algunas implementaciones con pesos definidos por el usuario pueden ser necesarias palabras más largas. En la Tabla 2.3 se muestran los factores de peso por defecto.

Tabla 2.3 – Factores de peso para cada sub-banda (aplica con DWT entera).

Sub-banda	HH_1	HL_1	HL_1	HH_2	HL_2	LH_2	HH_3	HL_3	LH_3	LL_3
peso	2^0	2^1	2^1	2^1	2^2	2^2	2^2	2^3	2^3	2^3

El BPE procesa los coeficientes en grupos de 64, denominados *bloques*, cuya construcción se esquematiza en la Fig. 2.20. Estos son un arreglo de 8×8 coeficientes que contienen: 1 coeficiente de la banda LL_3 (también llamado coeficiente DC); 1 coeficiente de cada una de las sub-bandas de detalle de nivel 3, HL_3 , LH_3 , HH_3 , (también llamados coeficientes padres); 4 coeficientes de cada una de las sub-bandas de detalle de nivel 2, HL_2 , LH_2 , HH_2 , (también llamados coeficientes hijos); y 16 coeficientes cada una de las sub-bandas de detalle de nivel 1, HL_1 , LH_1 , HH_1 , (también llamados coeficientes nietos). Estos

64 coeficientes, 1 DC y 63 AC's, representan una región localizada de la imagen original.

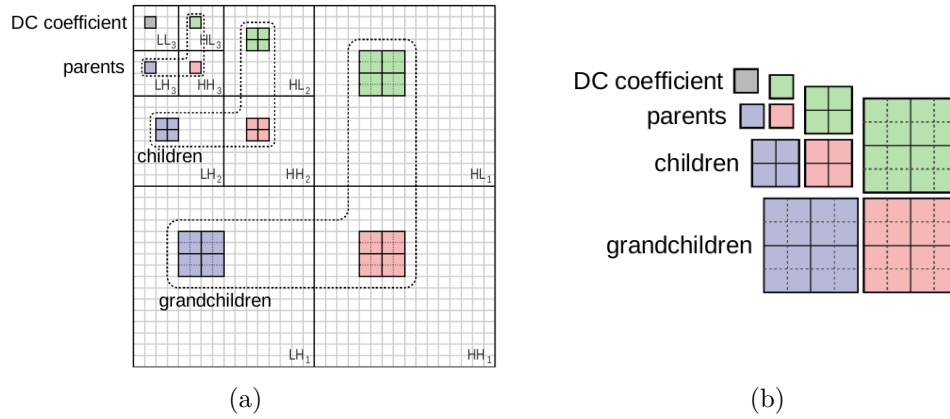


Fig. 2.20. Coeficientes de la DWT organizados en bloques para CCSDS-IDC [41]. (a) Coeficientes de la DWT de una imagen y (b) 64 coeficientes que componen un bloque.

Una vez generados los bloques para toda la imagen, se crean grupos de S bloques consecutivos. Estos nuevos grupos, denominados *segmentos*, son codificados independientemente uno del otro. S puede tomar cualquier valor tal que $16 \leq S \leq 2^{20}$, excepto para el último segmento de la imagen en el cual S puede ser menor a 16. El valor de S puede ser seleccionado acorde a la memoria disponible en el hardware. Además, dada una imagen de M filas y N columnas, y considerando que $\lceil M/8 \rceil$ es la cantidad de bloques en la dirección vertical y $\lceil N/8 \rceil$ la cantidad de bloques en la dirección horizontal, si $S = \lceil M/8 \rceil \times \lceil N/8 \rceil$, todos los bloques de la imagen pertenecen al mismo segmento y se refiere a este modo como *full frame compression* (compresión por área); por otro lado cuando $S = \lceil N/8 \rceil$, cada segmento corresponde a una tira de la imagen (de 8 filas y N columnas) y es referida como *strip compression* (compresión por franjas).

Un segmento de bloques es además dividido en grupos de 16 bloques (excepto el último segmento que puede ser menor) que se denominan *gaggles* en el estándar, aquí se usará el termino manada.

Una vez ordenados los coeficientes en segmentos, antes de llevar a cabo su codificación (independiente del resto de los segmentos), se calcula la cantidad de bits que son necesarios para representar al coeficiente DC más grande en el segmento, $BitDepthDC$, la cantidad de bits que son necesarios para codificar los coeficiente AC en en cada bloque, $BitDepthAC_Block_m$ para $m = \{0, 1, \dots, S-1\}$, y la cantidad de bits que

son necesarios para representar el coeficiente AC más grande del segmento, $BitDepthAC = \max_{m=\{0,1,\dots,S-1\}}[BitDepthAC_Block_m]$.

Finalmente el proceso de codificación se lleva a cabo en 4 pasos que aquí se resumen, la descripción completa puede encontrarse en [34].

- **Paso 1:** codificación del encabezado del segmento. Se codifica el encabezado que contiene: información de la posición del segmento en la trama completa, el número de filas y columnas que se utilizaron como relleno antes de realizar la DWT, S , $BitDepthDC$, $BitDepthAC$ y otras características de la etapa de codificación.
- **Paso 2:** codificación de coeficientes DC. Dependiendo de la relación entre el rango dinámico de los coeficientes DC y los coeficientes AC (es decir $BitDepthDC$ y $BitDepthAC$) se elige un número q y todos los coeficientes DC del segmento se cuantifican desplazándolos a la derecha q bits, es decir, se conservan los $(BitDepthDC - q)$ -bits más significativos. Los coeficientes DC cuantificados de una manada se transmiten codificando la diferencia con el coeficiente anterior usando códigos de Golomb-Rice de manera similar al estándar CCSDS-LDC. Cabe destacar que para el primer coeficiente del segmento no se transmite la diferencia, sino, en su forma original como referencia para los coeficientes posteriores. De esta manera se han codificado los $(BitDepthDC - q)$ -planos de bits más significativos de los coeficientes DC y resta codificar sus q -planos de bits. En el caso que $q > BitDepthAC$ los planos de bits $b = \{q, \dots, BitDepthAC + 1\}$ se agregan a la trama sin codificar. Los restantes $BitDepthAC$ -planos de bits de los coeficientes DC se codifican en la etapa 0 del paso 4.
- **Paso 3:** codificación de $BitDepthAC_Block_m$. Para cada manada, el valor de $BitDepthAC_Block_m$, de los bloques que la componen, se codifican de igual manera que los coeficientes DC cuantificados.
- **Paso 4:** BPE: La codificación por planos de bits se hace desde del plano más significativo hasta el menos significativo, es decir $b = \{BitDepthAC - 1, BitDepthAC - 2, \dots, 0\}$. Cada plano de bits se codifica en 5 etapas, numeradas 0 – 4. Dentro de cada etapa se codifican los bloques dentro del segmento desde el 0 hasta el $S - 1$, como se ilustra en la Fig. 2.21. En la etapa 0, se codifica el plano de bits b de los coeficientes DC de todos los bits (se inicia en $b = BitDepthAC$ ya que los planos superiores de los coeficientes DC fueron codificados en el paso 1). Durante

las etapas 1 – 4 se codifica el plano de bits b de los coeficientes de detalle iniciando por las sub-bandas de nivel 3 y finalizando con las sub-bandas de nivel 1. Para cada plano de bits, se forman palabras que son codificadas por códigos de longitud variable.

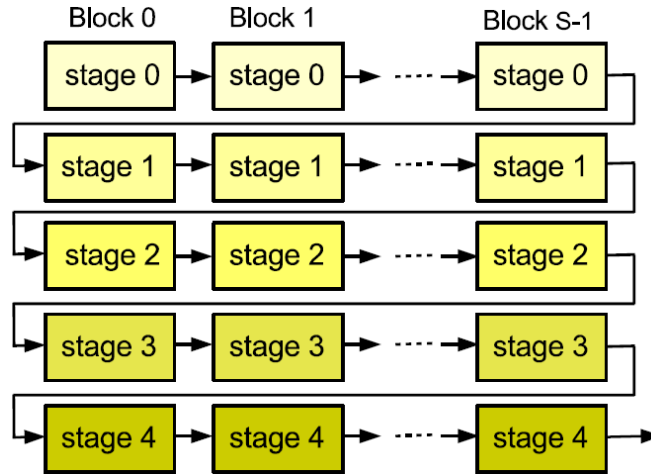


Fig. 2.21. Estructura de codificación de un plano de bits de un segmento [34].

Esta forma de codificación, naturalmente coloca la información de menor resolución, bits más significativos de los coeficientes DC, al principio de la trama y a medida que se avanza en la trama coloca información de mayor detalle hasta llegar a la de mayor resolución al final, con los bits menos significativos de las bandas de detalle de nivel 1. La compresión con pérdidas se logra interrumpiendo la codificación de un dado segmento cuando se alcanza la cantidad de Bytes por segmento límite dada por

$$SegByteLimit = \frac{CR(bpp)}{8} \times 64 \cdot S, \quad (2.23)$$

donde $CR(bpp)$ es el parámetro que especifica el usuario con la tasa de compresión deseada. De esta manera se descarta la información de detalle menos significativa de cada segmento que compone la imagen.

Calidad de imagen

Al remover la información considerada irrelevante durante un proceso de compresión con pérdidas, la calidad de la imagen se ve degradada, por lo tanto es necesario cuantificar la naturaleza de esta pérdida. Dos tipos de criterios de fidelidad pueden ser utilizados para tal evaluación: 1) criterios de fidelidad objetivos y 2) criterios de fidelidad subjetivos. Cuando la pérdida de información puede ser expresada como una función matemática de la entrada y la salida del proceso de compresión, se trata de criterios de fidelidad objetivos o métricas de calidad de imagen objetivas. Para imágenes que finalmente son vistas por humanos, el único método correcto para calificar la calidad de imagen es a través de una evaluación subjetiva. Sin embargo, en la práctica, la evaluación subjetiva es poco conveniente ya que consume demasiado tiempo y es costosa.

El presente capítulo tiene como objeto introducir los aspectos teóricos de las métricas empleadas en esta tesis para cuantificar la degradación de la calidad de imágenes sometidas a un proceso de compresión con pérdidas. Dicho tópico es abordado de dos maneras: en la Sección 3.1 como la degradación de calidad relativa ocurrida por la compresión, para imágenes generales, que es el abordaje clásico basado en métricas objetivas; y en la Sección 3.2 como la degradación relativa que se produce sobre una métrica de calidad absoluta utilizada para imágenes satelitales de alta resolución que tiene relación con un métrica subjetiva.

3.1. Métricas clásicas de degradación de la calidad de imagen

En la bibliografía existe una amplia cantidad de métricas para evaluar la degradación de la calidad de imagen por compresión [10, 14, 15]. En esta sección se tratan las métricas objetivas que normalmente se utilizan para caracterizar la

degradación de la calidad por compresión de imágenes. Éstas pueden dividirse en dos grupos: métricas derivadas de la imagen error, tratadas en la Sección 3.1.1, y métricas basadas en el sistema de visión humano, de las cuales aquí sólo se presenta un índice basado en la estructura de la imagen, tratado en la Sección 3.1.2.

3.1.1. Métricas basadas en la imagen error

Dada una imagen que es sometida a un proceso de compresión y luego a un proceso de descompresión, se considera $I_1[x, y]$ la imagen original e $I_2[x, y]$ la imagen recuperada. Entonces la imagen error, es decir la diferencia píxel a píxel entre la imagen original y la comprimida, está dada por

$$e[x, y] = I_2[x, y] - I_1[x, y], \quad (3.1)$$

si la imagen tiene una única componente espectral, y por

$$e[x, y, z] = I_2[x, y, z] - I_1[x, y, z], \quad (3.2)$$

si tiene varias componentes espectrales. De éstas derivan una serie de métricas de calidad de imagen que se presenta a continuación.

3.1.1.1. Error cuadrático medio

El error cuadrático medio (MSE, *Mean Square Error*) entre la imagen recuperada y la original [10], generalizando para una imagen de varias componentes, está dado por

$$MSE = \frac{1}{M \cdot N \cdot K} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sum_{z=0}^{K-1} e[x, y, z]^2. \quad (3.3)$$

3.1.1.2. Raíz del error cuadrático medio

La raíz del error cuadrático medio (RMSE, *Root Mean Square Error*) entre la imagen recuperada y la original [10], generalizando para una imagen de varias componentes, está dada por

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{M \cdot N \cdot K} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sum_{z=0}^{K-1} e[x, y, z]^2}. \quad (3.4)$$

3.1.1.3. Relación señal a ruido de compresión

La relación señal a ruido (SNR, *Signal to Noise Ratio*) entre la imagen recuperada y la original [10], generalizando para una imagen de varias componentes, suele expresarse tanto en veces como en dB y está dada por

$$SNR_C = \frac{SignalPower}{MSE} = \frac{\frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sum_{z=0}^{K-1} I_1[x, y, z]^2}{MSE} \quad (3.5)$$

y por,

$$SNR_C(dB) = 10 \cdot \log_{10}(SNR) \quad (3.6)$$

respectivamente. Para diferenciar esta métrica de la SNR de una imagen, definida en la Sección 3.2.3, se agrega a la presente el subíndice C .

3.1.1.4. Relación señal a ruido de pico

La relación señal a ruido de pico (PSNR, *Peak Signal to Noise Ratio*) entre la imagen recuperada y la original suele expresarse tanto en veces como en dB y está dada por

$$PSNR = \frac{\max |I_1[x, y]|^2}{MSE} \quad (3.7)$$

y por

$$PSNR(dB) = 10 \cdot \log_{10}(PSNR) \quad (3.8)$$

respectivamente. Esta métrica guarda una relación de escala con SNR_C para imágenes de una única componente y para imágenes de más de una componente PSNR no está definida.

El valor $\max |I[x, y]|$ tiene dos acepciones dependiendo de la bibliografía. Para el autor D. Salomon [15], $\max |I_1[x, y]|$ es el valor máximo que podría tomar un píxel (a fondo de escala), por ejemplo: si se cuantifica con 8 bits $\max |I_1[x, y]| = 255$, si se cuantifica con 12 bits $\max |I_1[x, y]| = 4095$. Para el autor S. Qian [10], $\max |I_1[x, y]|$ es el máximo que realmente toma la imagen original, y además menciona que utilizar el valor a fondo de escala podría resultar en un valor artificialmente alto de PSNR si el valor máximo de la imagen está demasiado lejos del valor a fondo de escala. En este trabajo se adopta el último criterio mencionado.

Esta métrica se utiliza para compresión de imágenes porque, contrario al MSE y RMSE, da valores mayores para mejor calidad de imagen y porque es muy poco sensible a pequeñas variaciones de la imagen reconstruida. Sin embargo tiene una limitada aproximación al error percibido por el sistema de

visión humano. Si bien mayores PSNR implican mayores semejanzas entre la imagen reconstruida y la original no da garantía de que al observador le agrade la imagen. El PSNR es una métrica relativa y es usada solamente para comparar el desempeño de diferentes métodos con pérdidas o el efecto de cambiar diferentes valores paramétricos de un algoritmo.

3.1.1.5. Máxima Diferencia Absoluta

La máxima diferencia absoluta (MAD, *Maximum Absolute Difference*) entre la imagen recuperada y la original [10], generalizando para una imagen de varias componentes, está dada por

$$MAD = \max |e[x, y, z]|. \quad (3.9)$$

Para el sistema de compresión JPEG-LS, MAD será igual al valor de parámetro *NEAR*.

3.1.2. Índice de similitud estructural

Las métricas derivadas del error entre la imagen original y la comprimida son ampliamente utilizadas para evaluar la degradación de la calidad de imagen por compresión, bajo la suposición de que la pérdida de la calidad percibida por un humano está directamente relacionada con el error píxel a píxel. No obstante, dos imágenes distorsionadas con el mismo MSE pueden tener diferentes tipos de error y en una de ellas este error ser mucho más visible, como se ilustra en la Fig. 3.1.

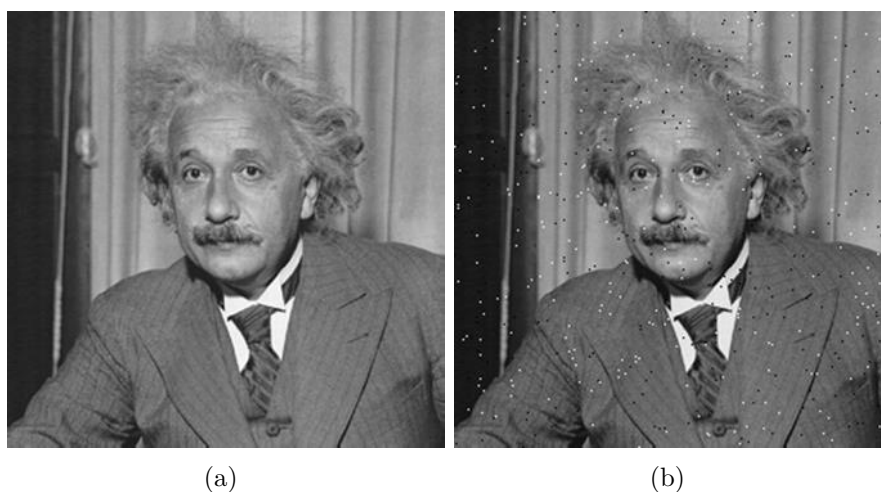


Fig. 3.1. Imágenes con igual MSE pero visualmente distintas. (a) $MSE=144$, $MSSIM=0,988$ y (b) $MSE=144$, $MSSIM=0,840$ [57].

Considerando que debido a la correlación entre píxeles las imágenes portan gran información estructural y que el sistema de visión humano extrae información de la estructura de las mismas, se ha desarrollado una métrica para evaluar la degradación de la calidad de imágenes por compresión basada en la degradación de la información estructural. Esta métrica es llamada índice de similitud estructural (SSIM, *Structural SIMilarity index*) y tiene en cuenta la luminancia, contraste e información de la estructura de la imagen [58].

Considerando que $I_1[x, y]$ es la imagen original e $I_2[x, y]$ es la imagen distorsionada, se calcula en primer lugar una función de comparación de luminancia, $l[x, y]$, en segundo lugar se extrae la información de luminancia de cada imagen y se calcula una función de comparación de contraste, $c[x, y]$, en tercer lugar se extrae la información de contraste de las imágenes y se calcula una función de comparación de la estructura entre las imágenes, $s[x, y]$, por ultimo se combinan las tres funciones de comparación para obtener el SSIM. Este proceso, ilustrado en la Fig. 3.2, da como resultado un mapa que muestra el SSIM entre la imagen original y la distorsionada para cada píxel. Analíticamente se calcula por

$$SSIM[x, y] = \frac{(2 \cdot \mu_1[x, y] \cdot \mu_2[x, y] + C_1) \cdot (2 \cdot \sigma_{12}[x, y] + C_2)}{(\mu_1[x, y]^2 + \mu_2[x, y]^2 + C_1) \cdot (\sigma_1[x, y]^2 + \sigma_2[x, y]^2 + C_2)}. \quad (3.10)$$

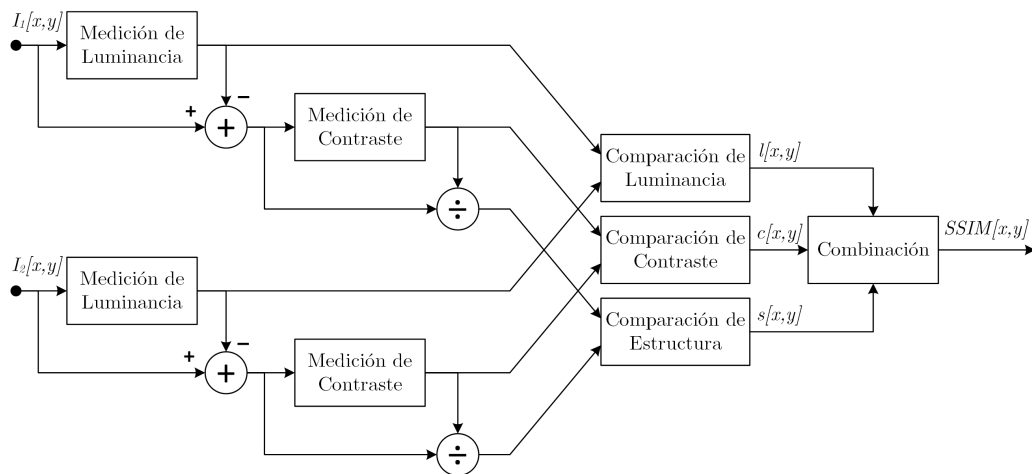


Fig. 3.2. Diagrama de flujo para el cálculo del SSIM [58].

Donde $C_1 = (0,01 \cdot 2^B)^2$ y $C_2 = (0,03 \cdot 2^B)^2$, considerando que B es la cantidad de niveles de cuantificación para cada píxel. Las estadísticas locales μ_1 , μ_2 , σ_1 , σ_2 y σ_{12} , se calculan ponderadas por una ventana Gaussiana circular

simétrica, $\omega[x, y]$, de $\sigma = 1,5$ y tamaño $W \times W$ con $W = 11$, es decir:

$$\begin{aligned}\mu_n[x, y] &= \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} \omega[i, j] \cdot I_n[x-i, y-j] && \text{para } n = \{1, 2\}, \\ &= \omega[x, y] * I_n[x, y]\end{aligned}\quad (3.11)$$

$$\begin{aligned}\sigma_n[x, y]^2 &= \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} \omega[i, j] \cdot \left(I_n[x-i, x-j] - \mu_n[x, y] \right)^2 \\ &= \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} \omega[i, j] \cdot I_n[x-i, x-j]^2 - \mu_n[x, y]^2 && \text{para } n = \{1, 2\} \\ &= \omega[x, y] * I_n[x, y]^2 - \mu_n[x, y]^2\end{aligned}\quad (3.12)$$

y

$$\begin{aligned}\sigma_{12}[x, y] &= \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} \omega[i, j] \cdot \left(I_1[x-i, x-j] - \mu_1[x, y] \right) \cdot \left(I_2[x-i, x-j] - \mu_2[x, y] \right) \\ &= \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} \omega[i, j] \cdot I_1[x-i, x-j] \cdot I_2[x-i, x-j] - \mu_1[x, y] \cdot \mu_2[x, y] \\ &= \omega[x, y] * \left(I_1[x, y] \cdot I_2[x, y] \right) - \mu_1[x, y] \cdot \mu_2[x, y].\end{aligned}\quad (3.13)$$

El $SSIM[x, y]$ es un mapa del tamaño de las imágenes utilizadas para su cálculo que muestra píxel a píxel la similitud entre las dos imágenes. Puede tomar valores en el intervalo $(-1, 1]$, logrando el máximo valor para todos los píxeles cuando ambas imágenes son iguales y valores menores cuando las imágenes difieren en su estructura [59]. Normalmente en la práctica, se necesita un factor de calidad de toda la imagen y se usa la media del índice de similitud estructural ($MSSIM$, *Mean Structural SIMilarity index*), dado por

$$MSSIM = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} SSIM[x, y].\quad (3.14)$$

En la página web de los autores del artículo sobre el $SSIM$ se provee un código de MATLAB para el cálculo de $SSIM$ y $MSSIM$ [57]. No obstante está desarrollado de manera tal que para imágenes grandes se utiliza una excesiva cantidad de memoria, por lo tanto en este trabajo se modificó para hacer un uso más eficiente de los recursos. Las modificaciones realizadas fueron: usar el tipo de variable necesario, limpiar la memoria cuando una variable ya no se utiliza y procesar las imágenes por franjas.

3.2. Métricas de calidad de imagen absolutas

En la sección anterior se presentaron las métricas normalmente utilizadas en compresión de imágenes, en esta sección se describen brevemente las métricas utilizadas como indicadores de calidad absoluta de una imagen satelital y antecedentes sobre su aplicación para evaluar la degradación de calidad debido a la compresión con pérdidas. Antes de entrar a estos temas, es menester realizar una introducción a las métricas de respuesta en frecuencia espacial de un sistema de imagen.

3.2.1. Métricas de respuesta en frecuencia de un sistema de imagen

Dado un sistema de adquisición de imágenes, éste modifica las propiedades espaciales de la escena que está observando borreandola debido a: la óptica, el detector, la electrónica y el movimiento de la plataforma que soporta al instrumento. Si se supone que la escena que es vista por el instrumento es una fuente puntual tal que en términos de señales representa una función impulso en el espacio, la respuesta del sistema a dicha fuente dependerá de las características de respuesta en frecuencia del instrumento y se denomina función de dispersión de punto (PSF, *Point Spread Function*) [60]. A modo de ejemplo en la Fig. 3.3 se representa la fuente puntual y la respuesta de un sistema de imagen en la dimensión espacial x .

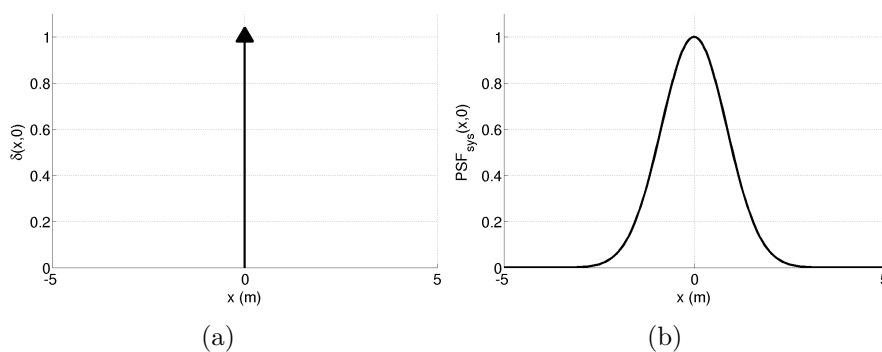


Fig. 3.3. (a) Función impulso unitario correspondiente a la fuente puntual y (b) PSF del sistema de imagen.

La función de respuesta en frecuencia del sistema se denomina función de transferencia óptica (OTF, *Optical Transfer Function*) y está dada por la

transformada de Fourier de de la PSF,

$$OTF_{sys}(u, v) = \mathcal{F}\{PSF_{sys}(x, y)\}. \quad (3.15)$$

Habitualmente no se utiliza la OTF sino la función de transferencia de modulación (MTF, *Modulation Transfer Function*) que es su módulo,

$$MTF_{sys}(u, v) = |OTF_{sys}(u, v)|. \quad (3.16)$$

Muchas veces se caracteriza la respuesta en frecuencia de un sistema de imagen por el valor de la MTF en una determinada dirección a la frecuencia de Nyquist, $F_{Ny} = F_s/2$ donde F_s es la frecuencia de muestreo espacial que queda determinada por el tamaño del píxel [61].

Alternativamente, también se utilizan para caracterizar la respuesta en frecuencia de un instrumento óptico, la función de dispersión de línea (LSF, *Line Spread Function*) que es la respuesta del sistema a una señal que es una línea en una de las direcciones y la función de dispersión de borde (ESF, *Edge Spread Function*) que es la respuesta del sistema a una señal de un borde en una de las direcciones [61]. Así, para una línea a lo largo del eje y , se tiene la LSF en la dirección x de la forma

$$LSF_x(x) = \delta(x) * PSF(x, y) = \int_{-\infty}^{\infty} PSF(x, y) dy, \quad (3.17)$$

y para un borde cuyo flanco está sobre el eje y , se tiene la ESF en la dirección x de la forma

$$ESF_x(x) = u(x) * PSF(x, y) = \int_{-\infty}^x \int_{-\infty}^{\infty} PSF(x', y) dy dx'. \quad (3.18)$$

En los instrumentos ópticos de alta resolución es común que una vez que las imágenes son descargadas a tierra se aplique un proceso de compensación de la MTF (MTFC, *MTFC Compensation*), que realiza la respuesta a frecuencias próximas a F_{Ny} , para remover o reducir el efecto del borroneo producido durante la adquisición [60, 62]. Dicho proceso se aplica por medio de un filtrado espacial que consiste en hacer la convolución de la imagen digital con el núcleo (*kernel*) del filtro de compensación, $Kernel_{MTFC}$,

$$I'[x, y] = I[x, y] * Kernel_{MTFC} \quad (3.19)$$

donde $Kernel_{MTFC}$ queda definido por una matriz de m filas por n columnas

$$Kernel_{MTFC} = \begin{bmatrix} k_{0,0} & k_{0,1} & \cdots & k_{0,n-1} \\ k_{1,0} & k_{1,1} & \cdots & k_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m-1,0} & k_{m-1,1} & \cdots & k_{m-1,n-1} \end{bmatrix}. \quad (3.20)$$

En estos casos se define una PSF neta que contempla tanto la respuesta en frecuencia del sistema como la respuesta en frecuencia de la MTFC, es decir,

$$PSF_{net}(x, y) = PSF_{sys}(x, y) * \mathcal{F}^{-1}\{MTFC(u, v)\} \quad (3.21)$$

y la respectiva MTF neta está dada por

$$MTF_{net}(u, v) = MTF_{sys}(u, v) \cdot MTFC(u, v). \quad (3.22)$$

3.2.2. NIIRS

El *National Imagery Interpretability Rating Scale* (NIIRS) es un índice utilizado para cuantificar la interpretabilidad o utilidad de una imagen adquirida por un sistema de imagen. El NIIRS surge como una manera más simple y directa de evaluar la calidad de una imagen ya que métricas como GSD y MTF no entregan la información suficiente o no son simples de interpretar por los usuarios [63].

El NIIRS se basa en el concepto de que un analista de imagen debería ser capaz de realizar tareas de interpretación más demandantes con imágenes de mayor calidad. El NIIRS consta de 10 niveles graduados (0 a 9), con varias tareas de interpretación o criterios que forman cada nivel. Estos criterios indican la información que puede ser extraída de una imagen de un dado nivel de interpretación. Por ejemplo, con una imagen PAN de NIIRS = 2 un analista debería ser capaz de detectar grandes hangares en un aeropuerto, mientras que con una imagen PAN de NIIRS = 6 el analista debería ser capaz de distinguir entre modelos de helicópteros medianos [63]. Aunque el NIIRS está definido como una escala de números enteros, calificaciones decimales son realizadas para medir pequeñas diferencias entre dos imágenes. Una diferencia de NIIRS de 0,1 por lo general no es visualmente perceptible y no afecta a la interpretabilidad de la imagen, mientras que una diferencia mayor a 0,2 es fácilmente perceptible [64].

Las primeras versiones del NIIRS se enfocaron principalmente en equipamiento militar, es decir, las tareas de interpretación hacen referencia a estructuras militares como aviones, radares, tanques, etc. De esta manera se construyeron las primeras 4 escalas de NIIRS que se muestran en la Tabla 3.1 para imágenes PAN, de radar, térmicas y MS. Luego, se construyó una escala con tareas de interpretación basadas en estructuras civiles (*Civil NIIRS*). La lista completa de criterios de interpretabilidad y nivel de NIIRS para todas las escalas mencionadas puede verse en [63, 65].

Tabla 3.1 – Escalas de NIIRS [63].

Escala	Tipo de Imagen
<i>Visible NIIRS</i>	Imágenes PAN
<i>Radar NIIRS</i>	Imágenes de radar de apertura sintética
<i>IR NIIRS</i>	Imágenes del espectro IR térmico
<i>MS NIIRS</i>	Imágenes MS que abarcan el espectro visible, NIR y SWIR.
<i>Civil NIIRS</i>	Imágenes PAN

El NIIRS estandariza la medición de la interpretabilidad de la imagen para: comunicar la utilidad de una imagen, especificar requerimientos de imagen, ayudar en el diseño y evaluación de futuros sistemas de imagen, y evaluar el desempeño de instrumentos durante su operación [63]. Si bien el NIIRS es una métrica subjetiva ya que el nivel lo determina un analista de imagen, en la próxima sección se describe una ecuación que permite predecir el mismo como función de una serie de parámetros físicos del instrumento.

3.2.3. Ecuación general de calidad de imagen

Dado un requerimiento de NIIRS para un instrumento, la tarea del diseñador es lograr un sistema que cumpla el desempeño deseado. Por lo tanto, se requiere una herramienta que permita predecir precisamente el desempeño del instrumento en términos del NIIRS previo a su construcción y ensayo. La ecuación general de calidad de imagen (GIQE, *General Image-Quality Equation*) fue desarrollada para proveer tal predicción como función de cinco parámetros físicos del instrumento: GSD, respuesta al borde relativa (RER, *Relative Edge Response*), SNR, G (ganancia de ruido) y H (altura del rebasamiento de borde), que se explican mas adelante. Existen dos versiones de la ecuación, GIQE 3.0, publicada en 1994, y GIQE 4.0, publicada en 1997 como resultado de una revisión

y validación de la versión anterior. En este trabajo se utiliza la GIQE 4.0 ya que predice el NIIRS con menor error [66].

La GIQE 4.0 fue desarrollada usando un modelo de regresión. Analistas de imágenes estimaron el NIIRS para un lote de imágenes. La mitad de estas imágenes fueron utilizadas para ajustar la ecuación y la otra mitad para validar la misma. La validación mostró que la GIQE 4.0 puede predecir el NIIRS con una desviación estándar de 0,3. En la Tabla 3.2 se muestra el rango de valores de los parámetros de las imágenes del lote de datos completo. La precisión y validez de la GIQE fuera de estos límites se desconoce [66].

Tabla 3.2 – Rango de valores de validación de la GIQE 4.0 [66].

Parámetro	Mínimo	Máximo
GSD	0,0762 m [3 in]	2,032 m [80 in]
RER	0,2	1,3
H	0,9	1,9
G	1	19
SNR	2	130

La ecuación general de calidad de imagen está dada por

$$NIIRS = c_0 + c_1 \log_{10}(GSD_{GM}) + c_2 \log_{10}(RER_{GM}) + c_3 \frac{G}{SNR} + c_4 H_{GM}, \quad (3.23)$$

donde los valores de c_0 , c_1 , c_2 , c_3 y c_4 para cada versión se muestran en la Tabla 3.3. A continuación se explica cada uno de los parámetros de los que ésta depende.

Tabla 3.3 – Coeficientes de la GIQE según la versión.

	c_0	c_1	c_2	c_3	c_4	
GIQE 3.0	11,810	-3,320	3,320	-1,000	-1,480	
GIQE 4.0	$RER \geq 0,9$	10,251	-3,320	1,559	-0,334	-0,656
	$RER < 0,9$	10,251	-3,160	2,187	-0,334	-0,656

Tamaño de píxel del suelo - GSD

GSD_{GM} es la media geométrica entre el GSD en la dirección x (*along-track*) y el GSD en la dirección y (*cross-track*) en pulgadas, es decir

$$GSD_{GM} = \sqrt{GSD_x \times GSD_y}. \quad (3.24)$$

Respuesta al borde relativa - RER

RER_{GM} es la media geométrica entre la RER_x y la RER_y

$$RER_{GM} = \sqrt{RER_x \times RER_y} . \quad (3.25)$$

La RER se calcula a partir de la ESF neta, como la diferencia entre el valor de la ESF a +0,5 píxel y a -0,5 píxel, es decir,

$$RER = ESF(+0,5) - ESF(-0,5), \quad (3.26)$$

como se ilustra en la Fig. 3.4.

Altura del rebasamiento de borde - H

H_{GM} es la media geométrica del la altura de rebasamiento de borde,

$$H_{GM} = \sqrt{H_x \times H_y} . \quad (3.27)$$

H modela el sobrepico en la ESF debido a la MTFc y es medida en el rango que va de 1 a 3 píxeles en incrementos de 0,25 píxel. Si la respuesta al borde crece de forma monótona, H está definida como el valor de la ESF a 1,25 píxeles, de otra manera H es el valor máximo, como puede apreciarse en la Fig. 3.4.

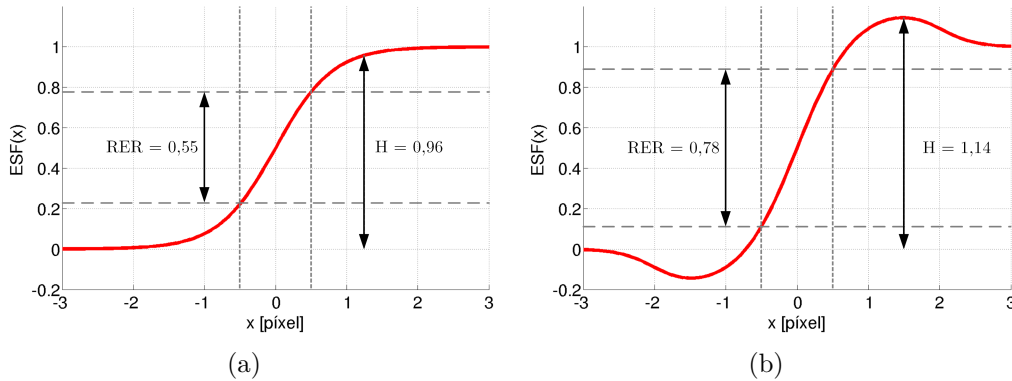


Fig. 3.4. Medición de RER y H en ESF. (a) ESF sin rebasamiento de borde y (b) ESF con rebasamiento de borde.

Ganancia de ruido - G

El parámetro G representa la ganancia de ruido debida a la MTFc y se calcula como:

$$G = \sqrt{\frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} k_{ij}^2} . \quad (3.28)$$

donde k_{ij} es cada uno de los elementos del núcleo del filtro (3.20).

Relación señal a ruido - SNR

Por último la SNR de la imagen también influye en la predicción. En la comunidad de la detección remota existen muchas métricas para definir la SNR, en su forma básica, todas expresan la relación entre la señal y el ruido, pero difieren en cómo se definen la señal y el ruido [64]. La mayoría de las métricas de SNR comparan la señal media de un objetivo, S_{target} , con la desviación estándar del ruido σ_{noise} , es decir

$$SNR = \frac{S}{N} = \frac{S_{target}}{\sigma_{noise}}. \quad (3.29)$$

Si se tienen dos objetivos en una imagen, éstos no pueden ser distinguidos uno del otro si la diferencia entre su nivel de señal es menor a la diferencia del nivel de señal causada por el ruido. Por lo tanto, resulta conveniente definir la señal en términos de la diferencia de reflectancia entre dos objetivos $\Delta\rho = \rho_{high} - \rho_{low}$. Donde los objetivos son superficies lambertianas ¹, uno de reflectancia baja, ρ_{low} , y otro de reflectancia alta, ρ_{high} . Así en la GIQE se define S_{target} como la diferencia de señal que causan en el detector dos superficies lambertianas, una de ρ_{low} y la otra de ρ_{high} ,

$$S_{target} = S_{target}|_{\rho_{target}=\rho_{high}} - S_{target}|_{\rho_{target}=\rho_{low}} = S_{target}|\Delta\rho. \quad (3.30)$$

Para la predicción del NIIRS se suele asumir $\rho_{low} = 7\%$ y $\rho_{high} = 15\%$, aunque pueden usarse los valores reales de otros objetivos [66]. El valor σ_{noise} se calcula teniendo en cuenta todos los ruidos presentes en la cadena de detección [64].

Una forma alternativa para el cálculo de la SNR, basado en imágenes digitales, que aplica cuando éstas contienen un objetivo con una zona clara (de ρ_{high}) y una zona oscura (de ρ_{low}), está dada por

$$SNR = \frac{S_{target}}{\sigma_{noise}} = \frac{\overline{DN}_{\rho_{high}} - \overline{DN}_{\rho_{low}}}{(\sigma_{\rho_{high}} + \sigma_{\rho_{low}})/2}. \quad (3.31)$$

Donde la señal es la diferencia entre el valor promedio de de los DN en la zona clara y el valor promedio de los DN en la zona oscura. Notar que $\overline{DN}_{\rho_{high}}$ y $\overline{DN}_{\rho_{low}}$ son análogos a $S_{target}|_{\rho_{target}=\rho_{high}}$ y $S_{target}|_{\rho_{target}=\rho_{low}}$ respectivamente. Y el ruido está dado por el valor promedio entre la desviación estándar de los DN en la zona clara y los DN en la zona oscura [64].

¹Superficie que refleja la radiación incidente de manera uniforme en todas las direcciones [67].

3.2.4. Caracterización de la calidad de imagen de un instrumento en órbita

Durante las etapas de lanzamiento y operación, un instrumento satelital está sometido a ambientes que pueden alterar la estructura del sistema electro-óptico. Ésto, puede cambiar las condiciones de operación del mismo y producir una degradación en su desempeño. Por tal motivo, se desarrollaron técnicas para la caracterización de la calidad de imagen en órbita.

Una manera de caracterizar la calidad de imagen de un instrumento es de acuerdo su respuesta en frecuencia espacial. Los principales métodos para dicha caracterización son: la caracterización visual, el método de fuente puntual, el método del borde objetivo, el método del banco de pulso y el método de objetivos periódicos [68, 69]. Exceptuando el método de caracterización visual, todos consisten en realizar adquisiciones de fuentes conocidas con el instrumento a caracterizar y luego procesar las imágenes para obtener el parámetro de caracterización en frecuencia deseado (MTF, $MTF@F_{Ny}$, PSF, LSF, ESF, RER, etc.).

El método del borde objetivo goza de la ventaja de que en las imágenes satelitales existe una gran cantidad de estructuras y construcciones que pueden ser utilizadas como objetivos de este tipo y fue ampliamente utilizado para obtener la ESF de un sistema óptico ya sea en laboratorio [70] o en vuelo [71–75]. Adicionalmente, se ha empleado para la caracterización del NIIRS de imágenes satelitales digitales [76–80]. Por dichos motivos el método del borde objetivo fue aplicado en este trabajo para estimar RER, H, SNR y consecuente el NIIRS.

3.2.4.1. Método del borde objetivo

Este método permite reconstruir la ESF a partir de la respuesta a un objetivo (*target*) con una transición abrupta entre un área oscura uniforme y un área clara uniforme, es decir un borde objetivo (*edge target*), como el que se muestra en la Fig. 3.5. Además, una vez obtenida la ESF puede derivarse para obtener la LSF y realizando la transformada discreta de Fourier (DFT, *Discrete Fourier Transform*) puede obtenerse la MTF [68].

Este método puede ser aplicado con objetivos denominados *artificiales* y objetivos denominados *naturales*. Los objetivos artificiales son especialmente diseñados para este fin, como por ejemplo el objetivo del Centro Espacial Stennis (SSC, *Stennis Space Center*) de la Administración Nacional de la Aeronáutica y del Espacio de Estados Unidos (NASA, *National Aeronautics and Space*

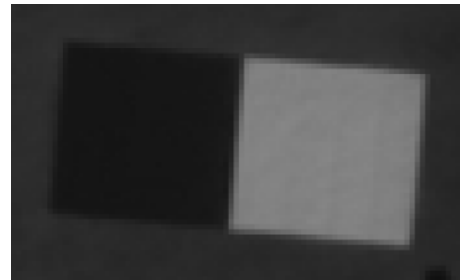
Administration) que consiste en una lona con reflectancia del 3,5% y otra con reflectancia del 52% y se ilustra en la Fig. 3.6. El uso de objetivos naturales consiste en aprovechar construcciones hechas por el hombre que se parecen a bordes objetivos, como por ejemplo una sombra producida por un edificio y campos con diferentes cultivos como los mostrados en la Fig. 3.7.



Fig. 3.5. Borde objetivo ideal.

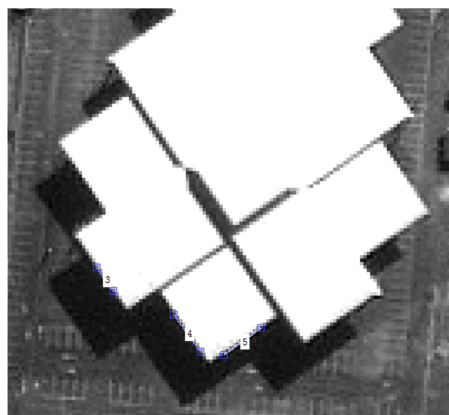


(a)

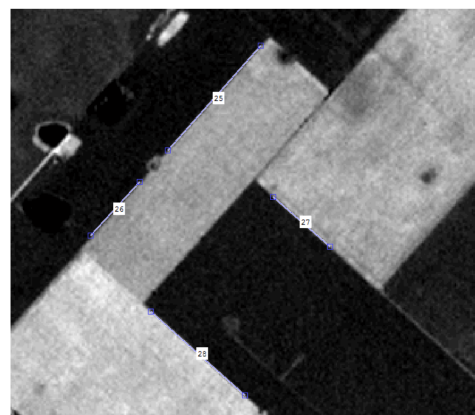


(b)

Fig. 3.6. Borde objetivo artificial del SSC compuesto por lonas de $\rho_{low} = 3,5\%$ y $\rho_{high} = 52\%$ [74]. (a) Tendido del objetivo y (b) imagen satelital del mismo.



(a)



(b)

Fig. 3.7. Bordes objetivos naturales.

Si se considera una única dirección espacial, un borde en la superficie terrestre puede ser considerado una función escalón unitario, $u(x)$. Éste borde resulta filtrado por la PSF del sistema, $PSF(x)$, y se obtiene la respuesta al borde del sistema, como se ilustra en la Fig. 3.8. Luego de ésto se tiene la imagen proyectada en el plano focal donde se encuentra el detector y la señal es muestreada por un intervalo de muestreo igual a la separación entre píxeles, P (que es equivalente al GSD en el espacio objeto). Sin considerar los efectos del ruido introducido en la cadena de detección, puede decirse que la salida de la etapa del muestreo representa la imagen.

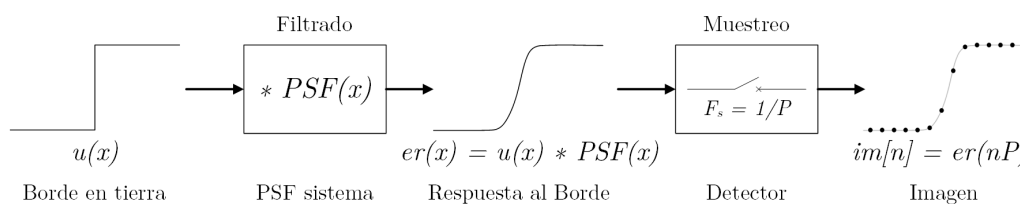


Fig. 3.8. Generación de la respuesta al borde.

El muestreo de la señal en el detector da origen a que la señal del borde resulte submuestreada y no pueda reconstruirse a partir de una única línea en la imagen.

Si ahora se considera una borde objetivo en el espacio bidimensional cuya línea que define el flanco forma un ángulo de cero grados respecto a la línea que describe el paso del satélite, como se muestra en la Fig. 3.9(a), el resultado será que para cada línea de la imagen se tendrá la misma versión del borde submuestreado y tampoco podrá reconstruirse la respuesta al borde del sistema. Si en cambio el borde está inclinado respecto a la línea que describe el paso de satélite, como se muestra en la Fig. 3.9(b), cada línea de la imagen tiene una versión submuestreada del borde pero desplazada respecto a la anterior. Alineando la respuesta de cada línea respecto a la posición del flanco, puede obtenerse la respuesta al borde con mayor cantidad de muestras en una distancia igual a un píxel, permitiendo reconstruir el borde con menor distorsión. Por lo tanto para mitigar el problema del submuestreo, es necesario utilizar objetivos de borde inclinados (*slanted edge target*).

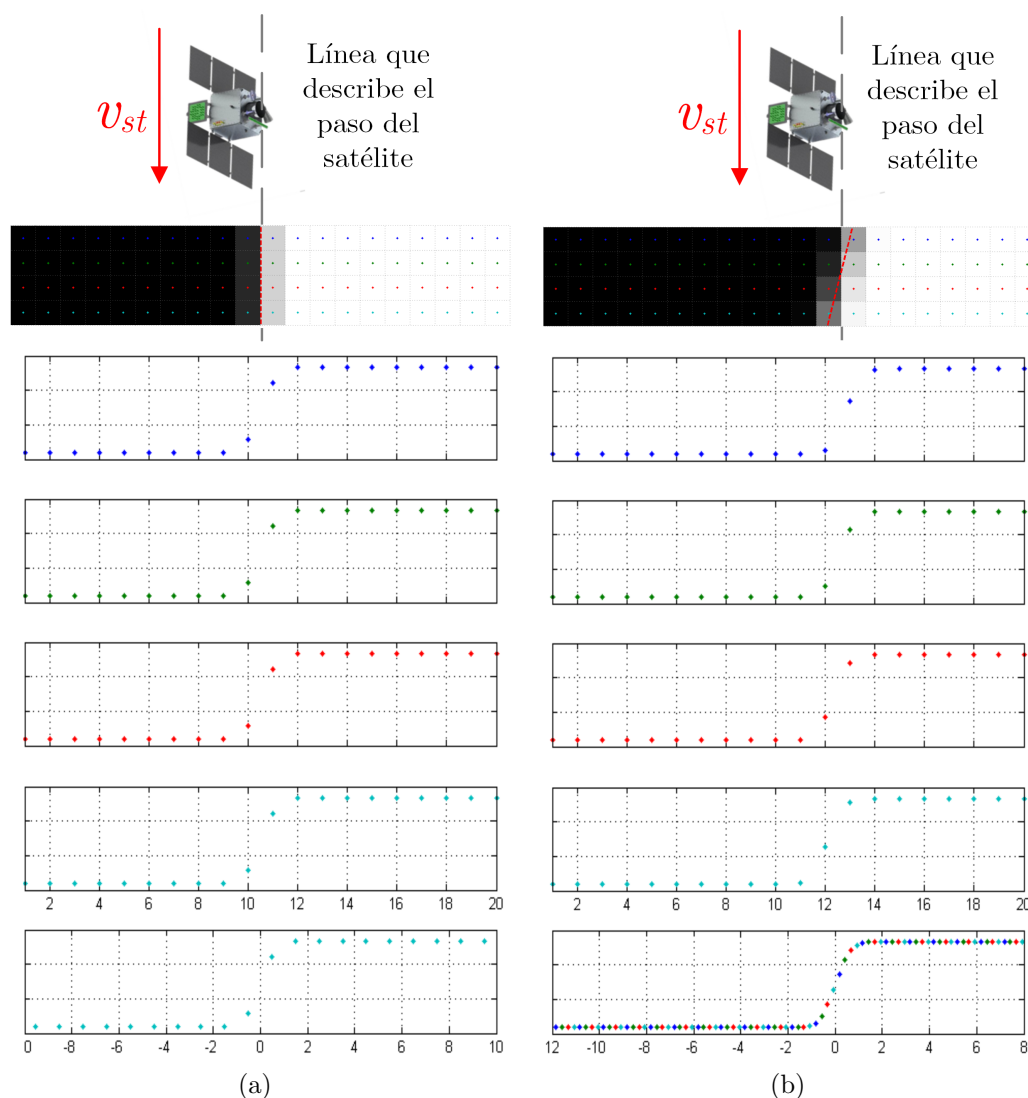


Fig. 3.9. Reconstrucción de la ESF a partir de respuestas al borde submuestreadas para (a) un borde objetivo no inclinado y (b) un borde objetivo inclinado respecto al paso de satélite.

La situación antes descrita se reproduce en la Fig. 3.10(a), y permite obtener la ESF en la dirección perpendicular al paso del satélite (*cross-track*). Si el objetivo de borde se dispone rotado 90° , como en la Fig. 3.10(b), se puede obtener la ESF a lo largo de la dirección del paso del satélite (*along-track*).

El método del borde objetivo ha sido ampliamente utilizado para obtener la MTF de sistemas de imagen de alta resolución aéreos [73], y satelitales como Ikonos-2 [72] y QuickBird [74]. En todos estos casos se utilizaron objetivos artificiales con ángulos de inclinación de 6° u 8° , ya que resultaron óptimos para la caracterización deseada. En líneas generales el procesamiento empleado implica: determinar la posición exacta del borde, alinear la respuesta de cada fila de la imagen respecto a la posición del borde, ajustar una curva al conjunto de datos

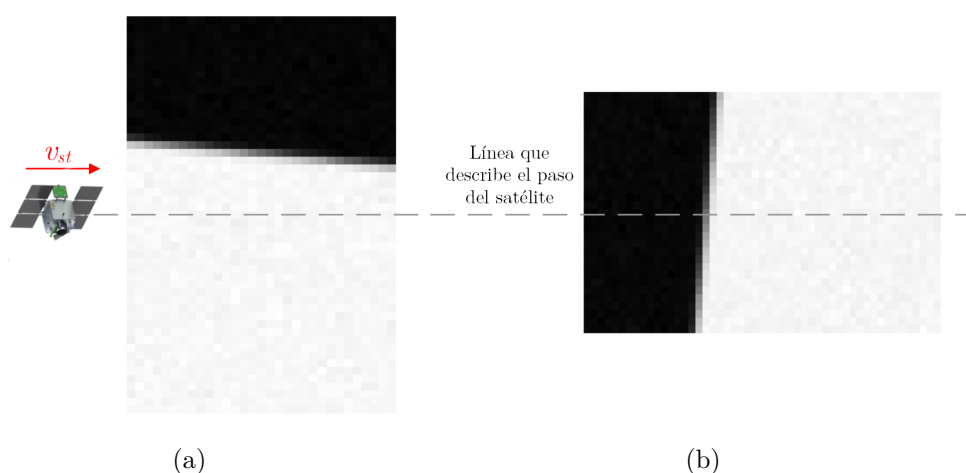


Fig. 3.10. *Objetivo de borde para caracterización (a) cross-track y (b) along-track.*

(a este punto se tiene la ESF) y por último derivar y hacer la DFT para obtener la MTF.

Para la caracterización de la MTF de instrumentos ópticos en laboratorio, la ISO publicó un estándar que contempla el uso de un borde objetivo inclinado 5° respecto a la vertical [81]. En [82] se propone una técnica para corregir las mediciones de MTF cuando los objetivos están inclinados ángulos distintos de 5° , que consiste en alterar la coordenada de la ESF de la forma

$$ESF(y) = ESF(\cos(\theta_{borde}).y) \quad (3.32)$$

donde θ_{borde} es el ángulo de inclinación del borde respecto a la vertical.

3.2.5. Aplicación del método del borde objetivo para estimación del NIIRS

El método del borde objetivo fue aplicado para la estimación del NIIRS por el autor T. Kim que publicó una serie de artículos donde se muestra el desempeño de un algoritmo de estimación de NIIRS basándose en objetivos naturales y objetivos artificiales [76–78]. El mismo propone que es posible estimar el NIIRS haciendo uso de objetivos artificiales u objetivos naturales con un error similar. La ventaja de la primera forma es que sólo se necesita un borde objetivo debidamente orientado, mientras que de la segunda manera es necesario usar varios objetivos naturales (debido a que éstos son irregulares) y obtener el NIIRS promedio. Los detalles sobre el algoritmo no se encontraron publicados, aunque de manera generar el procesamiento realizado para obtener el NIIRS a partir

de una imagen satelital con objetivos naturales consiste en: selección manual o autónoma de bordes objetivos, obtención de la ESF en cada borde, cálculo de RER y H en cada borde, cálculo de SNR en cada borde y finalmente cálculo del NIIRS a través de la GIQE. Los valores de GSD y G son obtenidos de la bibliografía o metadatos de la imagen. Adicionalmente, en el algoritmo, RER y H son estimados sin importar la dirección, es decir, para todos los objetivos se estima la RER, H y luego se obtiene el valor medio de cada parámetro. Los resultados muestran que para un conjunto de imágenes de Ikonos-2 y QuickBird el NIIRS se estima con un error menor a 0,3.

3.2.6. Degradación del NIIRS por compresión

En el artículo [83], se presenta un estudio que evalúa la degradación del NIIRS, indicado subjetivamente, en imágenes que son sometidas a compresión con pérdidas con sistemas basados en DCT, DWT y cuantificación vectorial para diferentes tasas de compresión. Dado un conjunto de imágenes, satelitales, aéreas y terrestres; éstas son comprimidas y luego descomprimidas. Para cada imagen, analistas de imagen indicaron el NIIRS antes y después de la compresión y por otro lado se calculó la degradación de la calidad utilizando métricas objetivas basadas en el error píxel a píxel. Los resultados mostraron que conforme se incrementa la tasa de compresión, el NIIRS disminuye para todos los sistemas de compresión evaluados, y que existe una buena correlación entre el MSE y el NIIRS para los sistemas de compresión basados en DCT y cuantificación vectorial.

En [80] se presenta un estudio que evalúa la degradación del NIIRS de imágenes satelitales RGB obtenidas de Google Earth debido a una compresión con pérdidas con los sistemas JPEG-Baselines y JPEG2000 a diferentes tasas de compresión. El NIIRS es estimado antes de comprimir la imagen y posterior a la descompresión de la misma mediante un algoritmo que utiliza bordes objetivos naturales. Aunque no se brindan detalles sobre el algoritmo para la estimación del NIIRS, la metodología difiere de las antes mencionadas. Este algoritmo aplica un detector de bordes Canny y transformada de Hough, calcula la SNR con una métrica similar a (3.5) y aplica la GIQE 3.0. Los resultados que se muestran finalmente no son concluyentes, el algoritmo de estimación de NIIRS no fue validado y existen errores en la interpretación de los parámetros de la GIQE.

Estimación del NIIRS de imágenes satelitales

En el capítulo anterior se presentó el NIIRS, la ecuación para su predicción y se introdujeron técnicas para la estimación del NIIRS en imágenes satelitales digitales. En el presente capítulo se muestra la aplicación de estos conceptos en el desarrollo de un algoritmo para la estimación del NIIRS de imágenes satelitales digitales. Dicho algoritmo se implementó en MATLAB y predice el NIIRS a través de la GIQE 4.0, donde los parámetros RER, H y el valor de SNR/G, son extraídos de objetivos de borde naturales presentes en la imagen y el GSD se obtiene de los metadatos de la misma.

La primera parte de este capítulo (Secciones 4.1 a 4.4) hace referencia a las alternativas de procesamiento evaluadas durante el desarrollo del algoritmo de estimación de RER, H, SNR y NIIRS, mientras que la segunda parte (Secciones 4.5 a 4.7) trata sobre la aplicación del algoritmo en la estimación del NIIRS de imágenes satelitales y la degradación del NIIRS debida a compresión. Durante el desarrollo del algoritmo, las distintas alternativas se evaluaron con imágenes simuladas que incluyeron efectos de degradación de respuesta en frecuencia por una PSF conocida y efectos de ruido en la cadena de detección conocidos. Esto permitió contrastar las estimaciones de los distintos métodos con el valor real. La metodología utilizada para la simulación de los bordes objetivos puede verse en el Anexo C.

4.1. Determinación de la posición del borde

Tal como se mencionó en el capítulo anterior, la estimación de RER y H inicia con la determinación de la posición del borde con una resolución subpixel y sigue con la reconstrucción de la ESF. Durante la revisión bibliográfica se

encontraron varios métodos para obtener la posición del borde. Algunos de ellos, basados en ajustar una curva a los niveles de gris de cada fila de la imagen para estimar la posición del borde en ésta y luego encontrar la recta de mínimos cuadrados que define el borde [70,71,75], y otro, basado en realizar una detección de bordes y luego una transformada de Hough [80]. Todos estos métodos junto a otros que se proponen en este trabajo fueron implementados y evaluados para seleccionar el mejor.

4.1.1. Método 1: Ajuste de función sigmoide

Dada una imagen de un borde objetivo como el de la Fig. 4.1, el procedimiento que emplea este método para obtener la posición del borde consiste en:

- 1) Derivación numérica de cada fila.
- 2) Estimación del nivel de la zona clara y de la zona oscura.
- 3) Ajuste de función sigmoide y obtención de la posición del flanco en cada fila.
- 4) Ajuste de recta de mínimos cuadrados.

4.1.1.1. Derivación numérica de cada fila

Ya que el flanco se encuentra en la posición de máxima pendiente, una primera aproximación a la posición del mismo puede hacerse encontrando el píxel de cada fila de la imagen donde la primera derivada toma su valor absoluto máximo.

Considerando que la imagen del borde objetivo de la Fig. 4.1 es de M filas por N columnas y está dada por $I[x, y]$, cada una de sus filas se representa por $I_x[y]$ donde $x = \{0, 1, \dots, M-1\}$. Entonces, la derivada de cada una de sus filas, dada por un diferenciador de primer orden, es

$$d_x[y] = I_x[y] - I_x[y+1]; \quad \text{para } x = \{0, 1, \dots, M-1\}, \quad (4.1)$$

y la posición del píxel de máxima pendiente queda dada por el valor de y donde $d_x[y]$ toma el valor máximo en cada fila, es decir,

$$m_x \Rightarrow d_x[y = m_x] = \max_{y=\{0,1,\dots,N-1\}} [d_x[y]]; \quad \text{para } x = \{0, 1, \dots, M-1\}. \quad (4.2)$$

En la Fig. 4.1 los puntos rojos indican el píxel donde se registró la máxima pendiente en cada fila y en la Fig. 4.2 se ve la respuesta al borde y su respectiva derivada para las filas 2 y 23 donde se indican también los puntos de máxima pendiente, m_x .

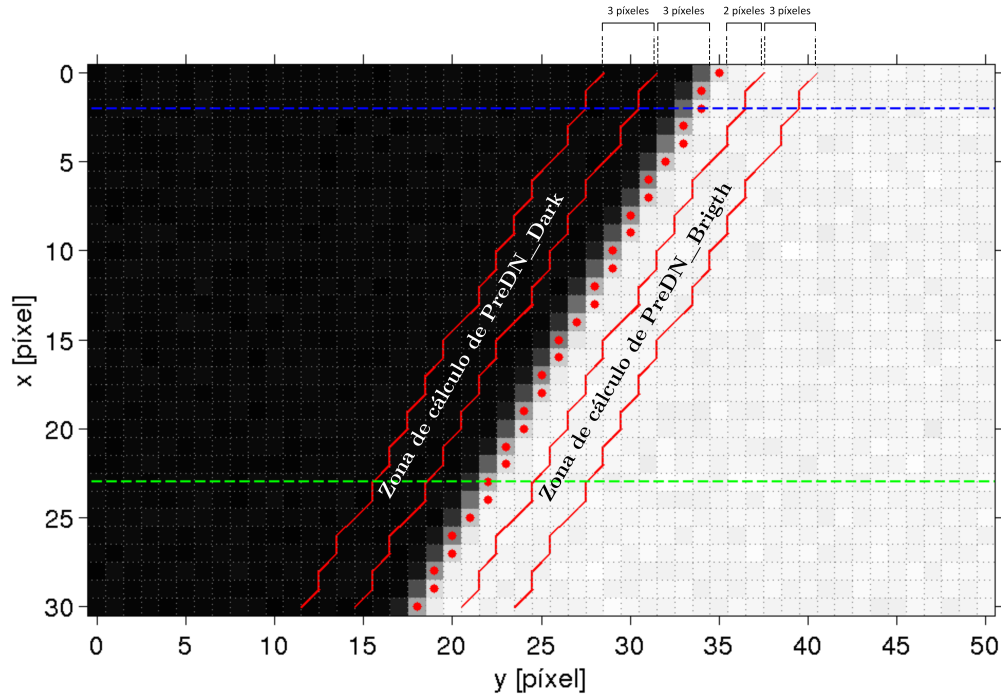


Fig. 4.1. Borde objetivo y zonas de cálculo de $preDN_{bright}$ y $preDN_{Dark}$.

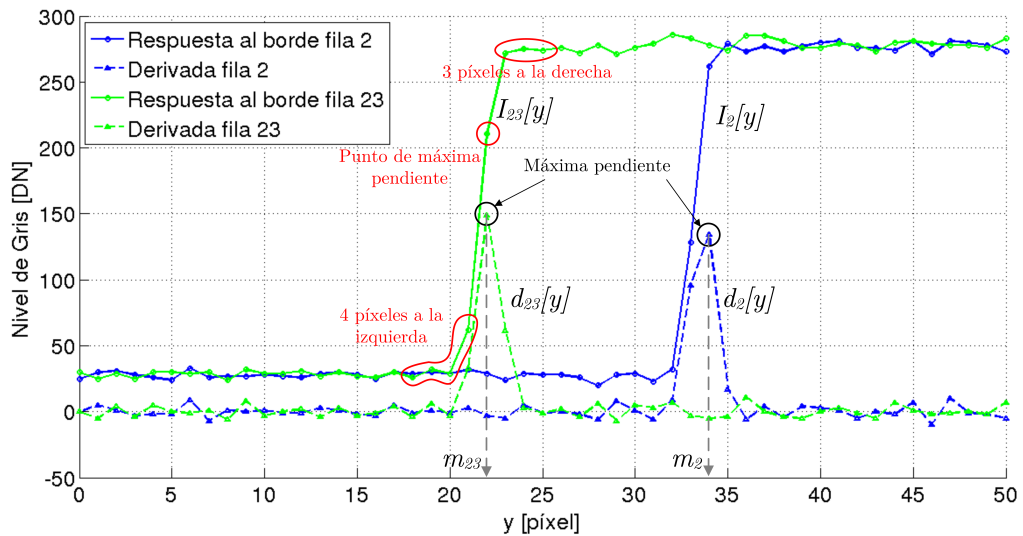


Fig. 4.2. Respuesta al borde de dos líneas y sus respectivas derivadas.

4.1.1.2. Estimación de el nivel de la zona clara y de la zona oscura

En esta etapa se desea estimar el nivel de gris promedio de la zona clara y de la zona oscura, denominados $preDN_{bright}$ y $preDN_{Dark}$ respectivamente.

En primer lugar se definen dos zonas para el cálculo de cada parámetro. Si se observa la Fig. 4.1 y en cada fila se considera el punto encontrado por derivación (puntos rojos) la posición $y = 0$, la zona de cálculo de *preDN_bright* está definida por los píxeles 3 a 5 de cada fila y la zona de cálculo de *preDN_Dark* está definida los píxeles -6 a -4 de cada fila. Luego se calcula el valor medio de cada zona para estimar *preDN_bright* y *preDN_Dark*. Adicionalmente se estima el nivel de gris que corresponde a la posición del flanco, denominado *preDN_Edge*, como el valor medio de *preDN_bright* y *preDN_Dark*.

4.1.1.3. Ajuste de función sigmoide y estimación de la posición del flanco en cada fila

En este paso, para cada fila se toman las 8 muestras cercanas al punto de máxima pendiente,

$$I'_x[y] = I_x[y]; \quad \text{para } y = \{m_x-4, m_x-3, \dots, m_x, \dots, m_x+3\}, \quad (4.3)$$

como se ilustra en la Fig. 4.2, y se ajusta una función sigmoide [70, 71],

$$f_x(y) = \frac{a_x}{e^{\left(\frac{y-b_x}{c_x}\right)} + 1} + d_x, \quad (4.4)$$

donde:

- a_x es la diferencia entre el nivel máximo y el mínimo;
- b_x es el valor de y donde $f_x(y)$ toma el valor medio entre su máximo y mínimo, por lo tanto, determina la posición del flanco;
- c_x es el valor de y donde la función llega al 73,1% de su crecimiento, por lo tanto es un indicador de la pendiente de la curva en su zona central, a menor valor de $|c|$ mayor es la pendiente, para flanco ascendente $c_x < 0$ y para flanco descendente $c_x > 0$;
- d_x : el valor mínimo que toma la función.

Entonces, la posición de flanco, en cada fila, P'_x , con una precisión subpíxel queda determinada por

$$P'_x = b_x; \quad \text{para } x = \{0, 1, \dots, N-1\}. \quad (4.5)$$

Para facilitar el ajuste se inicializaron los parámetros a los siguientes valores:

- $a_x = \text{preDN_bright} - \text{preDN_Dark}$;

- $b_x = m_x$, posición del flanco encontrado por derivación para cada fila;
- $c_x = -0,15$ corresponde a pendiente positiva y
- $d_x = preDN_Dark$.

Dada una imagen de un borde objetivo, a modo de ejemplo en la Fig. 4.3 se muestran los resultados del ajuste de la función sigmoide para 3 filas de la imagen para los casos a los que se aplicó y no se aplicó la MTF. Puede apreciarse que cuando la imagen presenta un rebasamiento de borde la función sigmoide no ajusta correctamente.

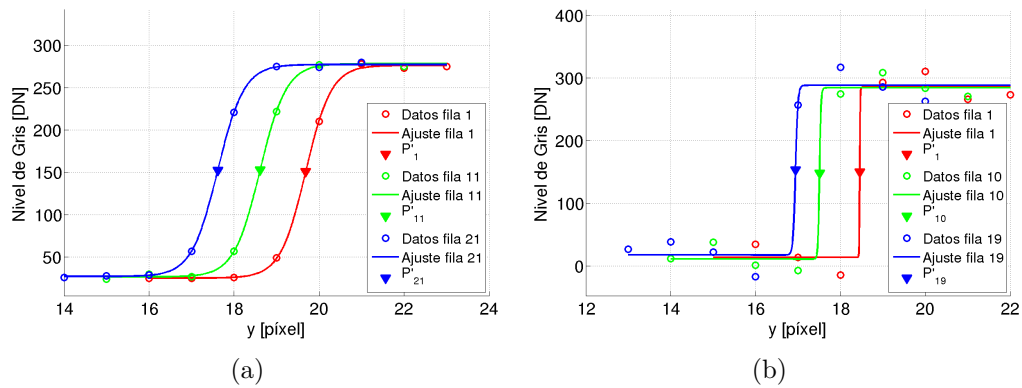


Fig. 4.3. Estimación de la posición del borde en cada fila por el método 1 (ajuste de función sigmoide) para imagen con borde inclinado 6° . Ajuste para 3 filas para (a) imagen sin MTFC aplicada y (b) imagen con MTFC aplicada.

4.1.1.4. Ajuste de recta de mínimos cuadrados

Para cada fila, $x = \{0, 1, \dots, M-1\}$, se conoce la posición del flanco con una precisión subpíxel, $P' = \{P'_0, P'_1, \dots, P'_{M-1}\}$, puntos azul en la Fig. 4.4. No obstante, debido al ruido en la imagen y posibles errores en el ajuste, este grupo de puntos podría no definir perfectamente la posición real del borde. Por lo tanto, se busca la recta de mínimos cuadrados que se ajusta a al conjunto de puntos $\{x; P'\}$,

$$y(x) = a \cdot x + b, \quad (4.6)$$

mostrada en verde en la Fig. 4.4. Finalmente la la posición del borde en cada fila queda determinada por

$$P_x = y(x); \quad \text{para } x = \{0, 1, \dots, M-1\}. \quad (4.7)$$

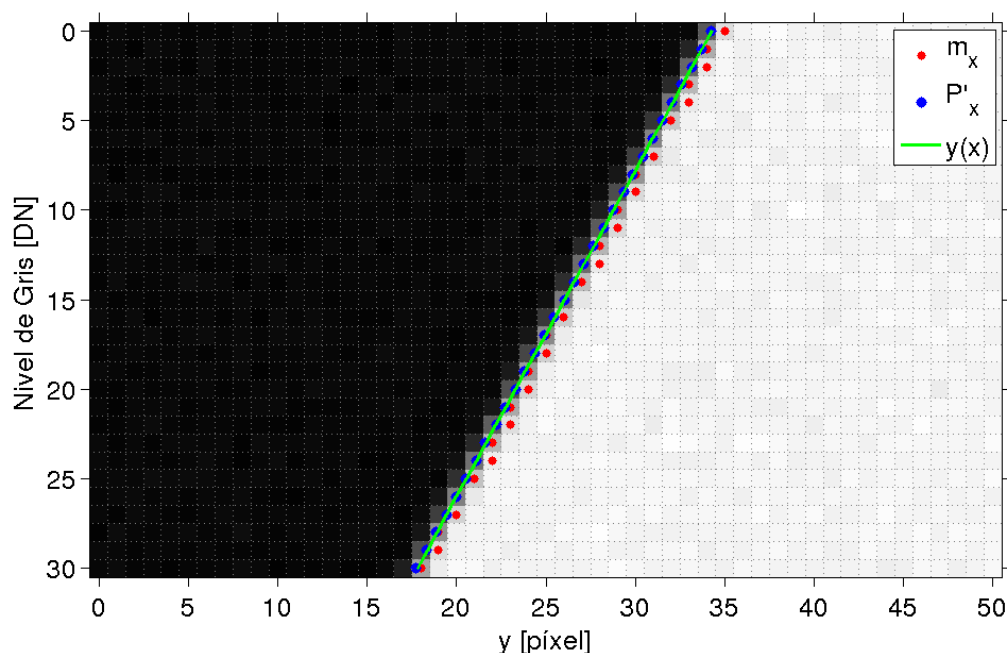


Fig. 4.4. Posición del flanco por derivación numérica, m'_x , por resultado de ajuste en cada fila, P'_x , y recta de mínimos cuadrados, $y(x)$.

4.1.2. Método 2: Ajuste de función cúbica

Dada una imagen de un borde objetivo como el de la Fig. 4.1 el procedimiento que emplea este método para obtener la posición del borde consiste en:

- 1) Derivación numérica de cada fila.
- 2) Ajuste de función cúbica y obtención de la posición del flanco en cada fila.
- 3) Ajuste de recta de mínimos cuadrados.

El primer paso se realiza acorde a lo descrito en la Sección 4.1.1.1. Luego, se ajusta a los 8 puntos cercanos al borde un polinomio cúbico para cada fila, $f_x(y)$, y la posición del flanco en cada fila, con una precisión subpíxel, se estima como el valor de y para el cual la pendiente de $f_x(y)$ es máxima [75]. A modo de ejemplo en la Fig. 4.5 se muestran los resultados del ajuste de la función cúbica para 3 filas de la imagen para los casos a los que se aplicó y no se aplicó la MTFC, en ambos casos se aprecia que la función ajusta correctamente. Finalmente se ajusta la recta de mínimos cuadrados que define la posición de borde acorde a lo mencionado en la Sección 4.1.1.4.

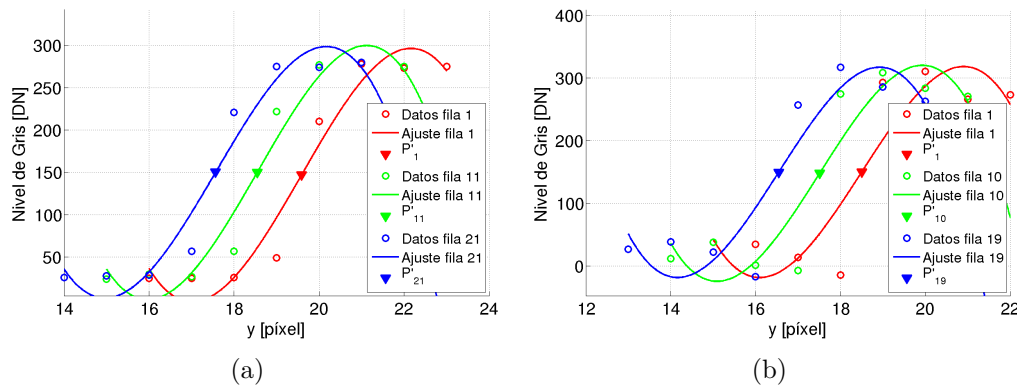


Fig. 4.5. Estimación de la posición del borde en cada fila por el método 2 (ajuste de función cúbica) para imagen con borde inclinado 6° . Ajuste para 3 filas para (a) imagen sin MTFC aplicada y (b) imagen con MTFC aplicada.

4.1.3. Método 3: Ajuste de función spline + `preDN_Edge`

Dada una imagen de un borde objetivo como el de la Fig. 4.1 el procedimiento que emplea este método para obtener la posición del borde consiste en:

- 1) Derivación numérica de cada fila.
- 2) Estimación del nivel de gris en la posición del flanco, `preDN_Edge`.
- 3) Ajuste de función spline y obtención de la posición del flanco en cada fila.
- 4) Estimación de la posición del borde por ajuste de recta de mínimos cuadrados.

El primer y segundo paso se realizan acorde a lo descrito en la Sección 4.1.1.1 y Sección 4.1.1.2 respectivamente. Luego se ajusta a los 8 puntos cercanos al borde una función spline (sumatoria de polinomios cúbicos) y se estima la posición del flanco como el valor de y que hace que $f_x(y) = \text{preDN_Edge}$. A modo de ejemplo en la Fig. 4.6 se muestran los resultados del ajuste de la función spline para 3 filas de la imagen para los casos a los que se aplicó y no se aplicó la MTFC, en ambos casos se aprecia que la función pasa por todos los puntos de datos y ajusta correctamente. Finalmente se ajusta la recta de mínimos cuadrados que define la posición del borde acorde a lo mencionado en la Sección 4.1.1.4.

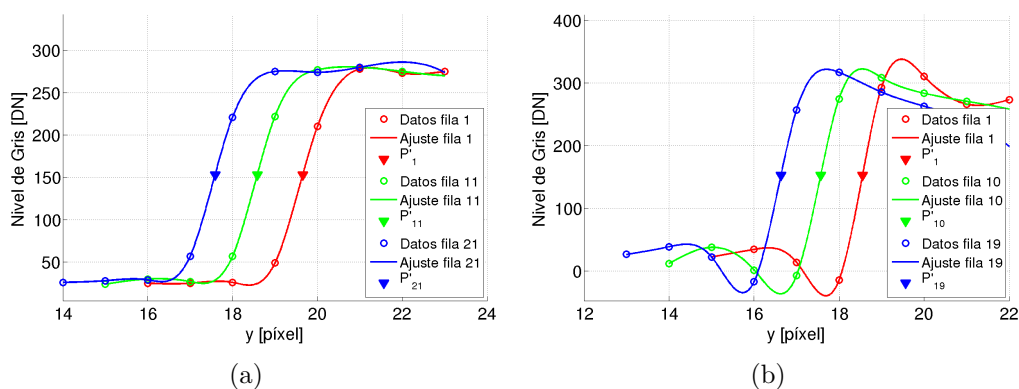


Fig. 4.6. Estimación de la posición del borde en cada fila por el método 3 (ajuste de función spline + *preDN_Edge*) para imagen con borde inclinado 6° . Ajuste para 3 filas para (a) imagen sin MTFC aplicada y (b) imagen con MTFC aplicada.

4.1.4. Método 4: Ajuste de función spline + máxima pendiente

De la misma manera que en el método anterior se utiliza como curva de ajuste una función spline y se estima la posición del flanco en cada fila, con una precisión subpíxel, como la el valor de y donde la derivada de $f_x(y)$ toma su valor máximo. No es necesario estimar *preDN_Edge*. Finalmente se ajusta la recta de mínimos cuadrados que define la posición de borde acorde a lo mencionado en la Sección 4.1.1.4.

4.1.5. Método 5: Detector de bordes Canny + transformada de Hough

Este método consiste en detectar líneas en una imagen utilizando un *detector de bordes Canny* y luego la *transformada de Hough* [14]. Dada una imagen de un borde objetivo como el de la Fig. 4.1 en primer lugar se aplica el detector de bordes *Canny* para obtener una imagen binaria que indica los píxeles donde existen bordes y luego se hace una transformación de Hough para obtener la posición de los bordes que presentan una forma recta [80]. Esta recta define la posición del borde de manera análoga a la recta de mínimos cuadrados de los cuatro métodos antes descritos. Dada una imagen de un borde objetivo, en la Fig. 4.7 se muestran los resultados de la estimación de la posición de borde para los casos a los que se aplicó y no se aplicó la MTFC, donde puede verse la posición real del flanco y la posición estimada. En ambos casos se observa que la estimación difiere aproximadamente medio píxel de la posición real.

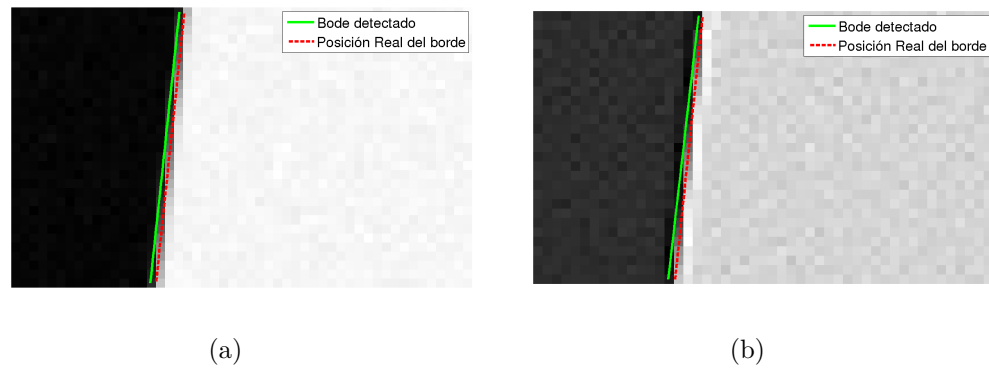


Fig. 4.7. Estimación de la posición del borde por el método 5 (detector de bordes Canny + transformada de Hough) para imagen con borde inclinado 6° . (a) Posición del flanco para (a) imagen sin MTFC aplicada y (b) con MTFC aplicada.

4.1.6. Comparación y selección

Ya presentados los cinco métodos para la determinación de la posición del borde con una resolución subpíxel, se procede a compararlos y seleccionar el que tenga el mejor desempeño. Los métodos fueron evaluados con imágenes simuladas de bordes objetivos de ángulos de inclinación que van de 1° a 45° con pasos de 1° . Además se evaluaron dos casos, imágenes a las que se aplicó MTFC e imágenes a las que no se aplicó MTFC, con el objeto de cubrir ambas situaciones ya que no se conocen con exactitud las características de la corrección aplicadas a las imágenes satelitales sobre las que se estima el NIIRS. Las imágenes simuladas se generaron con el con al algoritmo del Anexo C. Como métrica de comparación se utilizó el MSE entre la posición estimada del borde en cada fila y la posición real de borde en cada fila (obtenida durante la generación del borde objetivo ideal) y el tiempo de ejecución del algoritmo.

En la Fig. 4.8 se muestra el MSE entre la posición estimada y la posición real del borde para cada valor de ángulo ensayado y en la la Tabla 4.1 se muestran los valores promedio del MSE y el tiempo de ejecución para cada método. El método basado en el detector de bordes Canny y la transformada de Hough es el más rápido pero arroja grandes errores que quedan fuera del los gráficos mientras que los los cuatro métodos restantes presentan errores similares. Puede verse que cuando no se aplica la MTFC, el método basado en la función sigmoide presenta el menor error, aunque cuando se tiene una imagen con rebasamiento de borde, debido a las características de la función, ésta no se ajusta bien a los datos, y el error se vuelve mayor. Entre los tres métodos restantes, el método 3 arrojó

el menor error y además es el que requiere el menor tiempo. Por lo tanto se seleccionó el método 3 para la determinación de la posición del borde.

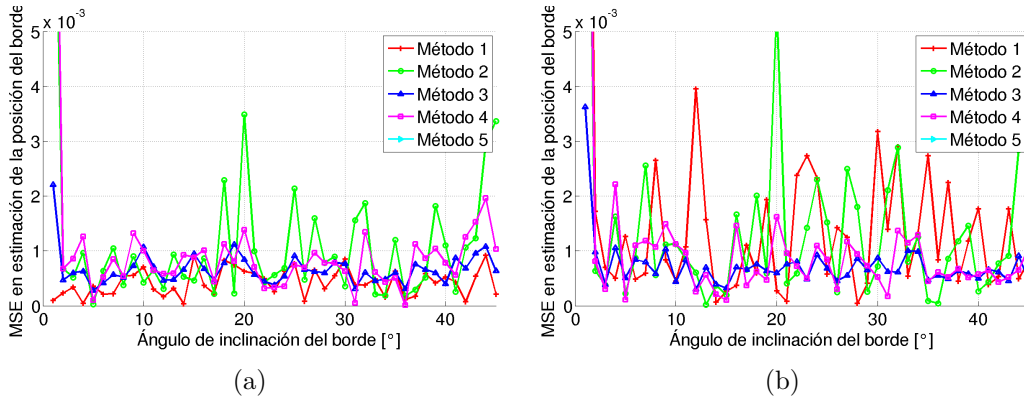


Fig. 4.8. Comparación de MSE en la estimación de la posición del flanco para los 5 métodos en función del ángulo de inclinación del borde, para (a) imágenes con MTFC aplicada y (b) sin MTFC aplicada.

Tabla 4.1 – Error y tiempos de ejecución promedio de cada método de estimación de la posición de borde.

	MTFC off		MTFC on	
	MSE promedio	Tiempo promedio [s]	MSE promedio	Tiempo promedio [s]
Método 1	0,0004	5,56	0,0016	8,68
Método 2	0,0012	0,33	0,0015	0,32
Método 3	0,0007	0,26	0,0007	0,27
Método 4	0,0011	0,80	0,0011	0,79
Método 5	0,4513	0,02	0,5320	0,02

4.2. Reconstrucción de la ESF

El segundo paso para la obtención de la ESF es la reconstrucción de la misma a partir de las respuestas al borde submuestreadas de cada fila. Para cada fila de la imagen se tiene la respuesta al borde submuestreada I_x , donde cada muestra está asociada a una coordenada y , es decir

$$\left\{ \begin{array}{l} I_x = \{I_x[0], I_x[1], \dots, I_x[N-1]\} \\ y = \{0, 1, \dots, N-1\} \end{array} \right\} \text{ para } x = \{0, 1, \dots, M-1\}. \quad (4.8)$$

Además, se conoce la la posición del flanco para cada fila, P_x . La reconstrucción de la ESF a partir de estos datos ocurre en tres etapas:

- 1) Alineación de las respuestas al borde submuestreadas de cada fila.
- 2) Corrección debido a la inclinación del borde.
- 3) Remuestreo y suavizado de la respuesta al borde.

A continuación se explica cada una de éstas.

4.2.1. Alineación de las respuestas al borde submuestreadas de cada fila

Dada la ecuación (4.8) que representa la respuesta al borde submuestreada para cada fila, si se grafica I_x respecto a y se obtiene la Fig. 4.9(a). Para obtener la respuesta al borde del sistema es necesario alinear estas respuestas, para ésto se hace coincidir la posición del flanco de todas las filas con la coordenada $y = 0$, entonces las coordenadas para cada fila, y_x , quedan dadas por

$$y_x = y - P_x = \{-P_x, 1-P_x, \dots, N-1-P_x\}, \quad (4.9)$$

y cada respuesta al borde queda definida por

$$\left\{ \begin{array}{l} I_x = \{I_x[0], I_x[1], \dots, I_x[N-1]\} \\ y_x = \{-P_x, 1-P_x, \dots, N-1-P_x\} \end{array} \right\} \quad \text{para } x = \{0, 1, \dots, M-1\}. \quad (4.10)$$

Graficando este último conjunto se obtiene la Fig. 4.9(b). Debido a que cuando se usan bordes naturales la extensión de las zonas uniformes es limitada, se acota la respuesta al borde al intervalo $[-5, 5]$, por lo tanto se eliminan todas las muestras con $|y_x| > 5$ lo que implica conservar $L = 11 \times M$ muestras de I_x con su respectiva coordenada y_x que se agrupan en las variables ER e y' respectivamente,

$$\left\{ \begin{array}{l} ER = \{ER[0], ER[1], \dots, ER[L-1]\} \\ y' = \{y'_1, y'_2, \dots, y'_{L-1}\} \end{array} \right\}. \quad (4.11)$$

4.2.2. Corrección debido a la inclinación del borde

El ángulo de inclinación del flanco tiene efectos sobre la distribución de los puntos y por lo tanto sobre las mediciones que se obtendrán del parámetro

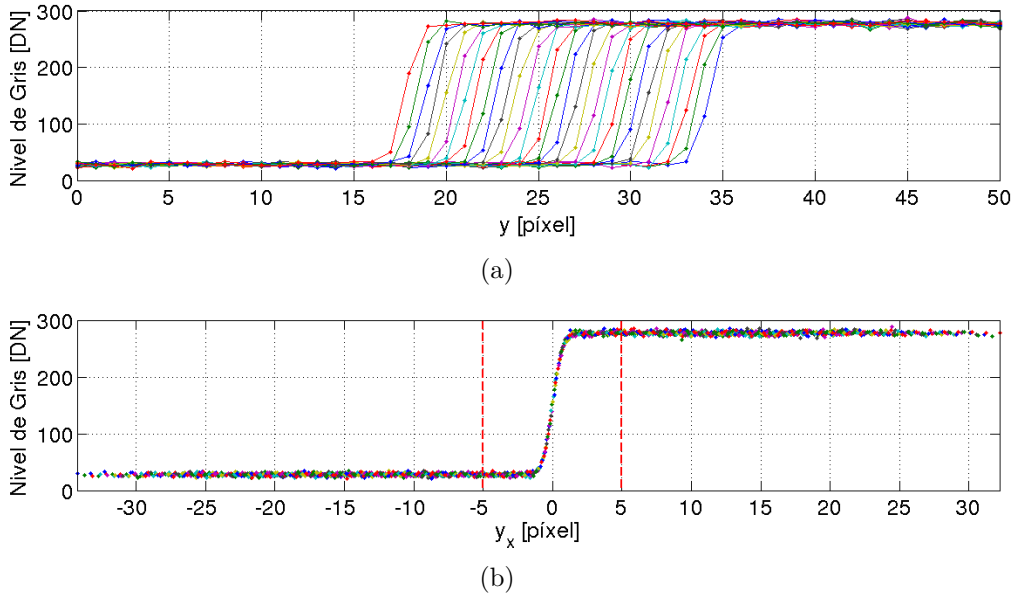


Fig. 4.9. Alineación de respuestas al borde submuestreadas de cada fila para una imagen con un borde inclinado 6° y sin MTFC aplicada. En (a) respuestas al borde de cada fila sin alinear y en (b) respuestas al borde alineadas.

de caracterización de la frecuencia espacial que se desea estimar. Como en este trabajo se desea obtener la respuesta al borde usando objetivos naturales, no se pudo asegurar que éstos mantengan un determinado ángulo. Por tal motivo se aplica una corrección por inclinación del borde de la forma (3.32). Conocido el ángulo de inclinación del borde, $\theta_{borde} = \arctan(a)$, donde a es la pendiente de la recta de mínimos cuadrados que determina la posición del borde, definida en (4.6), la corrección implica multiplicar por el factor $c = \cos[\arctan(a)]$ la coordenada y' del conjunto (4.11), entonces la respuesta al borde corregida por inclinación queda dada por

$$\left\{ \begin{array}{l} ER = \{ER[0], ER[1], \dots, ER[L-1]\} \\ y'' = \{c \cdot y'_1, c \cdot y'_2, \dots, c \cdot y'_{L-1}\} \end{array} \right\}. \quad (4.12)$$

En la Fig. 4.10 puede verse el valor absoluto del error en la estimación de RER y H sin corrección por inclinación y con la corrección. Estos resultados son obtenidos con el método que se termina por consolidar en este capítulo. Al momento se desea mostrar que la corrección produce grandes mejoras tanto en la estimación de RER como de H conforme el ángulo de inclinación aumenta.

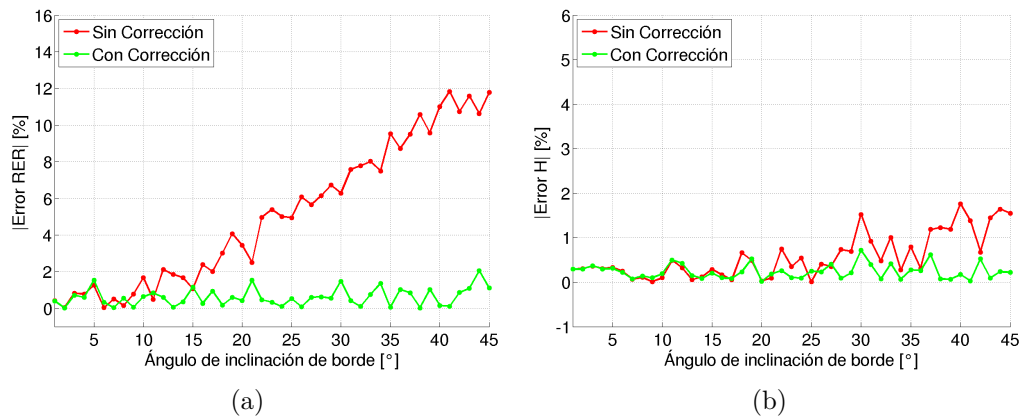


Fig. 4.10. Corrección de ESF por ángulo de inclinación para objetivo sin MTF aplicada. (a) Valor absoluto del error en estimación de RER y (b) valor absoluto del error en estimación de H.

4.2.3. Remuestreo y suavizado de la respuesta al borde

Cada una de las filas tiene un conjunto de putos de la respuesta al borde separados 1 píxel, gracias al desfase que existe entre cada línea, luego de la alineación, se obtiene una respuesta al borde con una mayor tasa de muestreo. No obstante, las muestras no están espaciadas uniformemente, debido a que el paso entre muestras depende del ángulo de inclinación del borde y la cantidad de líneas que se tengan en la imagen, y no se puede asegurar que se conozca el valor de la respuesta al borde en las posiciones necesarias para la medición de RER y H. Por lo tanto es necesario remuestrear la respuesta al borde (4.12) para obtener la ESF con un intervalo de muestreo constante. Además, la señal de la respuesta al borde está afectada por el ruido de la imagen por lo tanto es necesario suavizar los datos. Para el remuestreo y suavizado de los datos se evaluaron dos métodos que se analizan a continuación.

Previo a hacer el remuestreo es necesario conocer los niveles en la zona clara DN_Bright y la zona oscura DN_Dark , ahora que se sabe con mejor exactitud el centro del borde es posible estimar mejor los mismos. Para estimar el valor DN_Dark se toma la media de los puntos entre el píxel -5 y el píxel -3 y para el valor DN_Bright se toma la media entre el píxel 3 y el píxel 5 .

4.2.3.1. Método 1: ajuste de spline suavizada

En [75] se ajusta una función spline al conjunto de puntos, sin embargo, debido a la presencia de ruido, el resultado del ajuste puede causar una respuesta más ruidosa que la original, como se ilustra en la Fig. 4.11. En su lugar, en esta sección se propone ajustar una función spline suavizada.

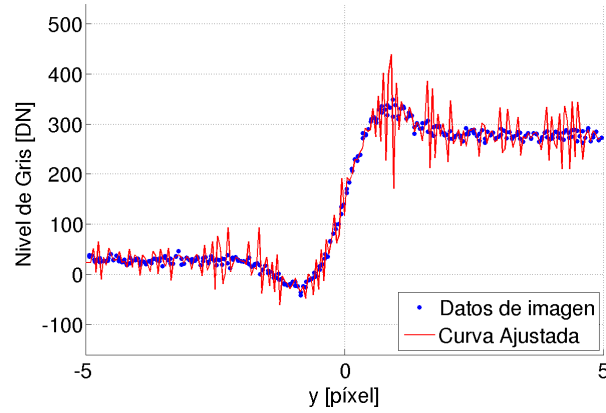


Fig. 4.11. Ajuste de spline.

Dado el conjunto de puntos (4.12) que definen la respuesta al borde corregida por inclinación, se ajusta la función spline suavizada, $f_{S_Spline}(y)$. Luego ésta se muestrea desde -5 hasta $+5$ en intervalos de $0,05$ píxel y finalmente se normaliza respecto a los valores DN_Bright y DN_Dark para obtener la ESF, es decir,

$$ESF[y_m] = \frac{f_{S_Spline}(y_m) - DN_Dark}{DN_Bright - DN_Dark}, \quad (4.13)$$

donde $y_m = 0,05 \cdot m$, para $m = \{-100, -99, \dots, -1, 0, 1, \dots, 99, 100\}$.

En la Fig. 4.12(a) puede verse el conjunto de puntos de la respuesta al borde y la función spline ajustada y en la Fig. 4.12(b) la respuesta al borde reconstruida y la respuesta al borde real para un borde objetivo al que no se aplicó MTFC. En la Fig. 4.13 se muestra lo mismo para el caso de una imagen a la que se ha aplicado una MTFC.

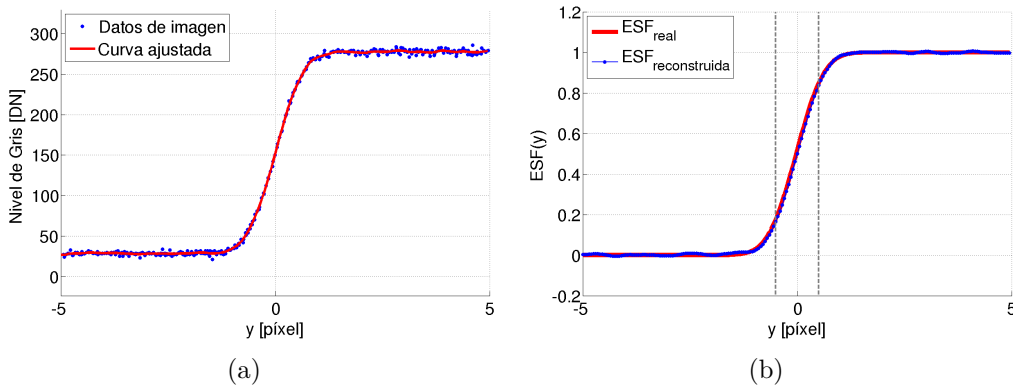


Fig. 4.12. Reconstrucción de ESF por método 1 (ajuste de spline suavizada) para imagen con un borde inclinado 6° y sin MTFC aplicada. En (a) ajuste de función spline suavizada y en (b) comparación de ESF reconstruida y ESF real.

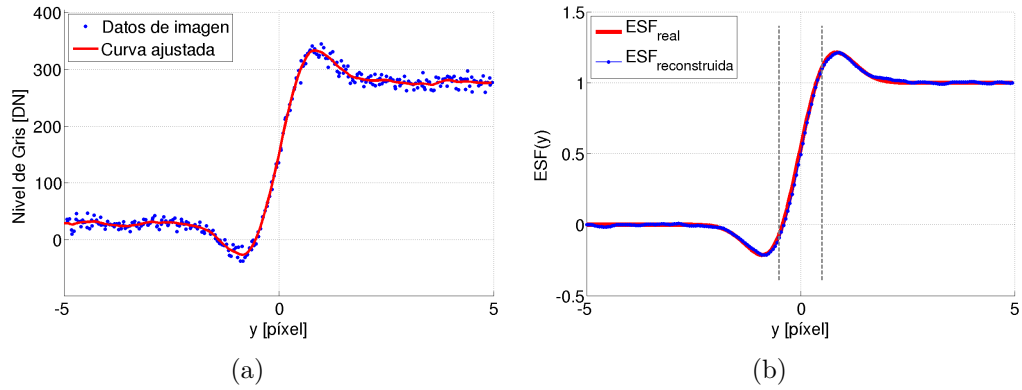


Fig. 4.13. Reconstrucción de ESF por método 1 (ajuste de spline suavizada) para imagen con un borde inclinado 6° y con MTFC aplicada. En (a) ajuste de función spline suavizada y en (b) comparación de ESF reconstruida y ESF real.

4.2.3.2. Método 2: ajuste de función que modela un borde con sobrepicos

En [75] se ajusta una función sigmoide al conjunto de puntos, sin embargo, en presencia de sobre picos, dicha función no ajusta bien, como se mostró en la Sección 4.1.1.3. En su lugar, en esta sección se propone ajustar una función que modela un borde con sobrepicos.

La ESF, para una imagen a la que no se le aplicó MTFC, no posee sobrepicos y puede modelarse con la función sigmoide

$$f(y) = \frac{a}{e^{\left(\frac{y-b}{c}\right)} + 1} + d. \quad (4.14)$$

Por otro lado, se tiene la respuesta al impulso en una dirección de la MTFC utilizada para la simulación de los bordes objetivos que está dada por la ecuación

$$\mathcal{F}^{-1}\{MTFC(0, v)\} = k_1 \cdot \delta(y) + k_2 \cdot \delta(y - 1) + k_2 \cdot \delta(y + 1), \quad (4.15)$$

desarrollada en el Anexo D. La respuesta al borde incluyendo la MTFC, es la convolución entre (4.14) y (4.15). Resolviendo la convolución

$$\begin{aligned} f_2(y) &= f(y) * \mathcal{F}^{-1}\{MTFC(0, v)\} \\ &= f(y) * k_1 \cdot \delta(y) + f(y) * k_2 \cdot \delta(y - 1) + f(y) * k_2 \cdot \delta(y + 1) \\ &= k_1 \cdot f(y) + k_2 \cdot f(y - 1) + k_2 \cdot f(y + 1) \end{aligned} \quad (4.16)$$

se obtiene una nueva función que se ajusta a datos de una respuesta al borde con sobrepicos, dada por

$$f_{S_Simoide}(y) = \frac{a_1}{e^{\frac{(y-b)}{c}} + 1} + \frac{a_2}{e^{\frac{(y-1-b)}{c}} + 1} + \frac{a_2}{e^{\frac{(y+1-b)}{c}} + 1} + D. \quad (4.17)$$

donde, $a_1 = a.k_1$, $a_2 = a.k_2$ y $D = d.(k_1 + 2.k_2)$. Para facilitar el ajuste se inicializaron los parámetros a los siguientes valores:

- $a_1 = 0,75.(DN_bright - DN_Dark)$;
- $a_2 = 0,125.a_1$;
- $b = 0$;
- $c = -0,15$ y
- $D = DN_Dark$.

Dado el conjunto de puntos (4.12) que definen la respuesta al borde corregida por inclinación, se ajusta la función (4.17). Luego ésta se muestrea desde -5 hasta $+5$ en intervalos de $0,05$ píxel y finalmente se normaliza respecto a los valores DN_Bright y DN_Dark para obtener la ESF, es decir,

$$ESF[y_m] = \frac{f_{S_Simoide}(y_m) - DN_Bright}{DN_Bright - DN_Dark}, \quad (4.18)$$

donde $y_m = 0,05 \cdot m$, para $m = \{-100, -99, \dots -1, 0, 1, \dots 99, 100\}$.

En la Fig. 4.14(a) puede verse el conjunto de puntos de la respuesta al borde y la función ajustada y en la Fig. 4.14(b) la respuesta al borde reconstruida y la respuesta al borde real para un borde objetivo al que no se aplicó MTFC. En la Fig. 4.15 se muestra lo mismo para el caso de una imagen a la que se ha aplicado una MTFC.

4.2.3.3. Comparación y selección

Ya presentados los dos métodos para la reconstrucción de la ESF, se procede a compararlos y seleccionar el que tiene el mejor desempeño. Los métodos fueron evaluados con imágenes simuladas de bordes objetivos con ángulos de inclinación que van de 1° a 45° con pasos de 1° . Además se evalúan dos casos, imágenes a las que se aplicó MTFC e imágenes a las que no se aplicó MTFC. Como métrica de comparación se utilizó el error entre RER y H obtenidos de la ESF reconstruida y RER y H obtenidos de la ESF conocida.

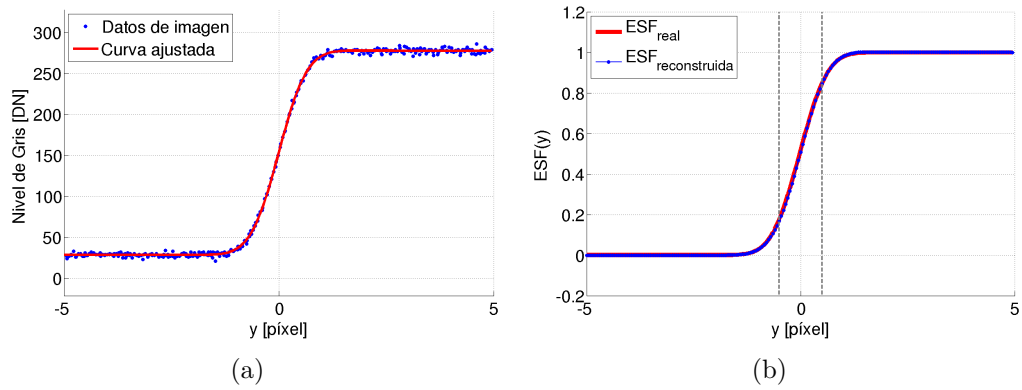


Fig. 4.14. Reconstrucción de ESF por método 2 (ajuste de función que modela un borde con sobrepicos) para imagen con un borde inclinado 6° y sin $MTFC$ aplicada. En (a) ajuste de función spline suavizada y en (b) comparación de ESF reconstruida y ESF real.

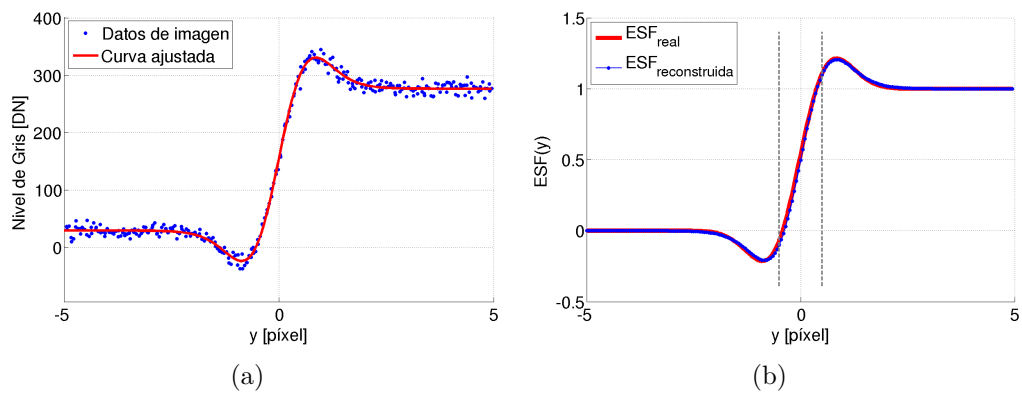


Fig. 4.15. Reconstrucción de ESF por método 2 (ajuste de función que modela un borde con sobrepicos) para imagen con un borde inclinado 6° y con $MTFC$ aplicada. En (a) ajuste de función spline suavizada y en (b) comparación de ESF reconstruida y ESF real.

En las Figs. 4.16 y 4.17, se muestra el error de estimación de RER y H de los dos métodos para bordes objetivos con un ángulo de inclinación desde 1° a 45° , para imágenes sin aplicación de $MTFC$ y con su aplicación respectivamente. En ambos casos el error de estimación de RER es, en general, menor para el método 1 y H se estima con un error menor en el método 2. El método 1, además de presentar el menor error en estimación de RER, es el que requiere menor tiempo, como se puede ver en la Tabla 4.2.

Observando la GIQE (3.23), se puede ver que el parámetro RER tiene un impacto mucho mayor respecto a H en el cálculo del NIIRS. Por lo tanto, el método 1 presenta mejores características para la estimación del NIIRS. Este criterio se comprueba nuevamente en la Sección 4.4 donde se muestran los resultados para la estimación del NIIRS por los dos métodos usando imágenes

simuladas y en la Sección 4.6 donde se ensaya con imágenes satelitales de NIIRS conocido.

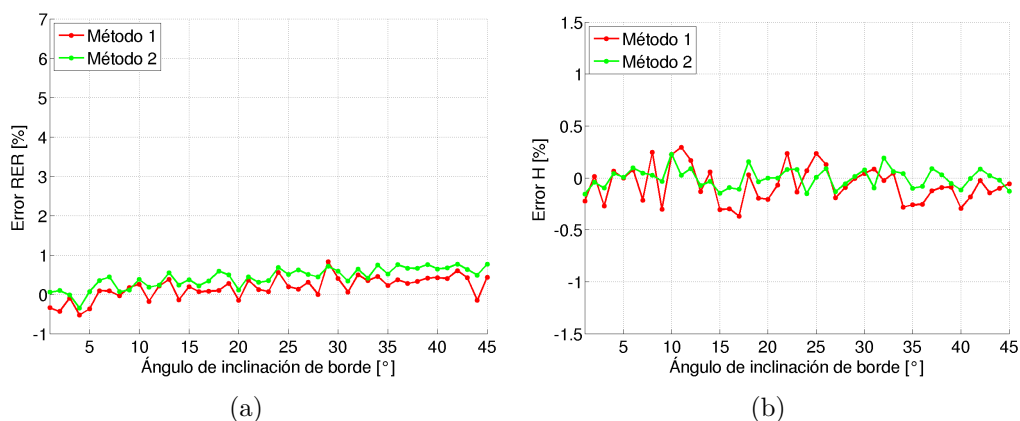


Fig. 4.16. Error de estimación de (a) RER y (b) H, para imágenes sin MTFC aplicada, en función del ángulo de inclinación de borde.

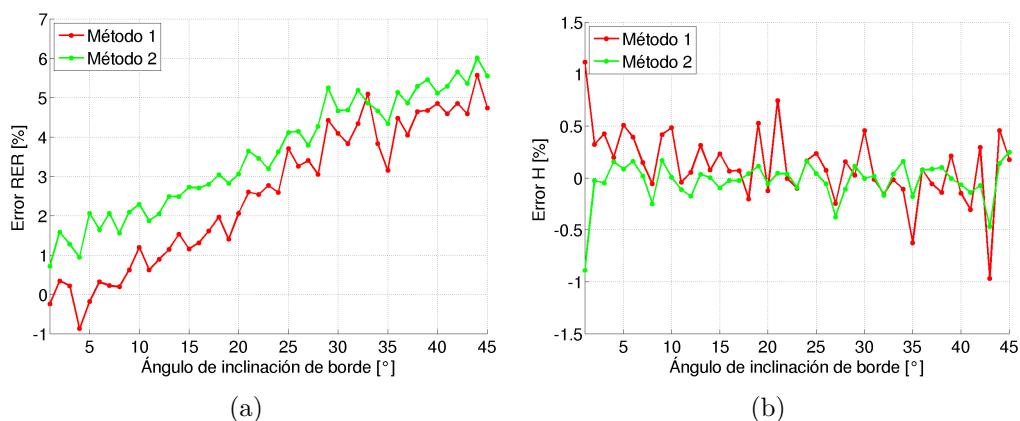


Fig. 4.17. Error de estimación de (a) RER y (b) H, para imágenes con MTFC aplicada, en función del ángulo de inclinación de borde.

Tabla 4.2 – Tiempo de ejecución promedio de cada método de reconstrucción de la ESF.

	MTFC on	MTFC off
	Tiempo promedio [s]	Tiempo promedio [s]
Método 1	0,016	0,019
Método 2	0,483	0,570

4.3. Estimación de la SNR

Dada una imagen que contiene un borde objetivo, usando el conjunto de datos resultantes de la respuesta al borde ordenada y corregida por inclinación

(4.12), es posible calcular la relación señal a ruido aplicando la métrica basada en la imagen (3.31). Para una respuesta al borde dada por el conjunto de datos (4.12) los valores comprendidos entre -5 y -3 píxel corresponden a la zona de ρ_{low} (zona oscura) y los valores comprendidos entre $+3$ y $+5$ píxel corresponden a la zona de ρ_{high} (zona clara), como se ilustra en la Fig. 4.18. Los valores comprendidos en entre -3 y $+3$ se excluyen ya que está presente el borde y eventualmente el sobrepico. Tomando el valor medio y la desviación estándar en cada zona, la SNR queda determinada por

$$SNR = \frac{\mu_{\rho_{high}} - \mu_{\rho_{low}}}{(\sigma_{\rho_{high}} + \sigma_{\rho_{low}})/2}. \quad (4.19)$$

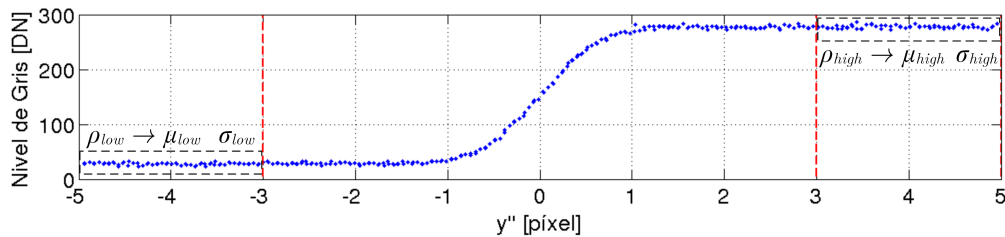


Fig. 4.18. Zonas de ρ_{low} y ρ_{high} para el cálculo de SNR en datos de respuesta al borde ordenada y corregida por inclinación.

En cuanto al valor de G , algunos autores asumen un valor arbitrario [80], utilizan valores publicados en la bibliografía o que resultan de acuerdo a los valores del núcleo del filtro [76, 77]. No obstante la *GIQE* es una ecuación que fue desarrollada para predecir el NIIRS de un instrumento durante su etapa de diseño incluyendo el posprocesamiento y el término $\frac{SNR}{G}$ indica la SNR de una imagen posprocesada [66, 84], es decir el valor final de SNR. Por lo tanto como la SNR es estimada directamente de la imagen en este caso se está estimando directamente el término $\frac{SNR}{G}$, lo que equivale a considerar $G = 1$ para el cálculo del NIIRS y SNR igual al valor estimado de la imagen.

En la Fig. 4.19 puede verse el error de estimación de la SNR, que se calcula entre el valor estimado de la manera antes mencionada y el valor de SNR real calculado durante generación del borde objetivo simulado. Se ensayó para objetivos con su borde inclinado entre 1° y 45° en imágenes con y sin MTFC aplicada. Para cada ángulo de inclinación se realizaron 10 corridas, en la Fig. 4.19 se muestra el error de estimación para cada una de las corridas y el valor promedio para cada ángulo. Puede verse que el error de estimación de SNR presenta valores aleatorios para todo el rango de ángulos de inclinación, mientras que el valor máximo relevado es del 23%.

Si bien un error del 23 % parece grande, debe considerarse su impacto en el NIIRS. Si se considera el término de la GIQE que depende de la SNR, con $G = 1$ por lo antes mencionado, se tiene

$$NIIRS_{SNR} = -0,334 \frac{1}{SNR}. \quad (4.20)$$

El $NIIRS_{SNR}$ con un error porcentual en la estimación de SNR, $\varepsilon_{\%SNR}$, puede calcularse por

$$NIIRS_{SNR+\varepsilon_{\%SNR}} = -0,334 \frac{1}{SNR \left(1 + \frac{\varepsilon_{\%SNR}}{100}\right)}. \quad (4.21)$$

Considerando que para los objetivos simulados $SNR \cong 33$, el error de estimación del NIIRS es

$$\begin{aligned} \Delta NIIRS_{\varepsilon_{\%SNR}} &= NIIRS_{SNR+\varepsilon_{\%SNR}} - NIIRS_{SNR} \\ &= -0,334 \frac{1}{SNR} \left(\frac{1}{1 + \frac{\varepsilon_{\%SNR}}{100}} - 1 \right) \\ \Delta NIIRS_{\varepsilon_{\%SNR}} &= -0,334 \frac{1}{33} \left(\frac{1}{1 + \frac{23}{100}} - 1 \right) = 3,6 \cdot 10^{-3}, \end{aligned} \quad (4.22)$$

es decir, 27 veces menor que el valor mínimo visualmente perceptible (0,1).

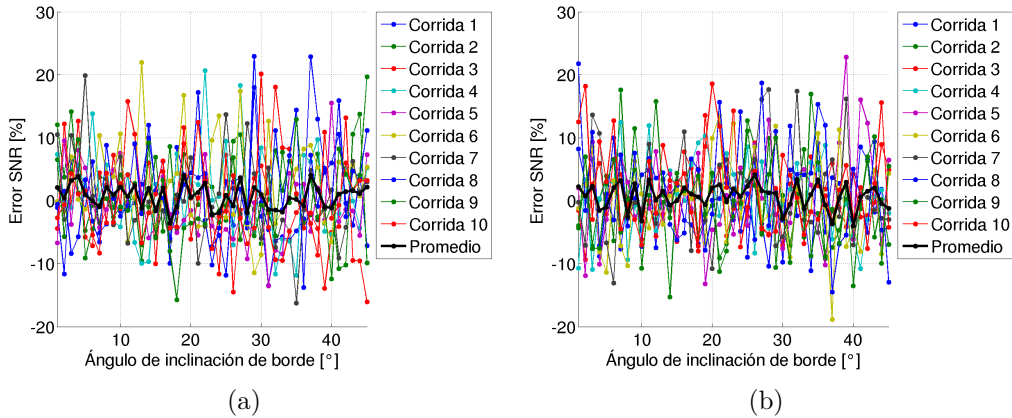


Fig. 4.19. Error de estimación de SNR para imágenes (a) sin MTFC aplicada y (b) con MTFC aplicada, en función del ángulo de inclinación de borde. Se muestra el resultado de 10 corridas para cada ángulo y el valor promedio.

4.4. Estimación del NIIRS

Finalmente la estimación del NIIRS se hace valuando la GIQE 4.0 con los valores de RER, H y SNR estimados con los algoritmos antes descriptos. Se

evaluó el desempeño en la estimación del NIIRS con bordes objetivos simulados con ángulos que van desde 1° a 45° en pasos de 1° para imágenes sin aplicación MTFC y con su aplicación. Si bien se ha mencionado que el método de remuestreo y suavizado basado en la spline suavizada tiene un mejor comportamiento, aquí se muestran los resultados de la estimación tanto de este método como del método 2 con el objeto de comprobar el criterio antes adoptado.

En Fig. 4.20 se muestra el error de estimación del NIIRS donde el NIIRS real se calcula a partir de los valores de RER, H y SNR dados durante la generación del borde objetivo y el GSD se asumió $1m$ tanto para el NIIRS estimado como para el NIIRS real.

En ambos casos puede observarse que, en general, el método 1 produce mejores estimaciones tanto para imágenes a las que se aplicó MTFC como las que no. Por otro lado se aprecia que el error de estimación es mayor cuando se ha aplicado el posprocesamiento a la imagen y que éste está principalmente dominado por el error en la estimación de RER y se incrementa conforme el ángulo de inclinación del borde se acerca a los 45° .

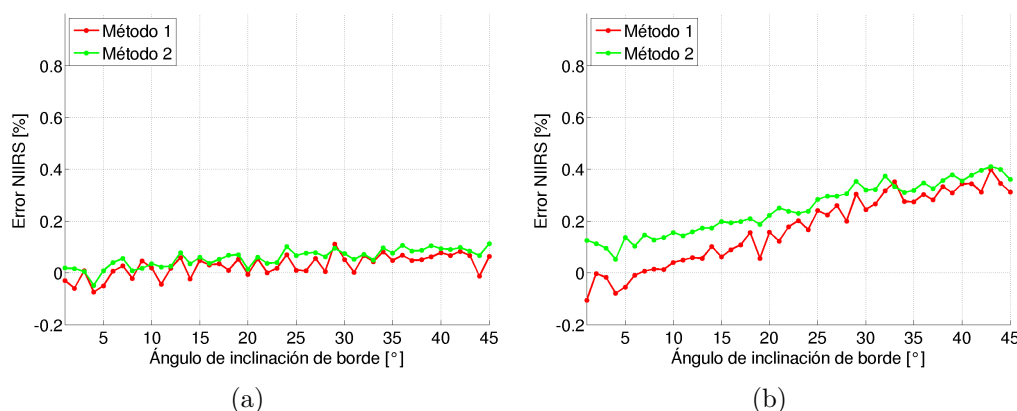


Fig. 4.20. Error de estimación de NIIRS para imágenes (a) sin MTFC aplicada y (b) con MTFC aplicada, en función del ángulo de inclinación de borde.

4.5. Algoritmo de estimación del NIIRS en imágenes satelitales

Ya evaluadas todas las alternativas de procesamiento para la obtención de la ESF y habiendo mostrado los resultados del desempeño del algoritmo de estimación del NIIRS en imágenes simuladas, resta aplicar el mismo a imágenes satelitales reales y evaluar su desempeño. La principal diferencia es que para bordes simulados la estimación se realizó sobre un único borde, mientras que

para imágenes satelitales, la estimación de NIIRS se hace en todos los bordes presentes en la imagen y se obtiene un valor medio.

En la Fig. 4.21 puede verse el diagrama de flujo del algoritmo para la estimación del NIIRS en una imagen satelital donde se aplican los algoritmos desarrollados en las secciones precedentes.

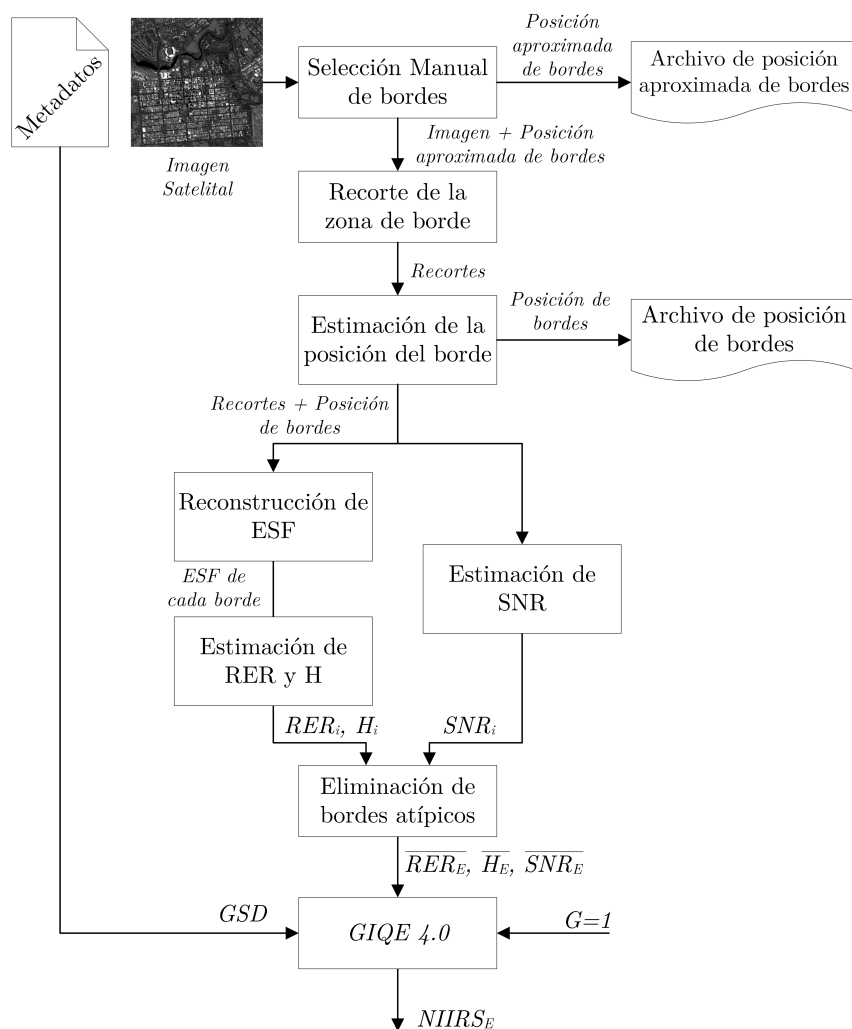


Fig. 4.21. Diagrama del algoritmo de estimación del NIIRS.

En primer lugar, dada una imagen satelital, se seleccionan manualmente los bordes aptos para la estimación de la ESF. No es necesario indicar la posición exacta del borde, sino sólo la posición aproximada. Para que un borde natural sea apto como borde objetivo debe cumplir una serie de requisitos:

- ser recto,
- tener una longitud mayor a 10 píxeles,
- la zona clara y la zona oscura del borde deben ser uniformes,

- desde la posición del flanco, la zona clara y la zona oscura deben tener un ancho de al menos 5 píxeles.

Si bien existen técnicas automáticas para la detección de bordes rectos como los detectores de borde de Sobel o Canny junto con la transformada de Hough [14], estas técnicas, aplicadas a las imágenes satelitales que se utilizan en este trabajo, entregan gran cantidad de bordes y resulta complejo desarrollar un algoritmo para descartar los bordes que no cumplen los requisitos antes mencionados. Por tal motivo, se optó por seleccionar manualmente la posición de los bordes. Esta tarea si bien es laboriosa y demanda demasiado tiempo, permitió asegurar que los bordes utilizados para la estimación del NIIRS tengan las características necesarias.

Para facilitar la tarea de la selección manual de bordes se desarrolló una herramienta que muestra la imagen satelital en la que el usuario puede navegar, trazar una línea en la posición aproximada del borde y al finalizar la selección, almacena un archivo que contiene la posición aproximada de todos los bordes seleccionados. En la Fig. 4.22(a) puede verse una imagen satelital en la que se han seleccionado 135 bordes y en la Fig. 4.22(b) pueden verse algunos de los bordes seleccionados.

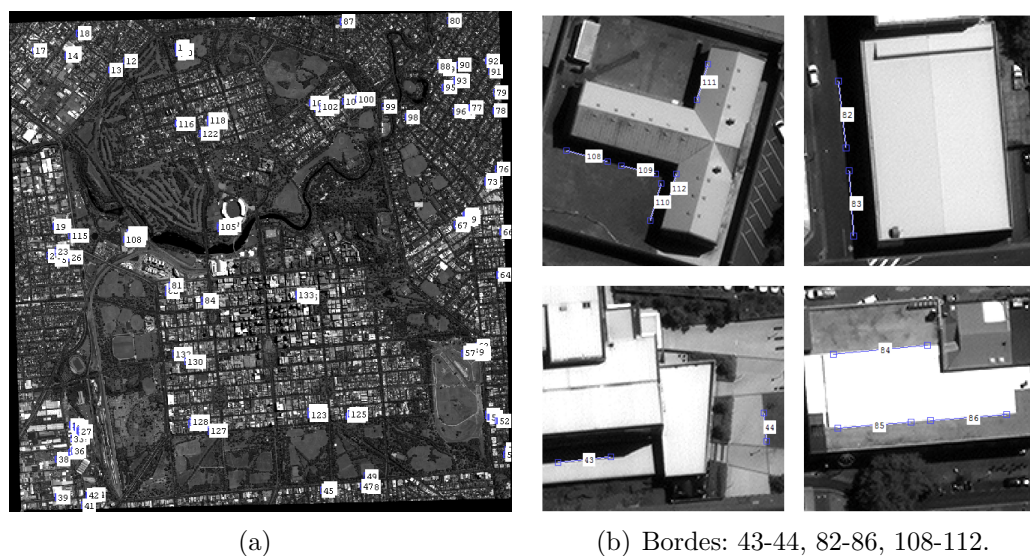


Fig. 4.22. Bordes seleccionados manualmente. (a) Imagen WV2-01-P y (b) bordes: 43-44, 82-86, 108-112.

Una vez seleccionados todos los bordes, se utiliza su posición aproximada (indicada manualmente) para realizar un recorte de la imagen que lo contenga. Los recortes con bordes cuyos ángulos respecto a la vertical son $|\theta_{borde}| > 45^\circ$ se rotan 90° de modo que todos queden con $|\theta_{borde}| < 45^\circ$. Los bordes con

$|\theta_{borde}| = 45^\circ$ se descartan ya que para estos casos las respuestas al borde de cada fila están separadas un píxel y por lo tanto no se logra mitigar el problema de submuestreo. Luego la imagen se espeja respecto a la vertical para que la zona oscura se encuentre a la izquierda y la clara a la derecha. De este modo todos los bordes se tratan por igual y las estimaciones de RER y H no diferencian entre la dirección x e y , sino que se obtiene un único valor promedio.

Para cada recorte, se aplica el algoritmo de estimación de la posición del borde descrito en la Sección 4.1 y se almacena un archivo que contiene la posición del borde en el recorte.

Luego, para cada recorte, haciendo uso de la información de la posición de cada borde, se aplica el algoritmo de reconstrucción de la ESF descrito en Sección 4.2. En las imágenes de la primera y tercera columna de la Fig. 4.23 pueden verse los recortes para algunos de los bordes mostrados en la Fig. 4.22, donde en azul se indica la recta definida por la selección manual del borde, en puntos rojos la primera aproximación a la posición del borde por la derivada de primer orden y en verde la posición del borde estimada con el algoritmo completo. En las imágenes de la segunda y cuarta columna de la Fig. 4.23 pueden verse los puntos de las respuesta al borde de cada fila alineados y la ESF resultante para cada borde. A partir de cada una de las ESF se obtiene el valor de RER y H para cada borde.

Paralelamente a la reconstrucción de la ESF, se calcula el valor de SNR usando la métrica basada en la imagen acorde a la Sección 4.3.

Hasta este punto, para cada borde seleccionado manualmente, se tienen los valores de RER, H y SNR, denominados RER_i , H_i y SNR_i . Antes de calcular el NIIRS, se calculan la media, μ , y la desviación estándar, σ , para RER_i , H_i y SNR_i ; y se extraen del conjunto de datos todos los bordes que están fuera del intervalo de $\mu \pm 2\sigma$ para cualquiera de los tres parámetros. Luego, se calcula nuevamente la media para cada parámetro con los bordes que están dentro del intervalo $\mu \pm 2\sigma$, resultando los valores estimados de cada parámetros para la imagen, denominados \overline{RER}_E , \overline{H}_E y \overline{SNR}_E .

Finalmente, con estos valores medios, más el valor del GSD extraído de los metadatos y considerando $G = 1$ (ya que SNR se estimó sobre la imagen) se aplica la GIQE 4.0 para estimar el NIIRS de la imagen, $NIIRS_E$.

En las Figs. E.1 a E.11 del Anexo E puede verse el histograma de cada uno de los parámetros, para las imágenes utilizadas para la validación del algoritmo.

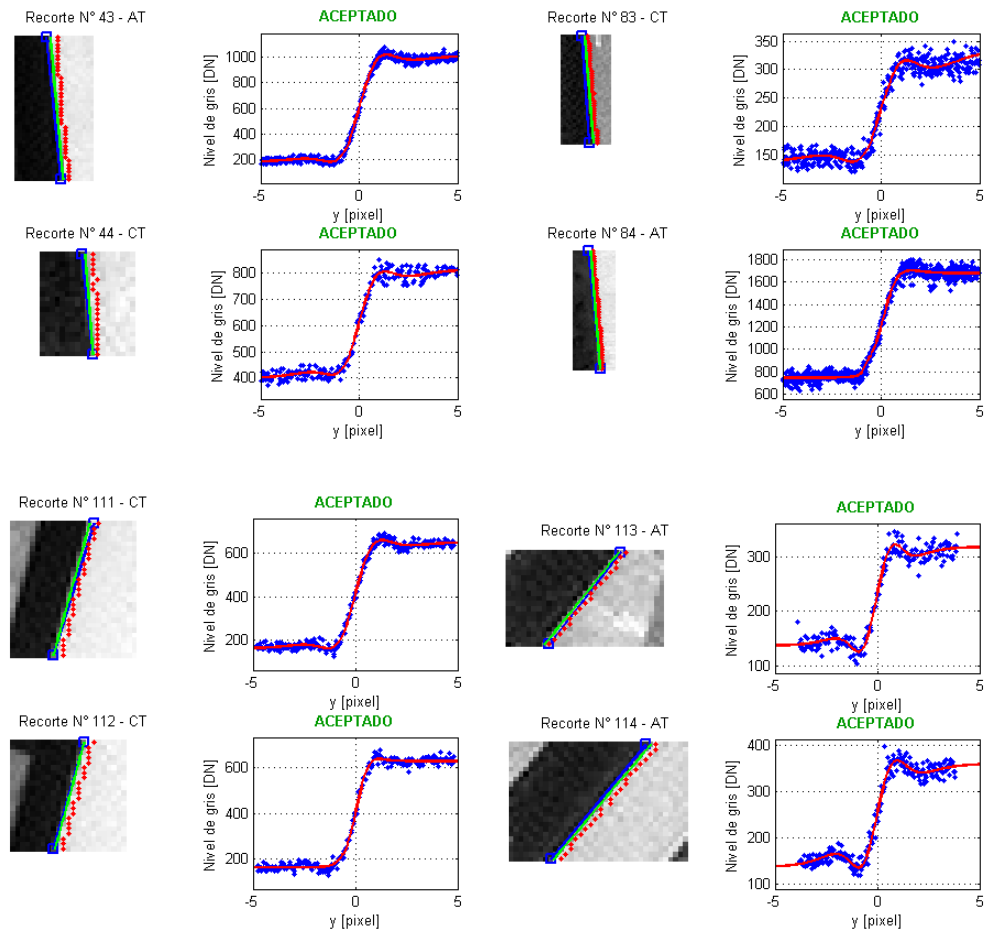


Fig. 4.23. Recortes de bordes, posición del borde y ESF reconstruida.

4.6. Validación del algoritmo de estimación del NIIRS

La validación del algoritmo de estimación del NIIRS se llevó a cabo utilizando un conjunto de imágenes cuyo NIIRS es conocido, ya sea porque está publicado en los metadatos de la imagen, como PNIIRS (*Predicted NIIRS*), o publicado en un documento de la empresa proveedora de las imágenes. En la Tabla 4.3 se muestran algunas características de las imágenes de prueba tales como: un código asignado en este trabajo para referirse a la imagen; el nombre del satélite que la adquirió; la zona y fecha de adquisición; el tipo de posprocesamiento aplicado; la cantidad de bits B , con que está cuantificada la imagen; la media geométrica del GSD, GSD_{GM} ; el GSD resultante del proceso de ReMuestreo (RM), GSD_{RM} ; el RER en los casos que se encontró publicado y el NIIRS de la imagen. Para la estimación del del NIIRS se utilizó el GSD_{GM} ya que el mismo valor es empleado para el cálculo del PNIIRS [66].

Tabla 4.3 – Características de las imágenes usadas para la validación de NIIRS.

Código	Satélite	Fecha	Zona	Banda	Posproceso	B [bpp]	GSD_{GM} [m]	GSD_{RM} [m]	RER	NIIRS
GE1-01-P	GeoEye-1 ^a	31/07/2013	Ciudad del Cabo, Sudáfrica	PAN	RM y MTFC	11 ^c	0,434 ^c	0,50 ^c	-	5,40 ^c
GE1-01-M	GeoEye-1 ^a	31/07/2013	Ciudad del Cabo, Sudáfrica	MS	RM y MTFC	11 ^c	2,452 ^c	2,00 ^c	-	3,10 ^c
QB2-01-P	QuickBird-2 ^a	18/09/2013	Yaipur, India	PAN	RM y MTFC	11 ^c	0,623 ^c	0,60 ^c	0,76 a 0,84 [74]	4,70 ^c
QB2-01-M	QuickBird-2 ^a	18/09/2013	Yaipur, India	MS	RM y MTFC	11 ^c	2,490 ^c	2,40 ^c	-	2,70 ^c
IK2-01-P	Ikonos-2 ^a	09/09/2002	Sao Paulo, Brasil	PAN	RM y MTFC	11 ^c	0,830 ^c	0,80 ^c	0,70 [73]	4,28 ^d
IK2-02-P	Ikonos-2 ^b	29/12/2004	Puducherry, Inda	PAN	RM y MTFC	8 ^c	1,167 ^c	1,00 ^c	0,70 [73]	3,81 ^d
WV1-01-P	WorldView-1 ^a	21/05/2013	Boulder, USA	PAN	RM y MTFC	11 ^c	0,548 ^c	0,50 ^c	-	4,90 ^c
WV2-01-P	WorldView-2 ^a	21/08/2012	Sídney, Australia	PAN	RM	11 ^c	0,474 ^c	0,50 ^c	-	5,00 ^c
WV3-01-P	WorldView-3 ^a	27/11/2014	Adelaide, Australia	PAN	RM y MTFC	11 ^c	0,321 ^c	0,30 ^c	-	5,60 ^c
WV3-01-M	WorldView-3 ^a	27/11/2014	Adelaide, Australia	MS	RM y MTFC	11 ^c	1,284 ^c	1,20 ^c	-	3,60 ^c
ERB-01-P	EROS-B ^a	23/09/2007	New York, USA	PAN	-	10 ^c	0,67 ^c	-	-	4,60 [85]

^a Muestras gratis Apollo Mapping.

^b Obtenida del servidor web *Earth Explorer* de USGS.

^c Extraído de metadatos de la imagen.

^d Para $GSD = 0,82 \Rightarrow NIIRS = 4,3$ [86], se calculó el valor de NIIRS para un GSD de 0,82m y 1,167m con [84]

$$NIIRS@GSD_2 = NIIRS@GSD_1 - 3,16 \cdot \log_{10} \left(\frac{GSD_2}{GSD_1} \right). \quad (4.23)$$

La imagen MS del satélite WorldView-3, WV3-01-M, tiene 8 bandas en lugar de 4 como el resto [87], por lo tanto se utilizaron las bandas 2, 3, 5 y 7 que corresponden a los espectros B, G, R y NIR respectivamente.

Para cada una de las imágenes se seleccionaron N_b bordes manualmente y luego se aplicó el algoritmo de estimación del NIIRS. Los resultados se muestran en la Tabla 4.4, donde: $NIIRS_R$ es el valor extraído de metadatos o bibliografía, que se asume real, $NIIRS_E$ el estimado y $\Delta NIIRS$ el error entre estos dos. Los valores de $\overline{RER_E}$, $\overline{H_E}$ y $\overline{SNR_E}$ son la media de todos los bordes luego de extraer los valores atípicos.

Tabla 4.4 – Resultados de validación de NIIRS (método 1 para la reconstrucción de la ESF).

Código	$NIIRS_R$	N_b	GSD [in]	$\overline{RER_E}$	$\overline{H_E}$	$\overline{SNR_E}$	$NIIRS_E$	$\Delta NIIRS$
GE1-01-P	5,40	79	17,09	0,7419	1,01	24,2	5,31	-0,09
GE1-01-M	3,10	116	96,54	0,8792	1,16	12,2	3,04	-0,06
QB2-01-P	4,70	73	24,53	0,7876	1,08	11,3	4,83	0,13
QB2-01-M	2,70	44	98,03	0,8717	1,13	8,4	3,00	-0,30
IK2-01-P	4,28	86	32,69	0,6876	1,03	15,5	4,31	0,03
IK2-02-P	3,81	91	45,96	0,6932	1,12	8,1	3,76	-0,05
WV1-01-P	4,90	93	21,57	0,8376	1,15	17,1	5,04	0,14
WV2-01-P	5,00	79	18,66	0,4585	0,93	41,8	4,66	-0,34
WV3-01-P	5,60	135	12,64	0,7391	1,06	24,1	5,69	0,09
WV3-01-M	3,6	116	50,55	0,8171	1,09	35,7	3,89	0,29
ERB-01-P	4,60	95	26,38	0,8384	1,07	22,4	4,83	0,23

Para las imágenes de los satélites QuickBird-2 e Ikonos-2 se conoce el valor de la RER para la banda PAN que fue estimado por NASA haciendo uso del borde objetivo artificial del SSC [73, 74]. Considerando estos valores como reales, para el satélites Ikonos-2 la RER fue estimado con un error del 1% y para el satélite QuickBird-2 la RER se encuentra dentro del rango especificado.

También puede apreciarse que para todos los casos el valor de RER en la imagen multiespectral es mayor al valor de RER en la imagen PAN, como es esperado. Esto se debe a que el píxel del detector MS es cuatro veces más grande que el píxel del detector PAN, por lo tanto la F_{Ny} es menor, lo que implica un mayor valor de $MTF@F_{Ny}$ y consecuentemente mayor RER.

Si se observan las características de las dos imágenes de Ikonos-2, puede notarse que éstas difieren en el GSD geométrico, y en la cantidad de niveles de cuantificación. A pesar de estas diferencias se obtienen estimaciones de RER aproximadamente iguales, como es de esperar.

Analizando los valores de $NIIRS_E$ y el error respecto al $NIIRS_R$ puede verse que las estimaciones son representativas del nivel de interpretabilidad de

la imagen. Todos los errores se mantienen debajo de 0,34 y cinco de los once resultados (un 45 %) debajo de 0,1, diferencia de NIIRS que no es perceptible y no afecta a la interpretabilidad de la imagen. En la Fig. 4.24 se muestran los valores de NIIRS estimado junto con la recta de pendiente unitaria que expresa la situación deseada. Para el conjunto de imágenes de prueba el algoritmo de estimación del NIIRS predice el NIIRS con un $RMSE = 0,19$ y tiene un coeficiente de determinación de $R^2 = 0,95$.

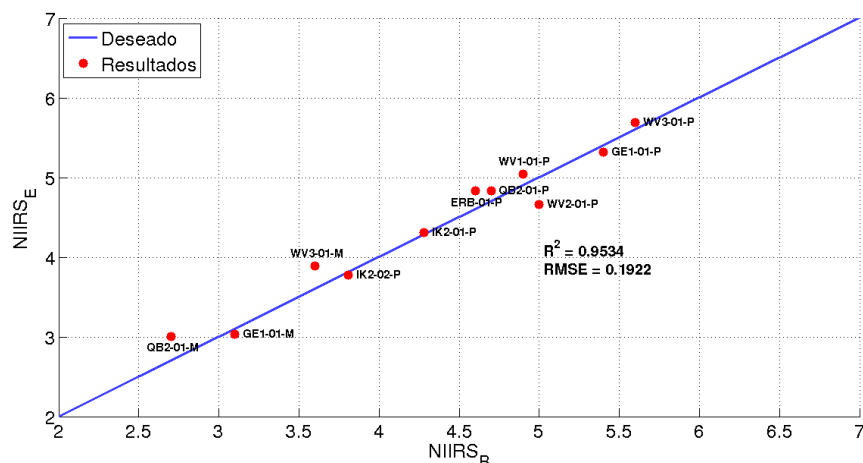


Fig. 4.24. Resultados de estimación de NIIRS vs. NIIRS Real (método 1 para la reconstrucción de la ESF).

Por completitud en el Anexo F se muestran los resultados de estimación de NIIRS para el mismo conjunto de imágenes utilizando el método 2 para el remuestreo y suavizado. En la Tabla 4.5 puede verse una comparación entre los dos métodos en términos del coeficiente de determinación, la raíz del error cuadrático medio en la estimación y el tiempo de ejecución promedio por borde. Esto muestra que el método 1 es el mejor en todos los aspectos, aseverando el criterio adoptado en la Sección 4.2.3.3.

Tabla 4.5 – Comparación de estimación de NIIRS de imágenes de NIIRS conocido entre método 1 y método 2 para la reconstrucción de la ESF.

	R^2	$RMSE$	Tiempo por borde promedio [s]
Método 1	0,9534	0,1922	0,3987
Método 2	0,9306	0,2345	1,1969

A modo de apreciación final, puede decirse que se evaluaron distintas alternativas para estimar el NIIRS tanto con imágenes simuladas como con imágenes satelitales reales y se seleccionó la opción que estima con menor error.

Cabe destacar que una validación completa del algoritmo debería llevarse a cabo contrastando los resultados de las estimaciones con el NIIRS observado por un analista de imagen y para mayor cantidad de imágenes de prueba. Situación que queda fuera del alcance de esta tesis por no disponer de tal idóneo ni de mayor cantidad de imágenes de NIIRS conocido. Por otro lado, los errores del algoritmo aquí informados quedan sujetos a la fidelidad de los valores de NIIRS que se asumieron como reales.

4.7. Algoritmo para cálculo de la degradación del NIIRS

En la Fig. 4.25 se muestra el diagrama de flujo del algoritmo para calcular la degradación del NIIRS en una imagen satelital debido a un proceso de compresión con pérdidas.

Dada una imagen satelital sobre la que previamente se ha realizado la selección manual de bordes, y por lo tanto se tiene el archivo de la posición aproximada de los bordes, se calcula el $NIIRS_0$ con el algoritmo ya descrito. En el proceso se tiene como salida el archivo de la posición de los bordes.

Luego, la imagen original es comprimida y descomprimida con el sistema de compresión y configuración deseada. Usando la posición aproximada de los bordes seleccionados manualmente se extraen los recortes de la imagen recuperada. Luego usando el archivo de la posición de los bordes calculada sobre la imagen original y los recortes de la imagen recuperada se completa el algoritmo de estimación del NIIRS para obtener el $NIIRS_1$.

Finalmente se obtiene la degradación de NIIRS, haciendo la diferencia $\Delta NIIRS = NIIRS_1 - NIIRS_0$.

Adicionalmente, se calcula el aporte a la degradación del NIIRS de cada uno de los términos de la GIQE que contiene alguno de los parámetros que se ve modificado por la compresión con pérdidas. Éstos están dados por las ecuaciones,

$$\Delta NIIRS_{RER} = c_2 \log_{10} \left(\frac{\overline{RER_1}}{\overline{RER_0}} \right), \quad (4.24)$$

$$\Delta NIIRS_{G/SNR} = c_3 \left(\frac{G}{SNR_1} - \frac{G}{SNR_0} \right), \quad (4.25)$$

$$\Delta NIIRS_H = c_4 (\overline{H_1} - \overline{H_0}). \quad (4.26)$$

Donde c_2 , c_3 y c_4 fueron definidos en la Tabla 3.3 y se consideró $G = 1$.

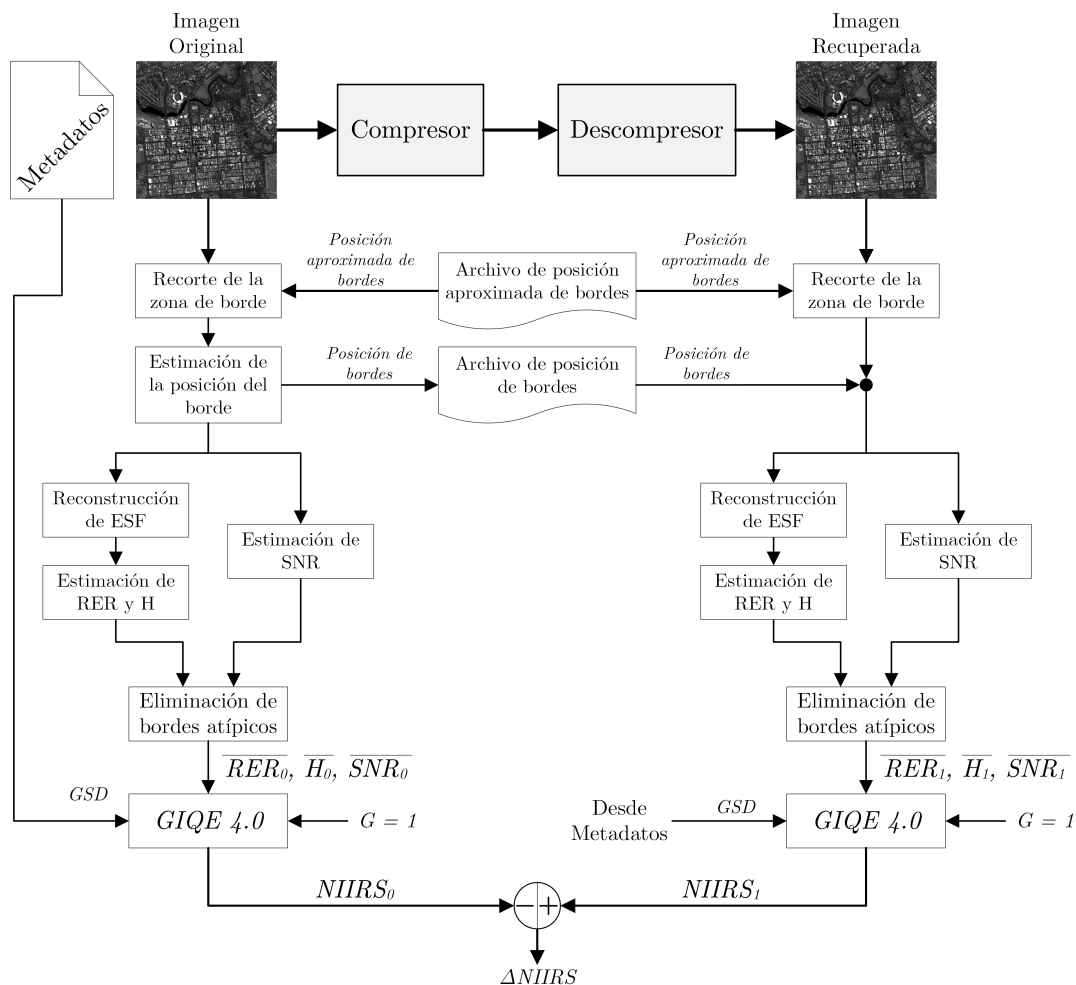


Fig. 4.25. Diagrama del algoritmo de estimación del $\Delta NIIRS$ por compresión con pérdidas.

Evaluación de sistemas de compresión

5.1. Selección de imágenes de prueba

Para evaluar el desempeño de los cuatro sistemas de compresión preseleccionados y finalmente seleccionar uno para la CAR, es necesario realizar ensayos con imágenes de prueba. Se desea realizar pruebas con imágenes de características iguales o similares a las que adquiriría la CAR, de zonas urbanas y zonas rurales, tanto PAN como MS.

Cuatro alternativas para imágenes de prueba son:

- 1) Utilizar imágenes del mismo instrumento, si este ya fue utilizado en órbita.
- 2) Utilizar imágenes del instrumento previamente montado en un avión.
- 3) Utilizar imágenes satelitales o aéreas de menor resolución espacial para simular imágenes adquiridas por el instrumento bajo estudio.
- 4) Utilizar imágenes satelitales de características similares a las del instrumento bajo estudio.

La primera alternativa, representa la situación ideal, ya que se tiene la certeza de que se seleccionará un sistema de compresión habiendo ensayado con imágenes iguales a las que el compresor procesará en su futura operación. Desafortunadamente, la CAR aun está bajo diseño y no se utilizó en órbita ni en aviones, lo que conduce a descartar tanto la opción 1 como la 2.

La tercera alternativa, implica utilizar imágenes de GSD menor adquiridas por otros instrumentos y procesarlas para simular imágenes representativas en términos de GSD, respuesta en frecuencia y radiometría. En [88] puede verse

un caso donde se simularon imágenes MS de $GSD = 20m$ a partir de imágenes de $GSD = 5m$. Para obtener las imágenes simuladas se realizaron correcciones atmosféricas, filtrado espacial acorde a las características del sistema óptico a simular, remuestreo y adición de ruido. Debido a la complejidad de la mencionada técnica y aspectos programáticos de esta tesis, esta metodología no fue empleada. No obstante, podría ser tenida en cuenta en trabajos futuros. El Instituto Geográfico Nacional (IGN) de Argentina realiza adquisiciones aéreas con una cámara Vexcel UltraCamXP cuyas características podrían ser aptas para simular imágenes de la CAR.

La cuarta alternativa, que consiste en utilizar imágenes de otros instrumentos satelitales con características similares a la CAR, fue la empleada en esta tesis. Para ello, se buscaron imágenes similares en términos de: características espectrales, GSD y respuesta en frecuencia que estuvieran disponibles en CONAE, INVAP o libres. Adicionalmente es deseable que las imágenes de prueba no hayan sido comprimidas con pérdidas y que no incluyan posprocesamientos como remuestreo, MTFC y georreferenciación. Dichas características se resumen en la Tabla 5.1 junto con otras características de interés.

Luego de un proceso de búsqueda se encontraron 6 fuentes de imágenes de alta resolución cuyas características se muestran en la Tabla 5.1. Puede verse que ninguna de las alternativas cumple totalmente con lo deseado, esto no es fácil de satisfacer ya que cada instrumento tienen sus propios requerimientos de diseño. Incluso en algunos casos los fabricantes del instrumento, o proveedores de las imagen, no publican todas las características del sistema. Por otro lado, es habitual que los instrumentos de alta resolución incluyan sistemas de compresión con pérdidas a bordo y que los productos entregadas por los proveedores incluyan remuestreo y MTFC.

La CONAE dispone de imágenes del territorio nacional de SPOT 6/7 y EROS-B en su catálogo. En el primer caso existe gran cantidad de imágenes en la base de datos. En el segundo caso sólo existe una reducida cantidad y de ciertas áreas específicas del país. Las imágenes de SPOT 6/7 cumplen la mayoría de lo requerido aunque su principal problema es que el instrumento adquiere con un GSD geométrico de $2,2m$ en la banda PAN y $8,8m$ en la banda MS y se aplica un remuestreo que reduce el GSD a $1,5m$ para la banda PAN y $6,0m$ para la banda MS. Además el posprocesamiento incluye una compensación de MTF que mejora $MTF@F_{Ny}$ de 0,16 a 0,3 [13]. Por otro lado, las imágenes de EROS-B son solamente de la banda PAN y tienen un GSD menor, pero gozan

Tabla 5.1 – Opciones de imágenes de prueba.

Característica	Deseado	SPOT 6/7	EROS-B	OrbView-3	Ikonos-2	Muestras Apollo Mapping	Imágenes de prueba de CCSDS _d	
Configuración de bandas	PAN + 4 MS	Si	No	Si	Si	Si, excepto imagen de EROS-B	No	
Longitudes de onda [nm]	PAN	450 - 900	450 - 745	500 - 900	450 - 900	526 - 929	^c	^e
	B	450 - 520	450 - 520	N/A	450 - 520	445 - 516	^c	^e
	G	520 - 590	530 - 590	N/A	520 - 600	506 - 595	^c	^e
	R	630 - 690	625 - 695	N/A	625 - 695	632 - 698	^c	^e
GSD [m]	PAN	1	1,5 _b (2,2)	0,7	1	1 (0,82) _b	^c	^e
	MS	4	6,0 _b (8,8)	N/A	4	4 (3,28) _b	^c	^e
$MTF@F_{ny}$ banda PAN	>0,2 ^a	0,3 _b (0,16)			0,15 _b (0,10)	0,48 _b (0,11)	^c	^e
B [bpps]	10	12	10	11	8	10; 11; 12; 14 ^c	8; 10; 12 _e	
Compresión con pérdidas	No	No	No	Compresión CR = 5,5	Si, ADPCM Kódac	^c	Se desconoce	
Posprocesamiento	No	Remuestreo y MTFC	No	PAN=MTFC MS=No	Remuestreo y MTFC	Depende de la imagen pero en general remuestreo y MTFC	Se desconoce	
Ancho de la imagen [pixel]	PAN	12.288	40.000	10.380	8.000	13.500	^c	^e
	MS	3.072	10.000	N/A	2.000	3.375	^c	^e

^a Valor del diseño actual de la CAR.

^b El valor fuera de los paréntesis es el final y el que está dentro de los paréntesis es el valor previo al posprocesamiento.

^c Depende del caso, hay una imagen de cada uno de los siguientes satélites: GeoEye-1, Ikonos-2, KOMPSAT-3, Pleiades-1, QuickBird-2, SPOT-6, WorldView-1, WorldView-2, WorldView-3 y EROS-B.

^d El conjunto de imágenes de prueba utilizadas para el diseño del algoritmo CCSDS-IDC contempla en total 30 imágenes no necesariamente de alta resolución, de observación de la tierra y de otros cuerpos celestes, ópticas y de radar; aquí sólo se tienen en cuenta las 4 imágenes ópticas de observación de la tierra de alta resolución.

^e Depende del caso, hay imágenes de varios satélites: SPOT-3, SPOT-5 y Pleiades-1.

con la ventaja de no haber sido comprimidas con pérdidas a bordo y no tener posprocesamientos [89].

Las imágenes de OrbView-3 están disponibles libremente en el servidor web *Earth Explorer* del Servicio Geológico de los Estados Unidos (USGS, *United States Geological Survey*). Estas imágenes fueron comprimidas a bordo, se desconoce con que sistema de compresión, pero debido a que $CR = 5,5$ se infiere en que se utilizó un sistema con pérdidas. Aunque estas imágenes fueron, presuntamente, comprimidas con pérdidas y las imágenes de la banda PAN tienen un posprocesamiento de MTFC, son las más representativas en términos de GSD [90].

En el mismo servidor del USGS pueden encontrarse imágenes del satélite Ikonos-2, que si bien presentan muchas características similares a lo deseado, están cuantificadas con $B = 8bpps$, valor menor al deseado. Además hay disponible una baja cantidad y sólo de zonas rurales. Por éstos motivos no fueron utilizadas y se muestran en la tabla sólo por completitud.

La quinta fuente de imágenes presentada en la tabla corresponde a las imágenes de muestra provistas por *Appollo Mapping*, de las cuales algunas (las de NIIRS conocido) fueron utilizadas en el capítulo anterior. Ya que hay una imagen de cada uno de los siguientes satélites: GeoEye-1, Ikonos-2, KOMPSAT-3, Pleiades-1, QuickBird-2, SPOT-6, WorldView-1, WorldView-2, WorldView-3 y EROS-B; las características dependen de cada caso. En líneas generales puede destacarse que:

- todas poseen banda PAN y al menos 4 bandas MS con longitudes de onda similares, excepto la imagen de EROS-B;
- todas son de $GSD_{PAN} < 1m$ excepto la de SPOT 6;
- la imagen de EROS-B es de $10bpp$, las de GeoEye-1, Ikonos-2, QuickBird-1, WorldView-1/2/3 de $11bpp$; las de Pleiades-1 y SPOT-6 de $12bpp$ y la de KOMPSAT-3 de $14bpp$;
- la mayoría incluye compresión con pérdidas o no se informa,
- la mayoría incluye un proceso de remuestreo y MTFC,
- en total son 10 imágenes de áreas urbanas.

La última fuente son 4 imágenes de observación de la tierra de alta resolución de los satélites SPOT-3, SPOT-5 y Pleiades-1 que fueron utilizadas para el diseño del sistema de compresión CCSDS-IDC. Debido a la baja cantidad y a que se desconoce de las características, finalmente no fueron utilizadas y se presentan en la tabla por completitud.

Las imágenes de SPOT 6/7 y de OrbView-3 de la banda PAN sirvieron como datos de prueba tanto para zonas urbanas como para zonas rurales y las de la banda MS sólo sirvieron para zonas rurales, ya que en zonas urbanas el GSD resultó demasiado grande para las estructuras presentes y no fue posible seleccionar bordes para la estimación del NIIRS. Las imágenes de EROS-B, fueron utilizadas como datos de prueba para zonas urbanas ya que sólo hay disponible de este tipo estas áreas. Dado que con estas tres fuentes no fue posible contribuir a conjunto de datos MS de zonas urbanas, se utilizaron las imágenes de GeoEye-1, QuickBird-2 y WorldView-3 que fueron utilizadas en el capítulo anterior.

Los productos obtenidos de cada satélite, en algunos casos debieron ser modificados para ser aptos como imágenes de prueba para los sistemas de compresión. Estas modificaciones fueron:

- **Recorte del producto:** para obtener una imagen lo suficientemente grande como para contener una cantidad de bordes adecuada para la estimación del NIIRS y lo más pequeña posible para evitar consumir demasiado tiempo durante su compresión, descompresión y cálculo de métricas.
- **Ajuste del Rango Dinámico (RD):** para que todas las imágenes queden cuantificadas con $10bpp$, se aplicó sólo a los productos de $B > 10bpp$. En algunos casos, donde los histogramas mostraron que la cantidad de píxeles por encima de 1023 era insignificante, sólo bastó con limitar la intensidad a 1023; mientras que en otros casos fue necesario descartar el bit menos significativo y luego, si aun existían píxeles que excedían este valor, limitar a 1023 .

En la Tabla 5.2 se muestra la información de los productos utilizados de cada satélite y la información sobre el recorte que se extrajo de la misma. Dentro de la información del producto pueden verse los nombres de los productos utilizados, su tamaño original en píxeles, M y N , la cantidad de niveles de cuantificación, B , y la banda. Dentro de la información del recorte puede verse: la zona del producto de la que se extrajo el recorte, las características del ajuste de RD y el código que se asignó al recorte (que será utilizado desde aquí en adelante para referirse al mismo). En la columna “Zona de recorte”, x_0 e y_0 indican la esquina superior, h el alto (cantidad de filas) y w en ancho (cantidad de columnas). En la columna “Ajuste de RD”, el símbolo “ \rightarrow ” indica que se descartó el bit menos significativo y la abreviación “*Lim.*” indica que se limitó la intensidad a 1023.

Finalmente en las Tablas 5.3, 5.4, 5.5 y 5.6 se muestran las características de las imágenes que componen los conjuntos de datos de prueba de imágenes PAN de zonas urbanas (P_Urb), PAN de zonas rurales (P_Rur), MS de zonas urbanas (M_Urb) y MS de zonas rurales (M_Rur) respectivamente.

Tabla 5.2 – Información del producto y de los recortes usados como imágenes de prueba.

Información del producto						Información del recorte					
Satélite	Nombre	M	N	B [bpp]	Banda	Zona de recorte				Ajuste de RD	Código
						x_0	y_0	h	w		
SPOT 6	SPOT6_20160421_1339415_BUNDLE.W058S34.L2A.16JR_035x108_32721	72232	23524	12	PAN	38885	7025	12288	12288	→ - <i>Lim.</i>	SPT_P_01
SPOT 6	SPOT6_20151013_1356407_BUNDLE.W064S31.L2A.16JR_061x059_32720	39204	40344	12	PAN	10817	14017	12288	12288	→ - <i>Lim.</i>	SPT_P_02
SPOT 7	SPOT7_20170312_1338039_BUNDLE.W060S35.L2A.16JR_069x106_32721	70484	46040	12	PAN	11729	12425	12288	12288	→ - <i>Lim.</i>	SPT_P_08
SPOT 7	SPOT7_20170312_1338039_BUNDLE.W060S35.L2A.16JR_069x106_32721	17621	11510	12	MS	2933	3107	3072	3072	→ - <i>Lim.</i>	SPT_M_08
SPOT 7	SPOT7_20170312_1338039_BUNDLE.W060S35.L2A.16JR_069x106_32721	70484	46040	12	PAN	17665	31105	12288	12288	→ - <i>Lim.</i>	SPT_P_09
SPOT 7	SPOT7_20170312_1338039_BUNDLE.W060S35.L2A.16JR_069x106_32721	70484	11510	12	MS	4417	7777	3072	3072	→ - <i>Lim.</i>	SPT_M_09
OrbView-3	3V040730P0000431131A520000100252M_001648913	25168	8016	11	PAN	1	1	15504	8016	<i>Lim</i>	OV3_P_03
OrbView-3	3V060203P0001075421A520013901622M_001623760	24576	8016	11	PAN	9037	1	15504	8016	<i>Lim</i>	OV3_P_04
OrbView-3	3V050114P0000593421A520007601012M_001643423	25816	8016	11	PAN	1	1	18624	8016	<i>Lim</i>	OV3_P_06
OrbView-3	3V060621P0001256951A520002800552M_001609655	28432	8016	11	PAN	1	1	18624	8016	<i>Lim</i>	OV3_P_07
OrbView-3	3V060507M0001199331A520002300292M_001658348	6786	2004	11	MS	1	1	4656	2004	<i>Lim</i>	OV3_M_08
OrbView-3	3V060320M0001135731A520004300492M_001659021	7168	2004	11	MS	1	1	4656	2004	<i>Lim</i>	OV3_M_09
EROS-B	1A-MBT1-e2335977	8593	10148	10	PAN	1	1	8576	10144		ERB_P_01
EROS-B	1A-MBT1-e2335978	8850	101448	10	PAN	1	1	8576	10144		ERB_P_02
GeoEye-1	13JUL31084528-M2AS-053413654010_01_P001	3633	3616	11	MS	90	110	3432	3432	→	GE1_M_01
QuickBird-2	13SEP18045419-M2AS-053398467010_01_P001	2970	2955	11	MS	21	19	2912	2928	<i>Lim</i>	QB2_M_01
WorldView-3	14NOV27004452-M2AS-054191978040_01_P001	4277	4259	11	MS	103	101	4056	4080	→	WV3_M_01

Tabla 5.3 – Imágenes de prueba PAN de zonas urbanas (*P_Urb*).

Código	M	N	Zona	NIIRS _E
SPT_P_01	12288	12288	Ciudad Autónoma de Buenos Aires	2,55
SPT_P_02	12288	12288	Córdoba (ciudad)	2,57
OV3_P_03	15504	8016	Salta (ciudad)	3,85
OV3_P_04	15504	8016	Mar del Plata	3,92
ERB_P_01	8576	10144	El Palomar	4,83
ERB_P_02	8576	10144	Morón	4,77

Tabla 5.5 – Imágenes de prueba MS de zonas urbanas (*M_Urb*).

Código	M	N	Zona	NIIRS _E
GE1_M_01	3432	3432	Ciudad del Cabo, Sudáfrica	2,97
QB2_M_01	2912	2928	Yaipur, India	3,01
WV3_M_01	4056	4080	Adelaide, Australia	3,89

Tabla 5.4 – Imágenes de prueba PAN de zonas rurales (*P_Rur*).

Código	M	N	Zona	NIIRS _E
SPT_P_08	12288	12288	Buenos Aires - Centro	2,49
SPT_P_09	12288	12288	Buenos Aires - Centro	2,43
OV3_P_06	18624	8016	Córdoba - Centro	3,78
OV3_P_07	18624	8016	Buenos Aires - Sur	3,67

Tabla 5.6 – Imágenes de prueba MS de zonas rurales (*M_Rur*).

Código	M	N	Zona	NIIRS _E
SPT_M_08	3072	3072	Buenos Aires - Centro	0,52
SPT_M_09	3072	3072	Buenos Aires - Centro	0,51
OV3_M_08	4656	2004	Córdoba - Noroeste	1,95
OV3_M_09	4656	2004	Córdoba - Este	1,89

5.2. Esquema de simulación

Los cuatro sistemas de compresión preseleccionados, JPEG-LS, JPEG-Baselines, JPEG2000 y CCSDS-IDC se implementaron en MATLAB haciendo uso de códigos existentes en C/C++ que incluyen tanto el algoritmo de compresión como el de descompresión. Éstos códigos fuentes fueron compilados con el compilador GCC 4.8.4 para generar los ejecutables.

En la Fig. 5.1 se esquematiza como se realizaron las simulaciones para evaluar el desempeño de los sistemas de compresión. Dicho esquema fue implementado en MATLAB y desde este entorno se invocan los ejecutables de los algoritmos de compresión y descompresión. Cada caso de simulación queda definido por: el conjunto de imágenes de prueba a utilizar, el sistema de compresión a evaluar, uno o más parámetros de configuración del sistema de compresión que se mantienen a un valor constante y un parámetro que se hace variar para obtener el desempeño del sistema de compresión en función de éste. En dicha figura, el conjunto de J imágenes se representa por I y está compuesto por $I = \{I_1, I_2, \dots, I_J\}$ y el conjunto de los K valores que puede tomar el parámetro de configuración variable se representa por P y está compuesto por $P = \{P_1, P_2, \dots, P_K\}$.

En la figura puede verse que existe un primer bucle que recorre cada una de las imágenes de prueba y un segundo bucle, interno al anterior, en el que para cada imagen se recorren los todos los valores de P . Una vez seleccionada la imagen I_j y el parámetro de configuración variable P_k , el sistema compresión se configura tomando como entrada P_k y los parámetros de configuración constantes. Luego, la imagen se comprime y descomprime, estos procesos dan como resultado la imagen comprimida y recuperada respectivamente que son utilizadas para el cálculo de CR, dado por (1.9), y las métricas de calidad de imagen.

Respecto a las métricas de calidad de imagen, se calcula el $\Delta NIIRS$, $\Delta NIIRS_{RER}$, $\Delta NIIRS_{G/SNR}$ y $\Delta NIIRS_H$ (acorde al algoritmo de la Fig. 4.25); el SSIM, la MSSIM y todas las métricas basadas en el error píxel a píxel mencionadas en la Sección 3.1.1. No obstante, aquí sólo se muestran los resultados de SNR_C , MSSIM, $\Delta NIIRS$, $\Delta NIIRS_{RER}$, $\Delta NIIRS_{G/SNR}$ y $\Delta NIIRS_H$.

Durante el proceso de compresión se mide el tiempo total que se invierte en comprimir la imagen, $T_{T.Comp.}$, y más adelante se calcula el Tiempo de ejecución medio Por Píxel (TPP), que está dado por el cociente entre el tiempo total y la

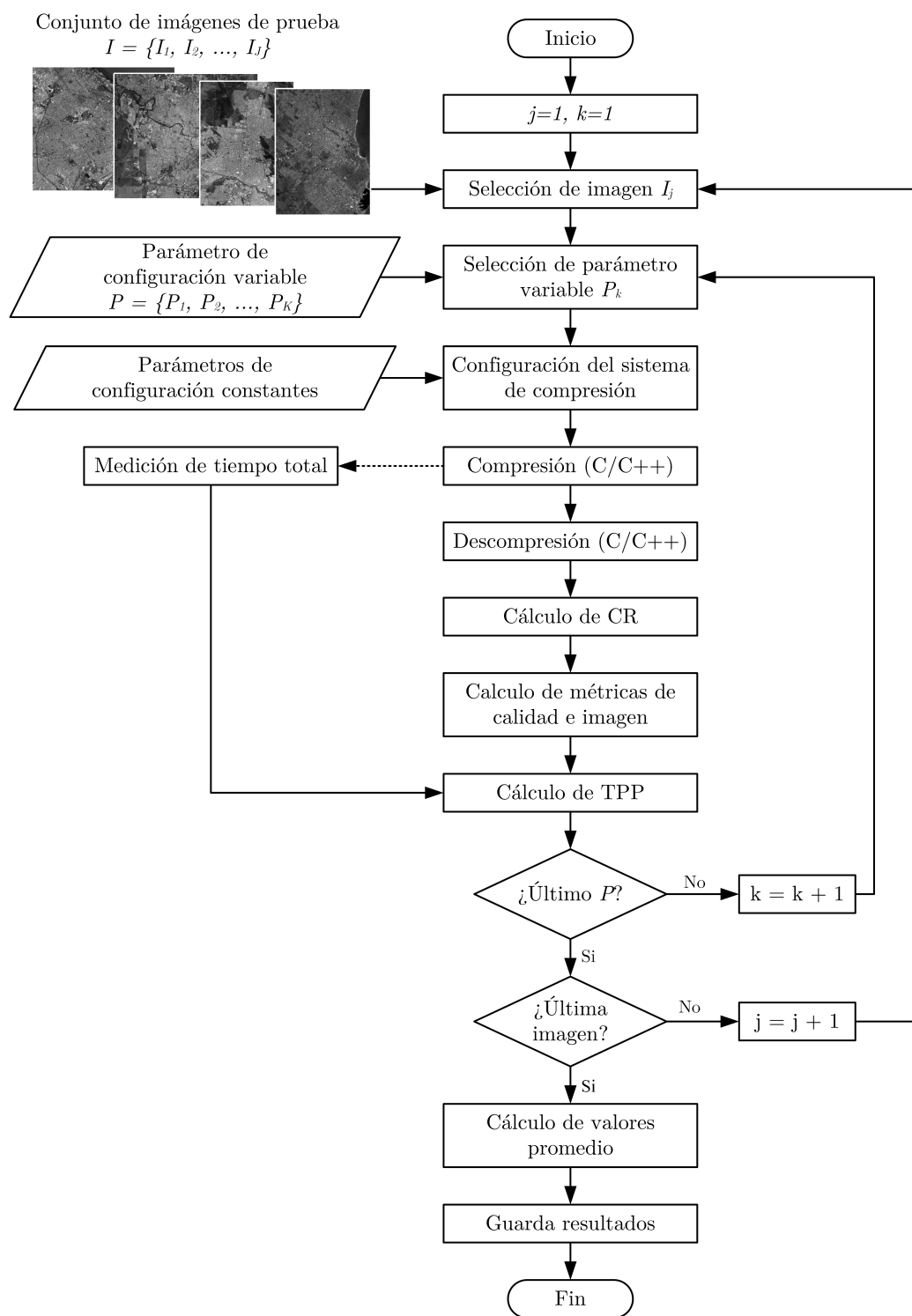


Fig. 5.1. Diagrama de flujo para la evaluación de sistemas de compresión.

cantidad total de píxeles de la imagen, es decir:

$$TPP = \frac{T_{T.Comp.}}{M \cdot N \cdot K} \quad (5.1)$$

Al finalizar ambos ciclos, se tiene un valor de cada una de las métricas (CR, TPP y las de calidad de imagen) para cada una de las imágenes I_j comprimidas con cada uno de los valores de P_k . Para simplificar futuros análisis, para cada valor de P_k , se obtiene el valor promedio del resultado de cada métrica para las imágenes del conjunto de prueba. Dicho proceso se ejemplifica en la Tabla 5.7 para SNR_C .

Tabla 5.7 – Cálculo de media de la métrica SNR_C para cada valor del parámetro de configuración variable, P_k .

$P_k \backslash I_j$	I_1	I_2	\dots	I_J	Media
P_1	$SNR_{C\ 1,1}$	$SNR_{C\ 1,2}$	\dots	$SNR_{C\ 1,J}$	$\overline{SNR_{C\ 1}}$
P_2	$SNR_{C\ 2,1}$	$SNR_{C\ 2,2}$	\dots	$SNR_{C\ 2,J}$	$\overline{SNR_{C\ 2}}$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
P_K	$SNR_{C\ K,1}$	$SNR_{C\ K,2}$	\dots	$SNR_{C\ K,J}$	$\overline{SNR_{C\ K}}$

Finalmente se almacenan los resultados que muestran el desempeño del sistema de compresión para cada una de las imágenes, y el valor promedio, en función del parámetro de configuración variable.

5.3. Criterios para la evaluación

Como ya se mencionó, para la selección del sistema de compresión se desea encontrar el sistema que presenta una relación de compromiso entre tres variables: CR, la complejidad del algoritmo y la degradación de la calidad de imagen. A continuación se presentan algunos criterios que se adoptaron en esta tesis para realizar la comparación de los sistemas de compresión.

- 1) Se comparó el desempeño de los algoritmos a tasas que van desde la compresión sin pérdidas hasta $CR = 10$ ya que como se mostró en la Sección 2.2.2, para misiones espaciales, las tasas se encuentran acotadas entre estos valores.
- 2) La complejidad del algoritmo se cuantificó a través del TPP y la cantidad de filas de la imagen que deben almacenarse en el bloque de memoria de entrada del compresor, M_{in} . Esto depende de si el algoritmo procesa la información por área o por franjas y del tamaño de estas últimas. Si bien estos indicadores están alejados de características del hardware y la potencia necesaria, permiten hacer una comparación relativa entre sistemas.

- 3) La degradación de la calidad de imagen se evaluó basándose el valor de $\Delta NIIRS$, SNR_C y $MSSIM$; cada una de estas métricas indica distintas características de la imagen que se ven afectadas, interpretabilidad, error cuadrático medio y estructura percibida por el sistema de visión humano, respectivamente.
- 4) De manera complementaria se realizó una análisis visual de las imágenes comprimidas para tomar conocimiento del estado general de las mismas y observar si existen distorsiones notables.
- 5) Para cada sistema de compresión se buscaron los parámetros de configuración que optimizan su desempeño haciendo uso del conjunto de imágenes P_Urb y P_Rur y por último evaluó el comportamiento para todos los conjuntos de datos.

5.4. Evaluación de JPEG-LS

5.4.1. Sobre la implementación de JPEG-LS

Para el algoritmo JPEG-LS se utilizaron los códigos fuentes en C/C++ desarrollados por HP Labs que pueden encontrarse en [91]. La implementación soporta tanto compresión sin pérdidas como con pérdidas y hay versiones para imágenes de 8 y 16bpp como máximo. En esta tesis se utilizaron los códigos de la versión para 16bpp como máximo ya que todas las imágenes de prueba utilizadas son de 10bpp.

Como parámetros de configuración la implementación acepta: un vector de 3 elementos con los niveles umbrales para la cuantificación de gradientes, $[T_1 T_2 T_3]$, *RESET* y el parámetro de control de calidad *NEAR*.

5.4.2. Casos de prueba para JPEG-LS

En la Tabla 5.8 se muestran los casos de prueba para el algoritmo JPEG-LS. Soló se evaluó una configuración donde se mantuvieron en sus valores por defecto $[T_1 T_2 T_3]$ y *RESET*, y se hizo variar el parámetro de control de calidad para los valores $NEAR = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32\}$, donde $NEAR = 0$ representa el caso sin pérdidas. En total se tienen cuatro casos de prueba, uno para cada conjunto de imágenes.

Aunque el vector $[T_1 T_2 T_3]$ forma parte de los parámetros de configuración que se consideran fijos, ya que siempre toma el valor por defecto, este dependen de

$NEAR$ y de $MAXVAL$, y se calculó con una función desarrollada en MATLAB acorde a lo indicado en el estándar [51]. En la Tabla 5.9 se muestran los valores de $[T_1 T_2 T_3]$ utilizados según el valor de $NEAR$, para $MAXVAL = 1023$ ya que todas las imágenes de prueba están cuantificadas con $B = 10bpp$.

Tabla 5.8 – *Casos de prueba para JPEG-LS.*

Nombre	$[T_1 T_2 T_3]$	Reset	NEAR	Conjunto de imágenes de prueba
JLS_01	Ver Tabla 5.9 ^a	64 ^a	Variable	P_Urb
JLS_21	Ver Tabla 5.9 ^a	64 ^a	Variable	P_Rur
JLS_31	Ver Tabla 5.9 ^a	64 ^a	Variable	M_Urb
JLS_41	Ver Tabla 5.9 ^a	64 ^a	Variable	M_Rur

^a Valor por defecto del algoritmo.

Tabla 5.9 – *Valores por defecto de $[T_1 T_2 T_3]$ para cada valor de $NEAR$ utilizado, con $MAXVAL = 1023$.*

$NEAR$	0	1	2	3	4	5	6	8	10	12	14	16	20	24	28	32
T_1	6	9	12	15	18	21	24	30	36	42	48	54	66	78	90	102
T_2	19	24	29	34	39	44	49	59	69	79	89	99	119	139	159	179
T_3	72	79	86	93	100	107	114	128	142	156	170	184	212	240	268	296

5.4.3. Resultados de JPEG-LS

En esta sección se presenta un resumen de los resultados mientras que en el Anexo G se muestran todos los resultados obtenidos para todos los casos de prueba de JPEG-LS.

Respecto a la tasa de compresión, en la Fig. 5.2(a) puede verse que para un dado valor de $NEAR$ el valor de CR depende de las características de la imagen. Conforme aumenta $NEAR$, se obtienen mayores tasas y también aumenta la dispersión de los valores de CR obtenidos para un dado $NEAR$. No obstante a cada valor de $NEAR$ puede asociarse un valor de tasa de compresión media, \overline{CR} , que define el comportamiento del sistema de compresión.

Este sistema de compresión fue desarrollado para tasas de compresión bajas, como puede verse debieron utilizarse valores de $NEAR$ cercanos a 32 para obtener tasas próximas a 10 para el conjunto P_Urb. Grandes valores de $NEAR$ producen imágenes donde muchas zonas son codificadas en el modo RLE y se pierde el detalle en zonas bajo contraste. Incluso, para las imágenes rurales, no se pudo calcular el $\Delta NIIRS$ para $NEAR > 24$ ya que algunos bordes de bajo rango dinámico fueron codificados como píxeles de igual nivel de gris. Más detalles sobre esta distorsión se muestran en la Sección 5.8.

Respecto a la complejidad, este sistema de compresión sólo necesita mantener en memoria una cantidad de píxeles equivalente a una fila de la imagen (ver Sección 2.4). Por otro lado, en la Fig. 5.2(b) puede verse que el TPP está en el orden de los $60ns$ y que disminuye conforme aumenta $NEAR$ ya que mayor cantidad de píxeles se codifican en el modo RLE. Para las imágenes rurales, donde existen grandes áreas casi uniformes, TPP es menor por el mismo motivo.

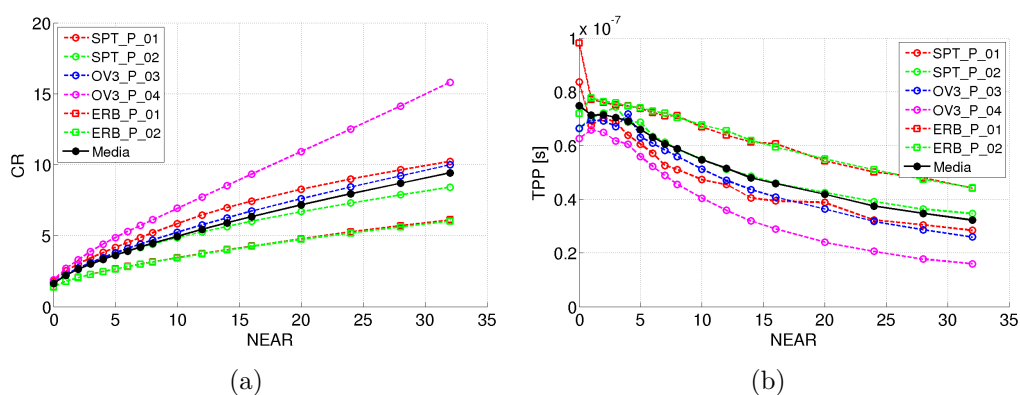


Fig. 5.2. Resultados de JPEG-LS con P_{Urb} . (a) CR y (b) TPP vs. NEAR.

En lo que hace a la calidad de imagen, para los valores de $NEAR$ ensayados, el sistema de compresión no degrada el NIIRS en una cantidad mayor al valor mínimo visualmente perceptible (0,1), como se muestra en la Fig. 5.3 para el conjunto P_{Urb} . Ésto se debe a que el predictor del sistema de compresión detecta los bordes y no se degradan, por lo tanto RER no se ve afectado, mientras que H y SNR tienen efectos despreciables.

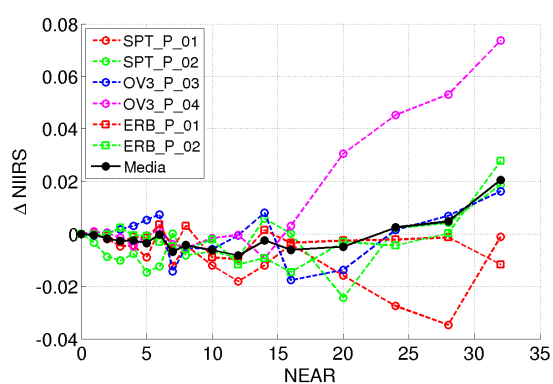


Fig. 5.3. Resultados de JPEG-LS con P_{Urb} . $\Delta NIIRS$ vs. NEAR.

En la Fig. 5.4(a) se puede ver SNR_C y en la Fig. 5.4(b) MSSIM en función de $NEAR$. En ambos casos se aprecia que todas las imágenes presentan diferentes degradaciones de calidad.

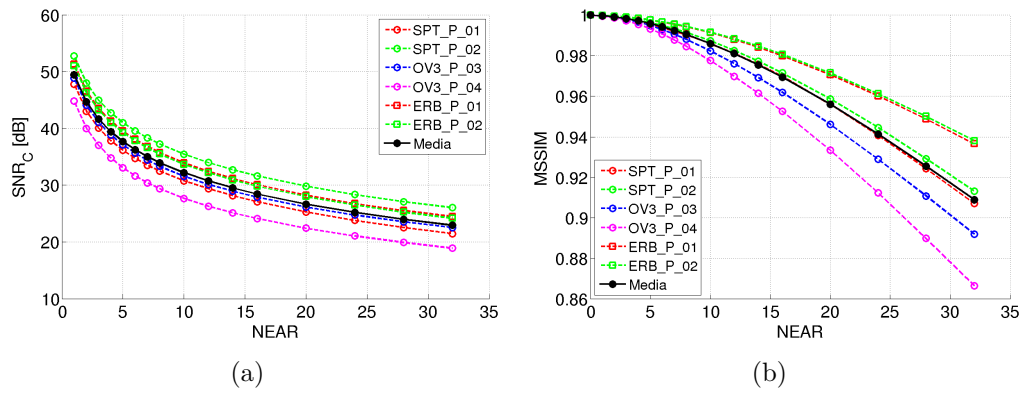


Fig. 5.4. Resultados de JPEG-LS con P_{Urb} . (a) SNR_C y (b) $MSSIM$ vs. $NEAR$.

5.5. Evaluación de JPEG-Baselines

5.5.1. Sobre la implementación de JPEG-Baselines

Para el algoritmo JPEG-Baselines se utilizaron los códigos fuentes en C/C++ desarrollados por el *Independent JPEG Group* (IJG) que se encuentran disponibles en [92]. Dicha implementación puede ser configurada para operar con imágenes de 8, 10 y 12**bpp**. En esta tesis se compilaron los códigos para operar con imágenes de 10**bpp**.

Como ya se mencionó en la Sección 2.5, las diferentes calidades de imagen se obtienen escalando la tabla de cuantificación. La forma de hacerlo no está especificada en el estándar y distintas implementaciones podrían utilizar diferentes mecanismos. IJG escala la tabla de cuantificación de acuerdo a un parámetro denominado *Quality* que va entre 100 (mejor calidad) y 1 (peor calidad). A continuación se detalla como se realiza el escalado.

Dada una tabla de cuantificación básica, como por ejemplo la que da el estándar por defecto,

$$Q_b(x, y) = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}, \quad (5.2)$$

ésta resulta escalada por un factor F que depende de $Quality$ de acuerdo a

$$F(Quality) = \begin{cases} 5000/Quality & \text{para } 1 \leq Quality \leq 49, \\ 200 - 2 \cdot Quality & \text{para } 50 \leq Quality \leq 100. \end{cases} \quad (5.3)$$

Conforme $Quality$ va de 100 a 1, F va de 0 a 5000, como se ilustra en la Fig. 5.5, para calidades que van desde la mejor hasta la peor respectivamente. La tabla de cuantificación final, $Q_f(x, y)$, que es utilizada para dividir los coeficientes de la DCT, está dada por

$$Q'_b(x, y) = \text{round}\left(Q_b(x, y) \times \frac{F}{100}\right) \quad (5.4)$$

$$Q_f(x, y) = \begin{cases} 1 & \text{si } Q'_b(x, y) < 1, \\ 2^{16} - 1 & \text{si } Q'_b(x, y) > 2^{16} - 1, \\ Q'_b(x, y) & \text{de otra manera.} \end{cases} \quad (5.5)$$

A modo de ejemplo notar que:

- cuando $Quality = 100$, $F = 0$ y $Q_f(x, y) = 1 \forall(x, y)$,
- cuando $Quality = 50$, $F = 100$ y $Q_f(x, y) = Q_b(x, y)$,

Mientras más pequeño sea $Quality$, mayor será F , mayores los elementos de la tabla de cuantificación final y por lo tanto, los coeficientes de la DCT serán cuantificados de manera más grosera y se tendrá una mayor tasa de compresión con una calidad de imagen menor.

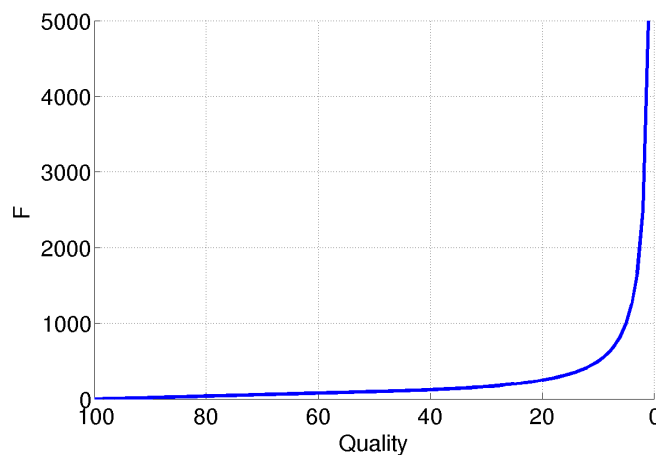


Fig. 5.5. Gráfico de $F(Quality)$.

La implementación recibe como parámetros de configuración el valor de $Quality$ y la tabla de cuantificación básica.

5.5.2. Casos de prueba para JPEG-Baselines

En la Tabla 5.10 se muestran los casos de prueba para el algoritmo JPEG-Baselines. Con los primeros seis casos de prueba (tres con P_Urb y tres con P_Rur) se desea determinar cuál es la tabla de cuantificación básica más conveniente entre las tres siguientes:

- 1) **Default**: tabla por defecto del algoritmo, (5.2),
- 2) **GWG**: tabla sugerida por el Geospatial intelligence standards Working Group (GWG),

$$Q_b(x, y) = \begin{bmatrix} 13 & 13 & 15 & 25 & 40 & 68 & 114 & 161 \\ 13 & 16 & 22 & 32 & 49 & 79 & 131 & 200 \\ 16 & 22 & 38 & 54 & 78 & 117 & 198 & 245 \\ 26 & 33 & 56 & 93 & 122 & 206 & 235 & 255 \\ 45 & 54 & 84 & 139 & 205 & 230 & 255 & 255 \\ 65 & 92 & 134 & 207 & 235 & 255 & 255 & 255 \\ 112 & 165 & 198 & 245 & 255 & 255 & 255 & 255 \\ 190 & 205 & 250 & 255 & 255 & 255 & 255 & 255 \end{bmatrix} \quad (5.6)$$

- 3) **R=1**: tabla de acuerdo a la expresión (2.16) con $R = 1$,

$$Q_b(x, y) = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{bmatrix} \quad (5.7)$$

Una vez determinada la tabla de cuantificación básica, los casos JPG_31 y JPG_41 tienen el objeto de mostrar el desempeño para los conjuntos de datos de prueba restantes con la configuración adoptada.

En todos los casos se se hizo variar el parámetro de control de calidad para los valores $Quality = \{100, 98, 96, 94, 92, 90, 88, 86, 84, 82, 80, 75, 70, 65, 60, 55, 50\}$.

Tabla 5.10 – *Casos de prueba para JPEG-Baselines.*

Caso de prueba	Tabla de cuantificación	<i>Quality</i>	Conjunto de imágenes de prueba
JPG_01	<i>Default</i>	Variable	P_Urb
JPG_02	CWG	Variable	P_Urb
JPG_03	$R = 1$	Variable	P_Urb
JPG_21	<i>Default</i>	Variable	P_Rur
JPG_22	CWG	Variable	P_Rur
JPG_23	$R = 1$	Variable	P_Rur
JPG_31	$R = 1$	Variable	M_Urb
JPG_41	$R = 1$	Variable	M_Rur

5.5.3. Resultados de JPEG-Baselines

En esta sección se presenta un resumen de los resultados mientras que en el Anexo H se muestran todos los resultados obtenidos para todos los casos de prueba de JPEG-Baselines.

Selección de tabla de cuantificación

En la Fig. 5.6 se muestran los resultados de CR , SNR_C , MSSIM y $\Delta NIIRS$ en función de *Quality* para las tres tablas de cuantificación básicas evaluadas, donde el valor para cada punto es el promedio del resultado obtenido para cada una de las 6 imágenes que forman el conjunto de datos de prueba P_Urb. Para las tablas de cuantificación *Default* y GWG se obtienen CR demasiado grande respecto los valores deseados ($CR < 10$) con un una distorsión grande, mientras que para la tabla $R = 1$ se obtienen imágenes con menor degradación y con tasas de compresión cercanas a las deseadas. Para la tabla $R = 1$, analizando curva de SNR_C , puede verse que la calidad se mantiene constante desde $Quality = 100$ hasta $Quality = 96$. Esto se debe a que para $Quality = 100 \Rightarrow F = 0$ y por lo tanto $Q_f(x, y) = 1 \forall (x, y)$, condición que se mantiene hasta $Quality = 96 \Rightarrow F = 8$ debido a los valores de los elementos de esta tabla. Los resultados con las imágenes de zonas rurales conducen a las mismas conclusiones y pueden verse en el Anexo H.

Configuración adoptada

En virtud de lo antes mencionado, la tabla de $R = 1$ se considera la más apta para la CAR. Los casos de prueba JPG_03, JPG_23, JPG_31 y JPG_41 muestran el desempeño del sistema de compresión para cada conjunto de datos de prueba con esta configuración adoptada.

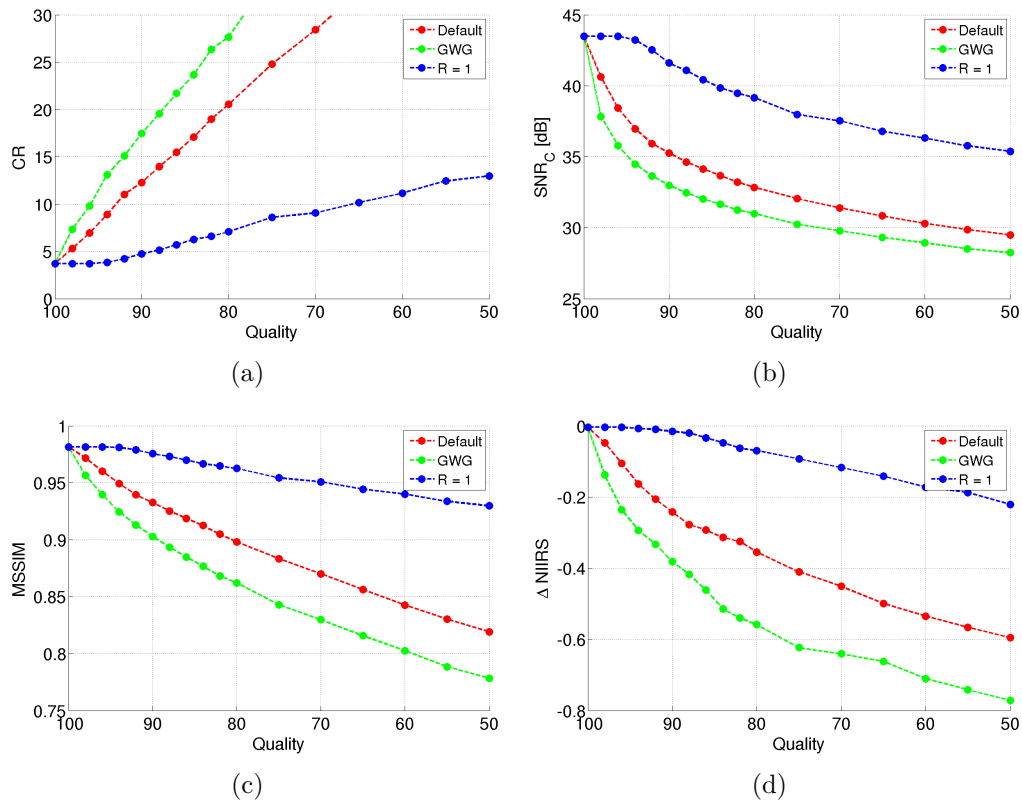


Fig. 5.6. Comparación de tres tablas de cuantificación para JPEG-Baselines con *P-Urb*. Valores promedio de (a) CR, (b) SNR_C , (c) MSSIM y (d) $\Delta NIIRS$ vs. *Quality*.

En lo que refiere a la tasa de compresión, en la Fig. 5.7(a) puede verse que para un dado valor de *Quality* el valor de CR depende de las características de la imagen. Conforme *Quality* es menor, y los coeficientes de la DCT se cuantifican de manera más grosera, CR aumenta como así también la dispersión alrededor del valor medio. No obstante a cada valor de *Quality* puede asociarse un valor de tasa de compresión media, \overline{CR} , que define el comportamiento del sistema de compresión.

Respecto a la complejidad, considerando que el compresor realiza la DCT de bloques de 8×8 píxeles, sólo es necesario almacenar 8 filas de la imagen (ver Sección 2.5). Por otro lado en la Fig. 5.7(b) puede verse que el TPP está en el orden de los $50ns$ y que disminuye conforme aumenta *Quality* ya que existe una mayor cantidad de coeficientes que son truncados a cero y por lo tanto la codificación de entropía se realiza más rápido. Para las imágenes rurales, donde son más ausentes las componentes de alta frecuencia de la DCT, TPP es menor por el mismo motivo.

En cuanto a la calidad de imagen, en la Fig. 5.8(d) se puede ver la degradación total del NIIRS y en las Figs. 5.8(a), 5.8(b) y 5.8(c) la contribución

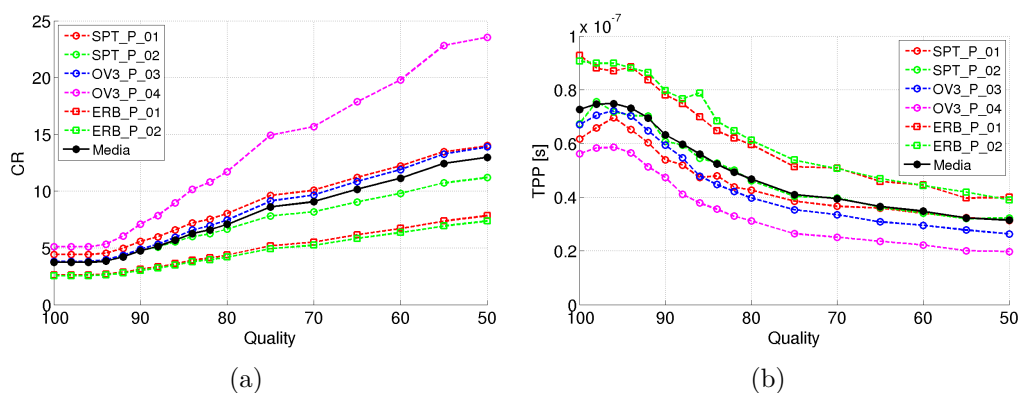


Fig. 5.7. Resultados de JPEG-Baselines con P-Urb. (a) CR y (b) TPP vs. Quality.

de cada uno de los términos de la GIQE que se ve modificado por la compresión, donde el principal impacto es causado por la degradación de RER. El resto de los términos también presentan características decrecientes pero sus valores están un orden de magnitud por debajo. Además, puede verse que para $Quality \geq 84$ el $NIIRS$ no se degrada más 0,1 para todas las imágenes del conjunto P-Urb y CR toma un valor promedio de 6,3. No obstante, es común que para estas tasas de compresión se presenten grandes distorsiones visibles, lo cual se verifica en la Sección 5.8.

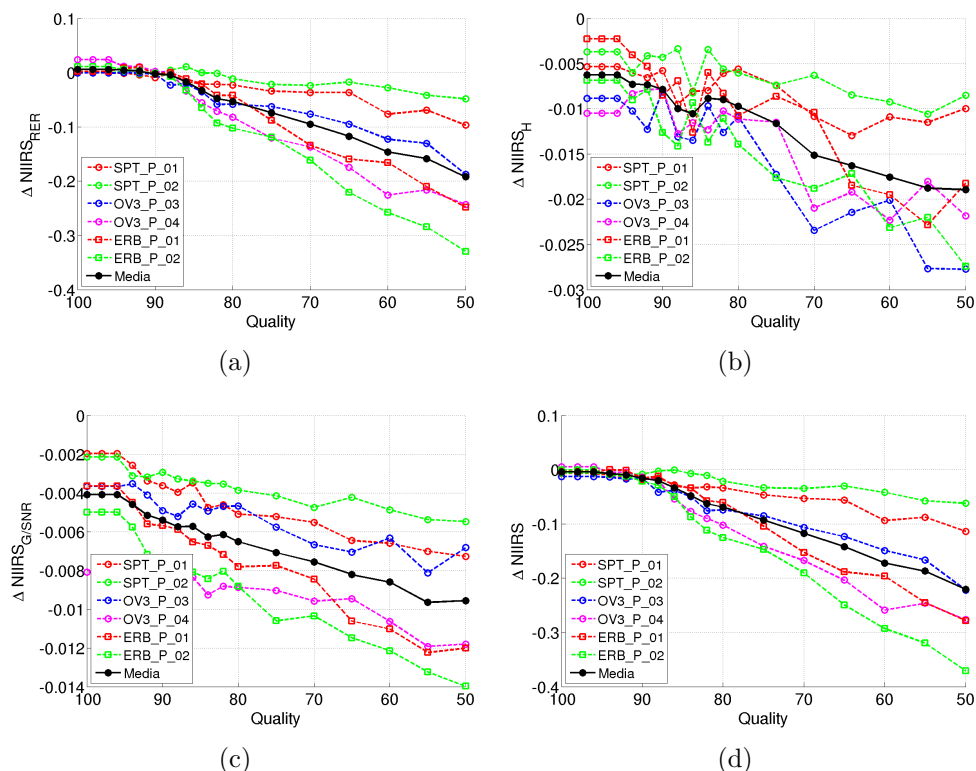


Fig. 5.8. Resultados de JPEG-Baselines con P-Urb. (a) $\Delta NIIRS_{RER}$, (b) $\Delta NIIRSH$, (c) $\Delta NIIRSG/SNR$ y (d) $\Delta NIIRS$ vs. Quality.

En la Fig. 5.9 se ven los resultados de SNR_C y MSSIM en función de $Quality$ para el conjunto P_Urb, que son utilizados para comparar el desempeño de este sistema de compresión con los restantes en la sección final de este capítulo. Puede apreciarse, que para un dado valor de $Quality$ la calidad depende de las características de la imagen.

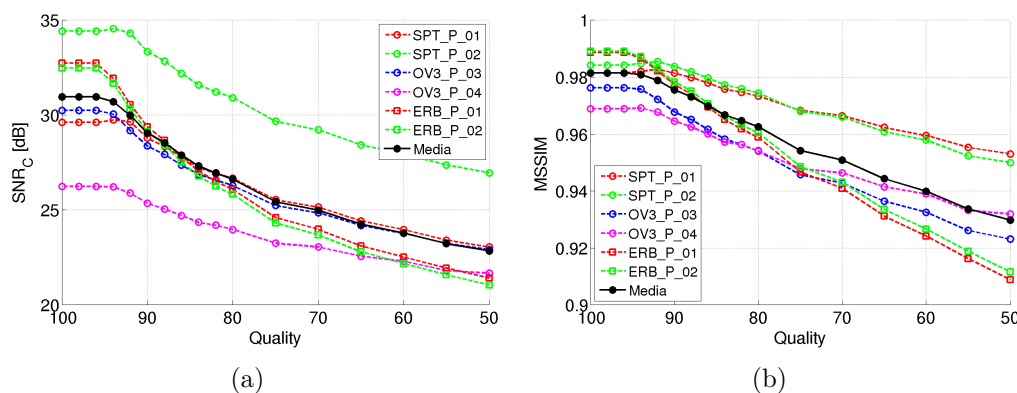


Fig. 5.9. Resultados de JPEG-Baselines con P_Urb. (a) SNR_C y (b) MSSIM vs. $Quality$.

5.6. Evaluación de JPEG2000

5.6.1. Sobre la implementación de JPEG2000

Para el algoritmo JPEG2000 se utilizaron los códigos fuentes en C/C++ desarrollados por el Grupo de Procesamiento Digital de Señales e Imágenes (ISPGGroup, *Image and Signal Processing Group*) de la Universidad Católica de Lovain con aportes de la CNES que están incluidos en la lista de software de referencia del JPEG y se encuentran disponibles en [93].

Como parámetros de configuración, la implementación recibe: el tamaño de las teselas, el tipo transformaciones a utilizar (reversible o irreversible), la cantidad de veces que se aplica la DWT 2D, el tamaño de los bloques de código, el tamaño de los recintos y la tasa de compresión. El tipo de transformación hace referencia tanto a la transformación de color como a la DWT. Si se seleccionan transformaciones reversibles se utiliza la DWT entera y si se seleccionan transformaciones irreversibles se utiliza la DWT flotante. Como en esta tesis no se aplicó transformación de color, sólo se hace referencia a tipo de DWT.

5.6.2. Casos de prueba para JPEG2000

En la Tabla 5.11 se muestran los casos de prueba para el algoritmo JPEG2000. En primer lugar, los casos de prueba JP2_01 y JP2_02 brindan información para comparar el desempeño de los dos tipos de DWT. Una vez seleccionado el tipo, se realizaron simulaciones para determinar la cantidad de veces que es necesario aplicar la DWT 2D, N_{DWT} . Este análisis se realizó tanto con imágenes de zonas urbanas, donde predominan las componentes de alta frecuencia (casos de prueba JP2_03 y JP2_04), como con imágenes de zonas rurales, donde predominan las componentes de baja frecuencia (JP2_23 y JP2_24). En éstos casos se hizo variar N_{DWT} mientras que el resto de los parámetros de configuración se mantuvo constante y se evaluó para dos situaciones extremas, compresión sin pérdidas (simbolizado por $CR = 0$) y con pérdidas con $CR = 10$.

Luego se realizaron simulaciones para determinar el tamaño óptimo de la tesela donde su ancho se ajustó al ancho de la imagen y se realizó un barrido para la cantidad de filas de la tesela, M_T . Este análisis se llevó a cabo tanto para imágenes de zonas urbanas (JP2_05 y JP2_06) como para imágenes de zonas rurales (JP2_25 y JP2_26). En éstos casos se hizo variar M_T mientras que el resto de los parámetros de configuración se mantuvo constante y se evaluó para las dos situaciones extremas antes mencionadas.

Los casos de prueba JP2_07, JP2_08 y JP2_09 muestran el desempeño del compresor en función de CR con tres tamaños de tesela diferentes, área, 512 filas y 32 filas, para el conjunto de datos P_Urb.

Todos los casos de prueba antes mencionados permitieron encontrar la mejor configuración para el compresor. Finalmente los casos de prueba JP2_08, JP2_21, JP2_31 y JP2_41 muestran el desempeño de la configuración adoptada para cada uno de los conjuntos de datos de prueba.

En los casos en que el parámetro de configuración variable es la tasa de compresión se hizo un barrido para los valores $CR = \{0; 2,5; 3; 3,5; 4; 4,5; 5; 6; 8; 10\}$, donde $CR = 0$ representa el caso sin pérdidas. Los barridos para la cantidad de DWT 2D se realizaron para los valores $N_{DWT} = \{1, 2, \dots, 7, 8\}$. Los barridos para la cantidad de filas de las teselas se realizaron para los valores $M_T = \{16, 32, 64, 128, 256, 512, 1024, 2048; M\}$, donde 16 es el valor mínimo debido a la cantidad de DWT 2D aplicadas y M representa la operación por área.

Tabla 5.11 – Casos de prueba para JPEG2000.

Caso de prueba	Cálculo de DWT	N_{DWT}	M_T	Tamaño de bloques de código	Tamaño de recintos	CR	Conjunto de imágenes de prueba
JP2_01	Entera	5 ^a	Área ^a	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	P_Urb
JP2_02	Flotante	5 ^a	Área ^a	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	P_Urb
JP2_03	Entera	Variable	Área ^a	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	0 ^b	P_Urb
JP2_04	Entera	Variable	Área ^a	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	10	P_Urb
JP2_05	Entera	4	Variable	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	0 ^b	P_Urb
JP2_06	Entera	4	Variable	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	10	P_Urb
JP2_07	Entera	4	Área ^a	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	P_Urb
JP2_08	Entera	4	512	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	P_Urb
JP2_09	Entera	4	32	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	P_Urb
JP2_21	Entera	4	512	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	P_Rur
JP2_23	Entera	Variable	Área ^a	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	0 ^b	P_Rur
JP2_24	Entera	Variable	Área ^a	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	10	P_Rur
JP2_25	Entera	4	Variable	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	0 ^b	P_Rur
JP2_26	Entera	4	Variable	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	10	P_Rur
JP2_31	Entera	4	512	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	M_Urb
JP2_41	Entera	4	512	64×64 ^a	2 ¹⁵ ×2 ¹⁵ ^a	Variable	M_Rur

^a Valor por defecto del algoritmo.

^b Implica compresión sin pérdidas, la tasa de compresión dependerá de la imagen.

5.6.3. Resultados de JPEG2000

En esta sección se presenta un resumen de los resultados mientras que en el Anexo I se muestran todos los resultados obtenidos para todos los casos de prueba de JPEG2000.

Comparación entre DWT entera y DWT flotante

En primer lugar se realizó la comparación entre el cálculo de la DWT en punto fijo y en punto flotante, en la Fig. 5.10 se ilustran los resultados para el conjunto de datos P_Urb. El TPP es en promedio 12 % mayor para la transformada flotante y ésta presenta una degradación mucho mayor en términos de SNR_C y MSSIM. Para $CR < 8$ la DWT entera presenta menor degradación del NIIRS y para $CR > 8$ la DWT flotante pasa a ser la que menos degrada. Debido a una menor complejidad y mayor calidad de imagen, se seleccionó la DWT entera.

Es menester mencionar que el resultado esperado era que la DWT flotante presentara mejor calidad en términos de SNR_C , acorde a los ensayos del JPEG2000 con las imágenes habitualmente utilizadas para evaluar sistemas de compresión [17, 18]. No obstante, como puede verse, al ensayar con imágenes satelitales se obtuvo el resultado opuesto, es decir, la transformación en punto fijo dio mejores resultados.

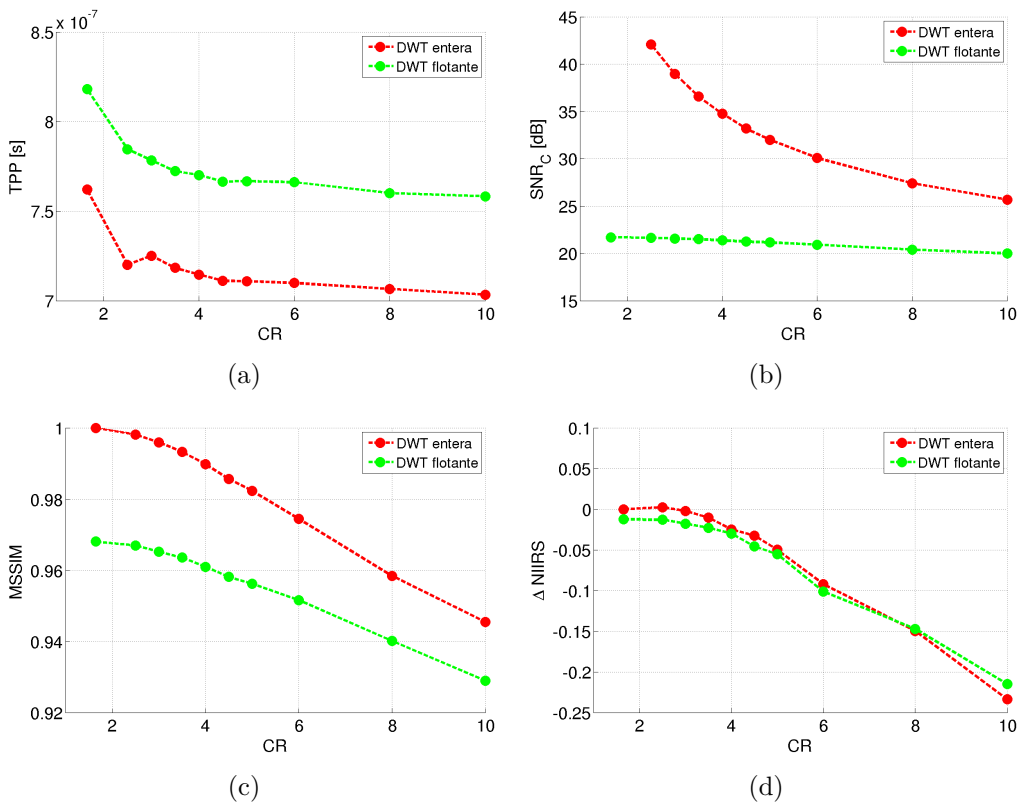


Fig. 5.10. Comparación entre DWT entera y DWT flotante para JPEG2000 con P_{Urb} . Valores promedio de (a) TPP, (b) SNR_C , (c) MSSIM y (d) $\Delta NIIRS$ vs. CR.

Desempeño de acuerdo a la cantidad de DWT 2D

Una vez seleccionado el tipo de DWT se analiza cuantas veces debe realizarse la DWT 2D para obtener un desempeño óptimo. En la Fig. 5.11(a) se ve la tasa de compresión en función de N_{DWT} para el caso sin pérdidas, donde puede apreciarse un leve aumento de CR conforme se aplican más DWT y que para $N_{DWT} > 4$ no hay mejora. En la Fig. 5.11(b) se ve la SNR_C en función de N_{DWT} para un caso con pérdidas con $CR = 10$, donde se aprecia que SNR_C aumenta hasta $N_{DWT} = 4$ y luego se mantiene constante. En ambos casos los valores que se muestran en las figuras son el valor promedio de las 6 imágenes del conjunto P_{Urb} , mientras que los resultados completos puede verse en el Anexo I.

Finalmente, si se observa el TPP en las Figs. 5.12(a) y 5.12(b), para ambos casos se aprecia que a medida que se aplica más veces la DWT el tiempo disminuye, esto se debe a que el bloque que más tiempo requiere no es la DWT sino la codificación y que a medida que se tienen más niveles de descomposición la codificación es más eficiente [18, 94].

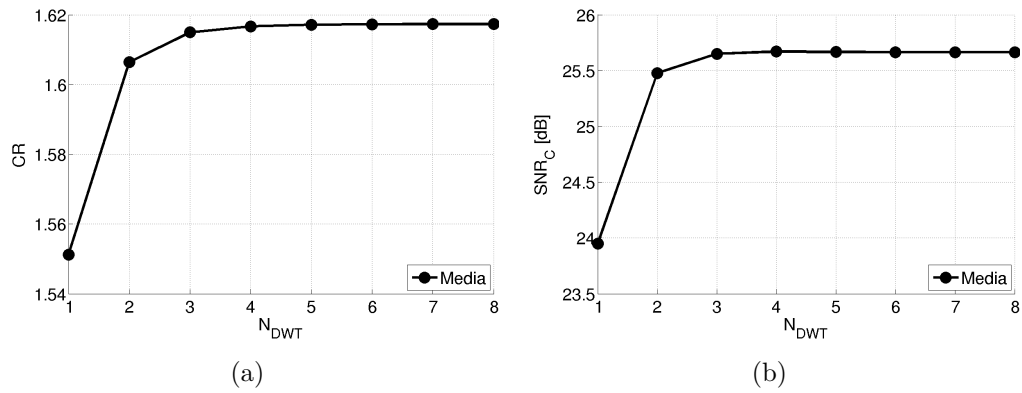


Fig. 5.11. Desempeño JPEG2000 en función de la cantidad de DWT 2D. Valores promedio de (a) CR vs. N_{DWT} para JP2_03 y (b) SNR_C vs. N_{DWT} para JP2_04.

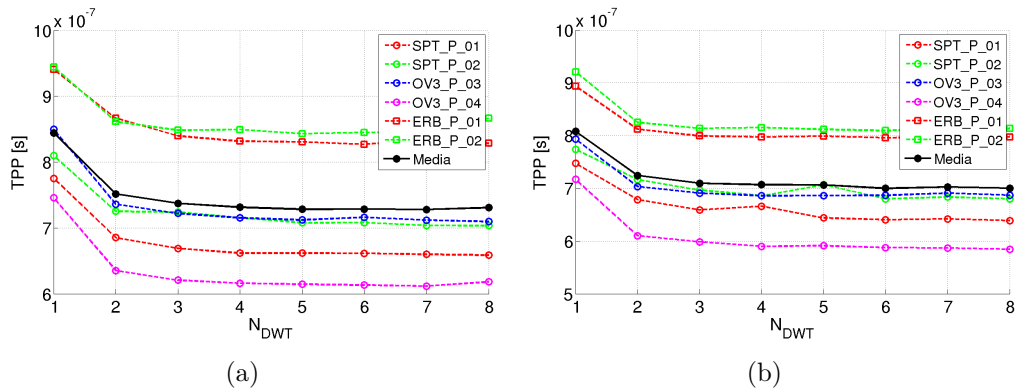


Fig. 5.12. Desempeño JPEG2000 en función de la cantidad de DWT 2D. (a) TPP vs. N_{DWT} para JP2_03 y (b) TPP vs. N_{DWT} para JP2_04.

En el Anexo I, en la Fig. I.19 puede verse que MSSIM sigue la misma ley de variación que SNR_C y en la Fig. I.20 que el $\Delta NIIRS$ no varía de manera significativa y concluyente. También puede verse que con imágenes rurales, casos JP2_23 y JP2_24, se llega a las mismas conclusiones.

En virtud de lo antes mencionados se seleccionó $N_{DWT} = 4$ como un valor óptimo para la presente aplicación.

Desempeño de acuerdo al tamaño de las teselas

En la Fig. 5.13(a) se ve el valor promedio de la tasa de compresión en función de la cantidad de filas de las teselas, M_T , para el caso sin pérdidas, y en la Fig. 5.13(b) se ve la SNR_C en función de M_T para un caso con pérdidas con $CR = 10$. En ambos casos los valores que se muestran en las figuras son el valor promedio de las 6 imágenes del conjunto P_Urb, mientras que los resultados completos puede verse en el Anexo I. Se aprecia que tanto CR como SNR_C aumentan conforme la tesela es mayor y que para $M_T > 512$ la pendiente de

crecimiento se vuelve prácticamente constante. Por lo tanto el valor de $M_T = 512$ se considera el óptimo.

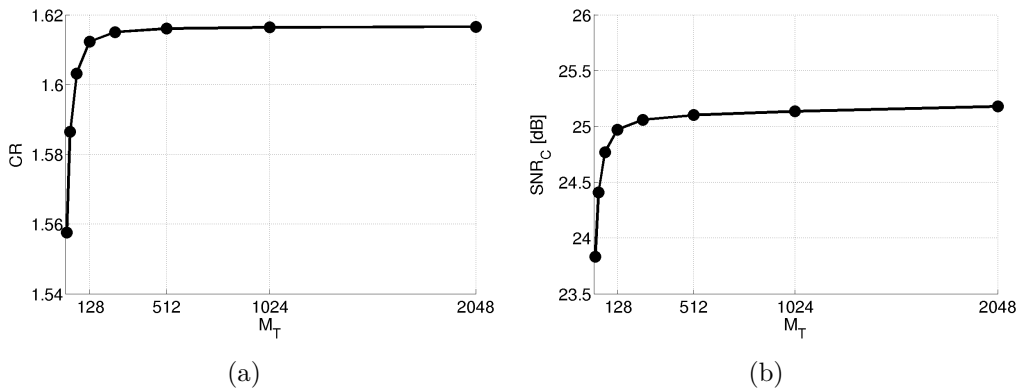


Fig. 5.13. Desempeño JPEG2000 en función del tamaño de las teselas. Valores promedio de (a) CR vs. M_T para JP2_04 y (b) SNR_C vs. M_T para JP2_06.

En el Anexo I, en la Fig. I.25 puede verse que MSSIM sigue la misma ley de variación y en la Fig. I.26 que el $\Delta NIIRS$ no varía de manera significativa y concluyente. También puede verse que con imágenes rurales, casos JP2_25 y JP2_25, se llega a las mismas conclusiones.

En la Fig. 5.14 se ilustra el desempeño del compresor en función de CR para tres tamaños de tesela distintos. Como puede verse $M_T = 512$ representa una situación intermedia en términos de SNR_C . Por otro lado el desempeño observando el $\Delta NIIRS$ es prácticamente idéntico para los tres casos.

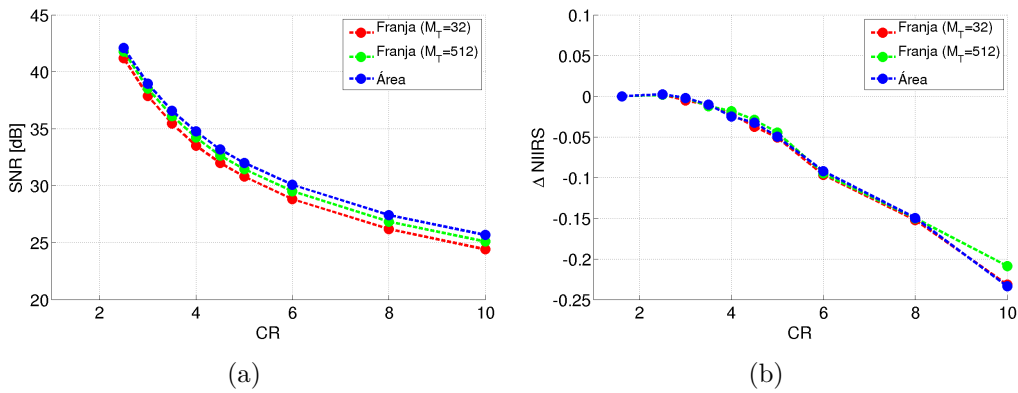


Fig. 5.14. Comparación del desempeño de JPEG2000 según tamaño de tesela en función de CR para P_Urb. Valores promedio de (a) SNR_C y (b) $\Delta NIIRS$ vs. CR.

Configuración adoptada

De acuerdo a lo antes mencionado se adopta la configuración que corresponde a la DWT entera con $N_{DWT} = 4$, $M_T = 512$ y el resto de los

parámetros en su valor por defecto. Los casos JP2_08, JP2_21, JP2_31 y JP2_41 muestran el desempeño con esta configuración en función de CR.

Para JPEG2000, en su modo con pérdidas, la tasa de compresión es constante e igual al valor especificado, independientemente de las características de la imagen, esto presenta una ventaja frente a los dos sistemas antes evaluados ya que se puede conocer exactamente cuanto espacio de memoria ocupará cada imagen.

Respecto a la complejidad, con la configuración adoptada se necesitan almacenar 512 filas de la imagen, no obstante, se ha mostrado que es posible obtener implementaciones que requieren menor cantidad de memoria con comportamientos similares. Por otro lado en la Fig. 5.15(a) puede verse que el TPP está en el orden de los $700ns$ para el conjunto P_Urb.

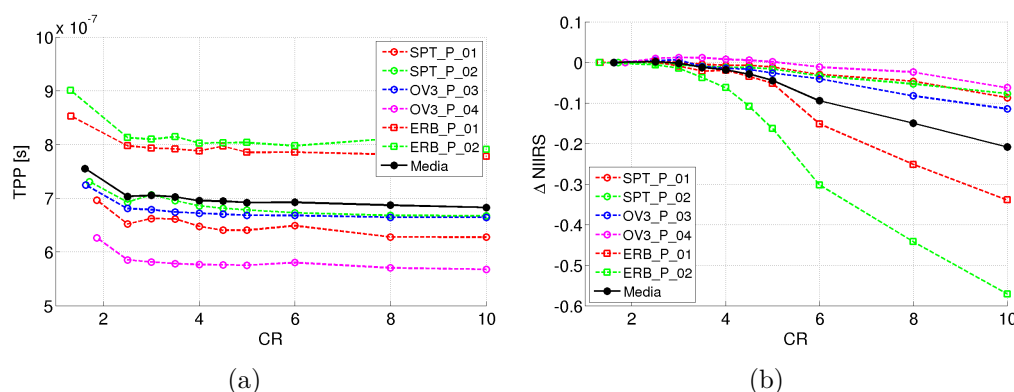


Fig. 5.15. Resultados de JPEG2000 con P_Urb. (a) TPP y (b) $\Delta NIIRS$ vs. CR.

En lo que hace a la calidad de imagen, para $CR \leq 4$ el NIIRS no se degrada más del valor mínimo visualmente perceptible (0,1) para ninguna de las imágenes del conjunto P_Urb, ver Fig. 5.15(b). Al igual que para JPEG-Baselines la degradación del NIIRS se debe principalmente al borrono en la imagen (disminución del RER), como puede verse comparando las Figs. I.37, I.38 y I.39.

En la Fig. 5.16(a) se puede ver SNR_C y en la Fig. 5.16(b) MSSIM en función de CR, donde se aprecia que la degradación de la calidad depende de la imagen comprimida.

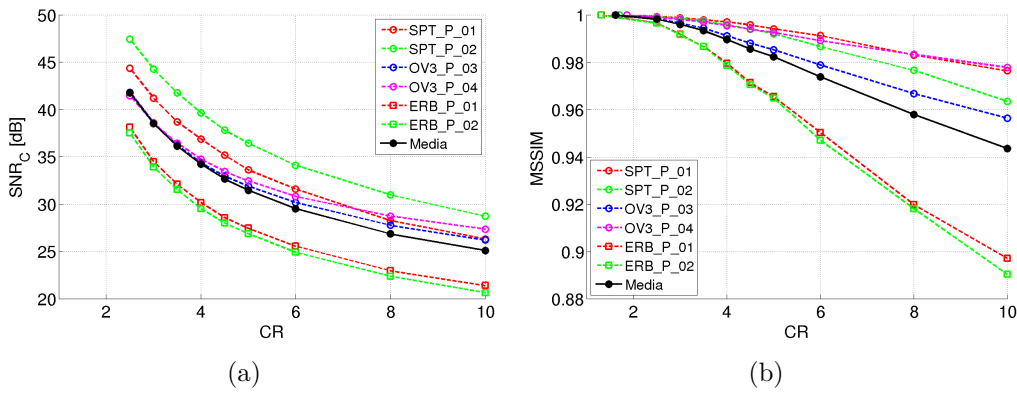


Fig. 5.16. Resultados de JPEG2000 con P_{Urb} . (a) SNR_C y (b) $MSSIM$ vs. CR .

5.7. Evaluación de CCSDS-IDC

5.7.1. Sobre la implementación de CCSDS-IDC

Para el algoritmo CCSDS-IDC se utilizaron los códigos fuentes en C/C++ desarrollados por la Universidad de Nebraska-Lincoln que se encuentran disponibles en [95].

Como parámetros de configuración, la implementación recibe: la cantidad de bloques por segmento, S , el tipo de DWT (entera o flotante) y la tasa de compresión, CR . El estándar diferencia entre la operación por área y por franjas de acuerdo al valor de S , como se mencionó en la Sección 2.7. No obstante, esto sólo tiene impacto en la cantidad de memoria requerida por el bloque de codificación BPE, mientras que la DWT se realiza para la imagen completa de cualquier manera. A diferencia de JPEG2000, CCSDS-IDC no contempla la división de la imagen en subimágenes (teselas) dentro del algoritmo. Dado que esta funcionalidad resulta útil para reducir la memoria requerida de todo el sistema, se agregó modificando los códigos existentes de manera que antes de iniciar la compresión propiamente dicha la imagen se divide en subimágenes de un ancho igual al ancho de la imagen por la cantidad de filas deseadas. Ésto agregó un parámetro de configuración, la cantidad de filas (o alto) de las subimágenes, M_{si} . Todas las subimágenes tendrán el mismo tamaño, excepto la última si la cantidad de filas de la imagen no es múltiplo de M_{si} . Cuando $M_{si} = M$, es decir igual a la cantidad de filas de la imagen, se tiene la operación por área, de otra manera se tiene la operación por franjas con M_{si} filas.

Debido a una limitación de la implementación, no pueden utilizarse imágenes de más de 64×2^{20} píxeles. La mayoría de las imágenes de prueba PAN

exceden este tamaño, por lo tanto, fue necesario restringir el tamaño máximo de las subimágenes. En efecto, se limitó M_{si} a $1/4$ de la cantidad total de filas de la imagen, es decir $M/4$, condición suficiente para las imágenes de prueba utilizadas. En adelante se referirá a esta configuración como $M/4$ y será considerada la operación en modo área.

5.7.2. Casos de prueba para CCSDS-IDC

En la Tabla 5.12 se muestran los casos de prueba para el algoritmo CCSDS-IDC. En primer lugar, los casos de prueba IDC_01 e IDC_02 brindan información para seleccionar entre la DWT flotante y entera. Luego se realizaron simulaciones para determinar el tamaño óptimo de las subimágenes donde se hizo un barrido para su cantidad de filas. Al igual que para JPEG2000, este análisis se llevó a cabo tanto para imágenes de zonas urbanas (IDC_03 e IDC_04) como para imágenes de zonas rurales (IDC_23 e IDC_24), para dos situaciones extremas en cuanto a pérdida de información. Los casos de prueba IDC_05 e IDC_06 muestran el desempeño del compresor en función de CR con el compresor configurado para operar por franjas con tamaño de 32 y 512 filas respectivamente, siendo el primer caso el valor mínimo sugerido por el estándar y el segundo el valor encontrado como óptimo.

Todos los casos de prueba antes mencionados permitieron encontrar la mejor configuración para el compresor. Finalmente los casos de prueba IDC_06, IDC_21, IDC_31 e IDC_41 muestran el desempeño de la configuración adoptada para cada uno de los conjuntos de datos de prueba.

En los casos en que el parámetro de configuración variable es la tasa de compresión se hizo un barrido para los valores $CR = \{0; 2,5; 3; 3,5; 4; 4,5; 5; 6; 8; 10\}$, donde $CR = 0$ representa el caso sin pérdidas. Los barridos para la cantidad de filas de las subimágenes se realizaron para los valores $M_{si} = \{8, 16, 32, 64, 128, 256, 512, 1024, 2048, M/4\}$, donde $M_{si} = 8$ representa la franja más pequeña posible ¹ y $M/4$ representa la operación por área. La cantidad de bloques por segmento, S , se mantuvo constante para todos los casos de simulación y se configuró en modo área (*full frame compression*), es decir, se hizo tal que existe un único segmento por subimagen con una cantidad de bloques $S = \lceil M_{si}/8 \rceil \times \lceil N/8 \rceil$, donde N es el ancho de la subimagen.

¹Aunque no cumple con la recomendación que da el estándar de que $M_{si} \geq 32$, $M_{si} = 8$ es la cantidad mínima de filas que puede tener una imagen que se va a comprimir con CCSDS-IDC si se quiere evitar que se realicen rellenos de filas para alcanzar el valor mínimo necesario para realizar la DWT 2D de 3 niveles.

Tabla 5.12 – *Casos de prueba para CCSDS-IDC.*

Caso de prueba	M_{si}	S	Tipo de DWT	CR	Conjunto de imágenes de prueba
IDC_01	$M/4$ (Área)	Área	Entera	Variable	P_Urb
IDC_02	$M/4$ (Área)	Área	Flotante	Variable	P_Urb
IDC_03	Variable	Área	Entera	0 ^a	P_Urb
IDC_04	Variable	Área	Entera	10	P_Urb
IDC_05	32	Área	Entera	Variable	P_Urb
IDC_06	512	Área	Entera	Variable	P_Urb
IDC_21	32	Área	Entera	Variable	P_Rur
IDC_23	Variable	Área	Entera	0 ^a	P_Rur
IDC_24	Variable	Área	Entera	10	P_Rur
IDC_31	512	Área	Entera	Variable	M_Urb
IDC_41	512	Área	Entera	Variable	M_Rur

^a Implica compresión sin pérdidas, la tasa de compresión dependerá de la imagen.

5.7.3. Resultados de CCSDS-IDC

En esta sección se presenta un resumen de los resultados mientras que en el Anexo J se muestran todos los resultados obtenidos para todos los casos de prueba de CCSDS-IDC.

Comparación entre DWT entera y DWT flotante

En la Fig. 5.17 se comparan los resultados de TPP, SNR_C , MSSIM y $\Delta NIIRS$ para la compresión CCSDS-IDC usando DWT flotante y DWT entera, donde cada punto del gráfico es el valor medio entre las seis imágenes del conjunto P_Urb. Se aprecia que TPP es aproximadamente igual para ambos casos. Aunque los filtros usados en ambas transformaciones son del mismo orden, el hardware necesario para realizar operaciones en punto flotante es más complejo que el necesario para realizar operaciones en punto fijo. La degradación en términos SNR_C y MSSIM no muestra diferencias significativas, con la excepción de que para la DWT flotante no se obtiene compresión sin pérdidas. En la Fig. 5.17(d) se puede ver que en términos de $\Delta NIIRS$ la DWT flotante es levemente superior, pero la diferencia es poco significativa.

Aunque la DWT entera presenta una calidad levemente inferior, debido a que ésta permite compresión sin pérdidas y tiene menor complejidad, fue adoptada en este trabajo. No obstante, si se requiere una implementación para maximizar la calidad de imagen debe seleccionarse su versión flotante.

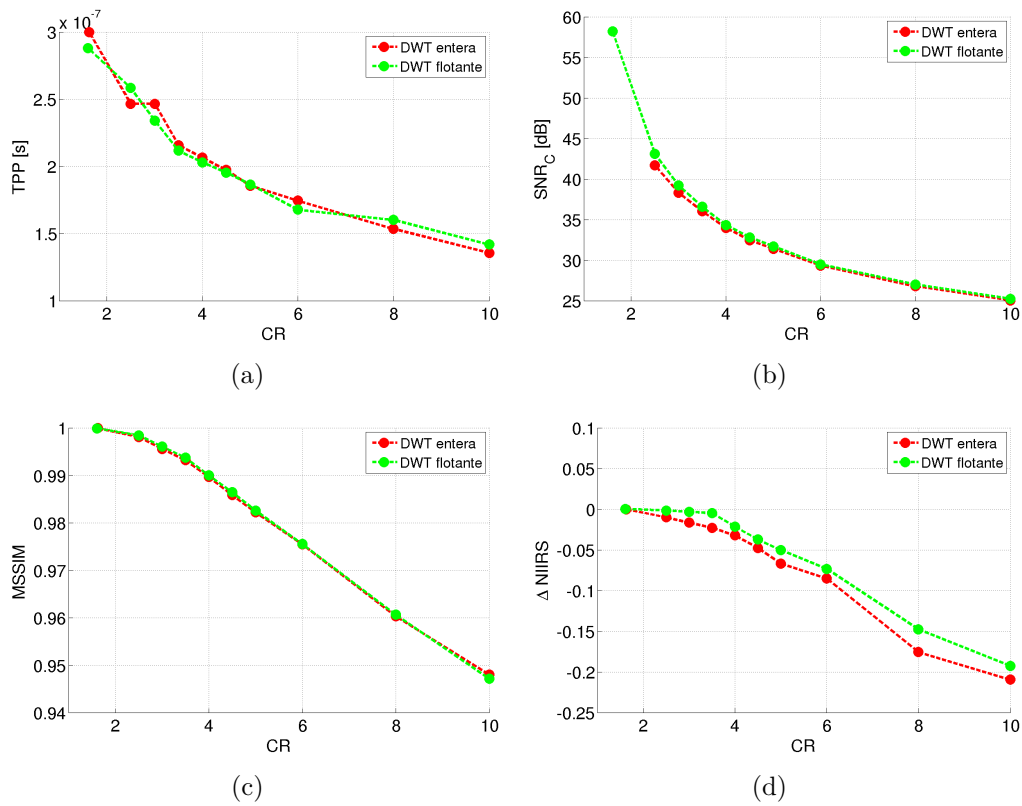


Fig. 5.17. Comparación entre DWT entera y DWT flotante para CCSDS-IDC. Valores promedio de (a) TPP, (b) SNR_C , (c) MSSIM y (d) $\Delta NIIRS$ vs. CR.

Desempeño de acuerdo al tamaño de las subimágenes

En la Fig. 5.18(a) se ve el valor promedio de la tasa de compresión en función de la cantidad de filas de las subimágenes, M_{si} , para el caso sin pérdidas, y en la Fig. 5.18(b) se ve la SNR_C en función de M_{si} para un caso con pérdidas con $CR = 10$. En ambos casos los valores que se muestran en las figuras son el valor promedio de las 6 imágenes del conjunto P_Urb, mientras que los resultados completos puede verse en el Anexo J. Se observa que tanto CR como SNR_C aumentan conforme la tesela es mayor y que para $M_{si} > 512$ la pendiente de crecimiento se vuelve prácticamente constante. Por lo tanto el valor de $M_{si} = 512$ se considera el óptimo.

En el Anexo J, en la Fig. J.19 puede verse que MSSIM sigue la misma ley de variación y en la Fig. J.20 que el $\Delta NIIRS$ no cambia de manera significativa ni presenta una ley de variación concluyente. También puede verse que con imágenes rurales, casos IDC_23 e IDC_24, se llega a las mismas conclusiones.

En la Fig. 5.19 se ilustra el desempeño del compresor en función de CR para tres tamaños de tesela distintos, $M_{si} = M/4$ (área), $M_{si} = 512$ filas (valor

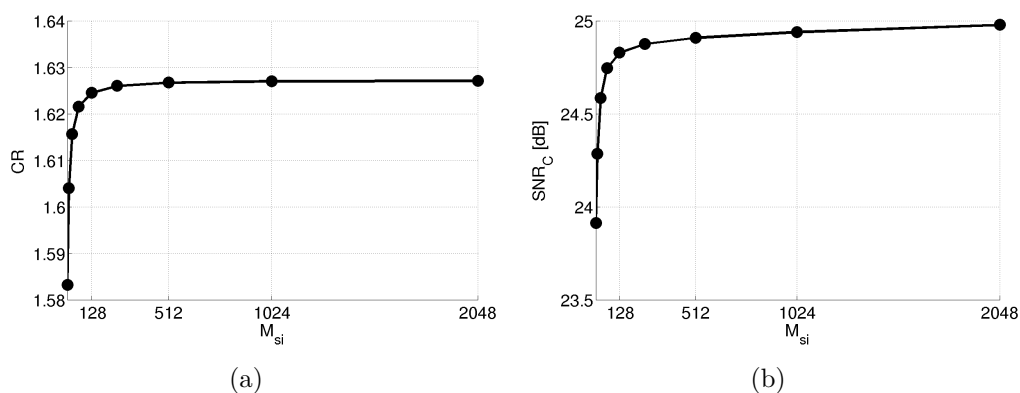


Fig. 5.18. Desempeño CCSDS-IDC en función de del tamaño de subimagen. Valores promedio de (a) CR vs. M_{si} para JP2_03 y (b) SNR_C vs. M_{si} para JP2_04.

óptimo) y $M_{si} = 32$ filas (valor mínimo sugerido por el estándar). Como puede verse $M_{si} = 512$ representa una situación intermedia en términos de SNR_C . Por otro lado, observando el $\Delta NIIRS$, aunque se ve una leve superioridad por parte de la operación por área, la degradación es prácticamente idéntica para los tres casos.

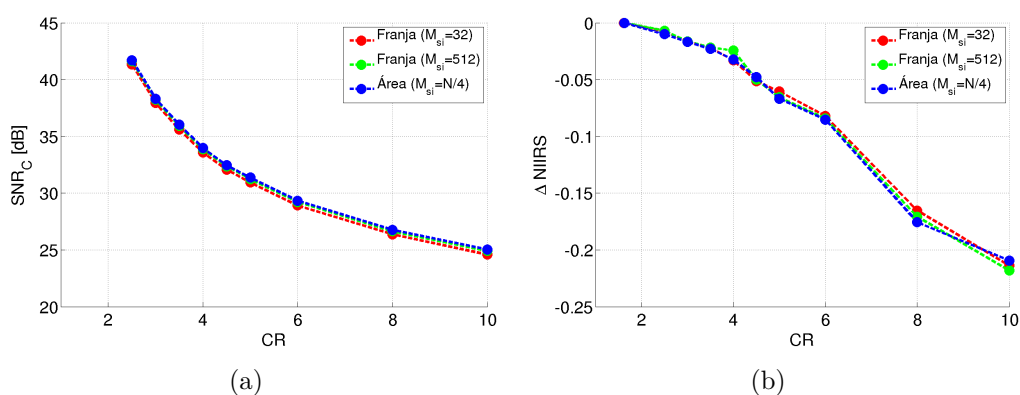


Fig. 5.19. Comparación del desempeño de CCSDS-IDC según tamaño de tesela en función de CR para P_Urb. Valores promedio de (a) SNR_C y (b) $\Delta NIIRS$ vs. CR.

Configuración adoptada

En virtud de lo antes mencionado se adopta la configuración que corresponde a la DWT entera con $M_{si} = 512$ y el resto de los parámetros en su valor por defecto. Los casos IDC_06, IDC_21, IDC_31 e IDC_41 muestran el desempeño con esta configuración en función de CR.

Al igual que para JPEG2000, en CCSDS-IDC, la tasa de compresión es constante e igual al valor especificado por lo tanto se goza de las mismas ventajas.

Respecto a la complejidad, con la configuración adoptada se necesitan almacenar 512 filas de la imagen, no obstante, se ha mostrado que es posible obtener implementaciones que requieren menor cantidad de memoria con comportamientos similares. Por otro lado en la Fig. 5.20(a) puede verse que el TPP está en el orden de los $200ns$ para el conjunto P_Urb.

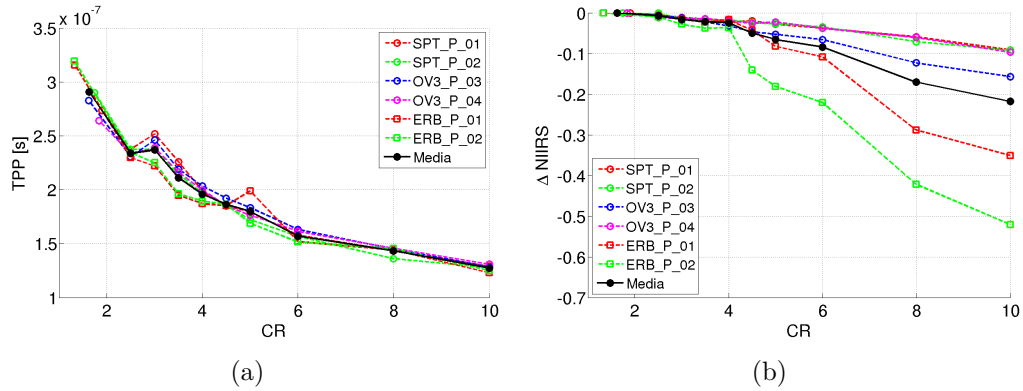


Fig. 5.20. Resultados de CCSDS-IDC con P_Urb. (a) TPP y (b) $\Delta NIIRS$ vs. CR.

En lo que hace a la calidad de imagen, para $CR \leq 4$, el NIIRS no se degrada más del valor mínimo visualmente perceptible (0,1) para ninguna de las imágenes del conjunto P_Urb, ver Fig. 5.20(b). Al igual que para JPEG-Baselines y JPEG2000, la degradación del NIIRS se debe principalmente a la disminución del RER, como puede verse comparando las Figs. J.31, J.32 y J.33.

En la Fig. 5.21(a) se puede ver SNR_C y en la Fig. 5.21(b) MSSIM en función de CR, donde se aprecia que la degradación de la calidad depende de la imagen comprimida.

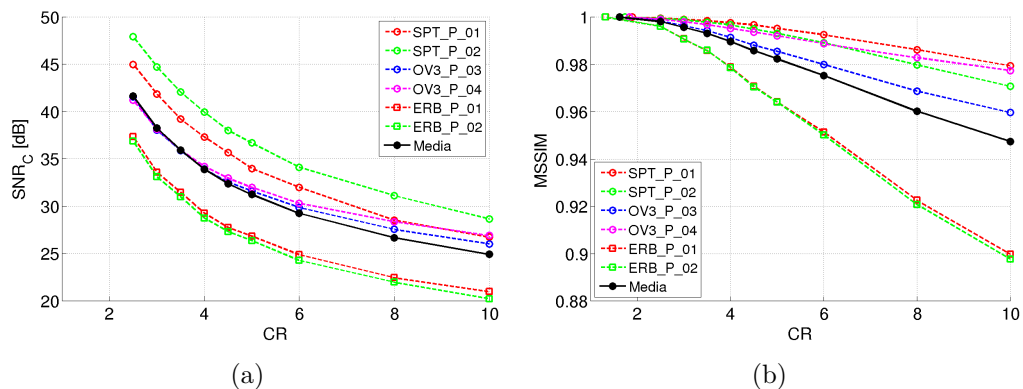


Fig. 5.21. Resultados de CCSDS-IDC con P_Urb. (a) SNR_C y (b) MSSIM vs. CR.

5.8. Análisis visual

Se realizó un análisis visual reducido para obtener una apreciación general del estado de las imágenes comprimidas y verificar la existencia de distorsiones notables, comúnmente llamadas artefactos de compresión. Dicho análisis se llevó a cabo mirando imágenes PAN comprimidas con $CR = 3,5$ y $CR = 8$. La calidad de imagen fue evaluada acorde a los grados indicados en la Tabla 5.13 que se utilizó en análisis similares [96]. Dicha calificación se basa en una apreciación global de la imagen.

Tabla 5.13 – Grados de calidad subjetiva para el análisis visual de imágenes comprimidas.

Calidad Subjetiva	Descripción
Excelente	Pequeñas diferencias comparando con la imagen original, la distorsión es pequeña y casi puede ser ignorada.
Buena	Pequeña degradación comparando con la imagen original, la distorsión es evidente pero no tiene efecto en la interpretación de la imagen y casi puede ser ignorada.
Aceptable	La distorsión es evidente comparando con la imagen original y la interpretación de la imagen se ve poco afectada.
Pobre	La distorsión es grave comparando con la imagen original y se ve afectada la interpretación de la imagen.

Para JPEG-LS se observaron las imágenes PAN urbanas comprimidas con $NEAR = 5$ y $NEAR = 24$ ya que para estos valores se obtienen las tasas de compresión promedio, \overline{CR} , cercanas a 3,5 y 8 respectivamente. Por los mismos motivos, las imágenes PAN rurales se observaron comprimidas con $NEAR = 2$ y $NEAR = 8$. Con $\overline{CR} \cong 3,5$, para las imágenes urbanas y rurales, se observó una calidad excelente, incluso para algunas no se vieron diferencias. En las Figs. 5.22(b) y 5.23(b) se ilustran ejemplos donde no se observaron diferencias comparando con las imágenes originales de las Figs. 5.22(a) y 5.23(a) respectivamente. Con $\overline{CR} \cong 8$, se observó que la calidad es pobre y en general la imagen está completamente distorsionada comparando con la imagen original. En toda la imagen pueden observarse muchos píxeles que fueron comprimidos en el modo RLE y por lo tanto son descomprimidos como píxeles de igual intensidad, como se muestra en las Figs. 5.22(c) y 5.23(c).

Para JPEG-Baselines se observaron las imágenes del conjunto P_Urb comprimidas con $Quality = 100$ y $Quality = 75$ ya que para estos valores se obtienen $\overline{CR} \cong 3,5$, y $\overline{CR} \cong 8$, respectivamente. Por los mismos motivos, las imágenes del conjunto P_Rur se observaron comprimidas con $Quality = 100$ y $Quality = 92$. Para $\overline{CR} \cong 3,5$, en las imágenes urbanas en general se observó

una calidad buena, que se aprecia inferior a lo observado para JPEG-LS y presenta artefactos de bloque que pueden verse en la Fig. 5.22(e). En las imágenes rurales, para $\overline{CR} \cong 3,5$, se apreció una calidad pobre con distorsiones que ya no permiten identificar la textura del terreno, como se ilustra en la Fig. 5.22(e). Con $\overline{CR} \cong 8$, para las imágenes urbanas se observaron calidades buenas y aceptables con muchos artefactos de bloque mientras que para las imágenes rurales prácticamente toda la imagen está distorsionada comparando con la imagen original. En las Figs. 5.22(f) y 5.23(f) se ven los casos con $\overline{CR} \cong 8$ donde hay gran cantidad de artefactos de bloque.

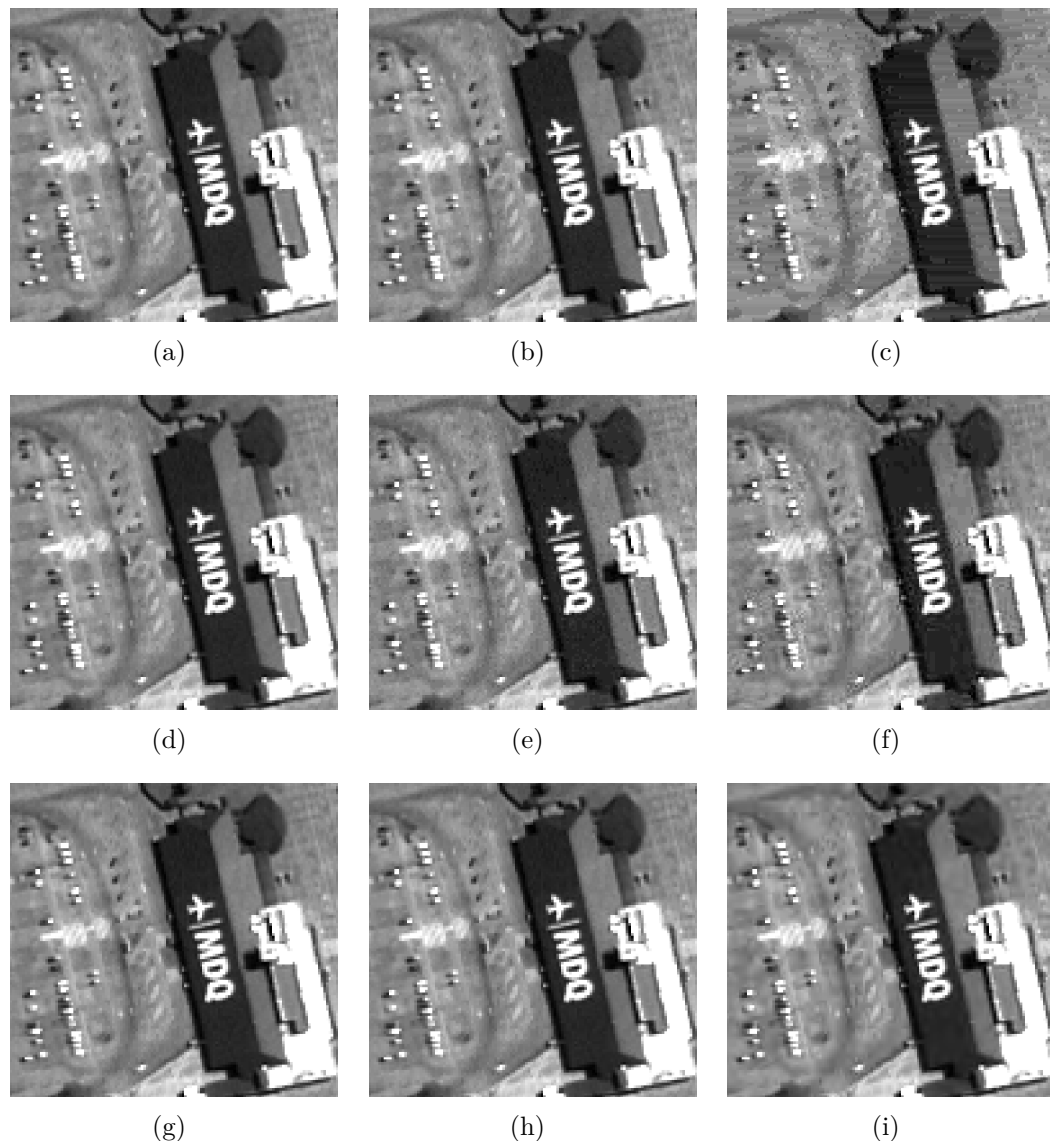


Fig. 5.22. Análisis visual de imágenes P -Urb comprimidas, recortes de 128×128 píxeles de OV3_P-03. (a), (d) y (g) Original. Compresión con JPEG-LS con (b) $NEAR = 5$ y (c) $NEAR = 24$. Compresión con JPEG-Baselines con (e) $Quality = 100$ y (f) $Quality = 75$. Compresión con CCSDS-IDC con (h) $CR = 3,5$ e (i) $CR = 8$.

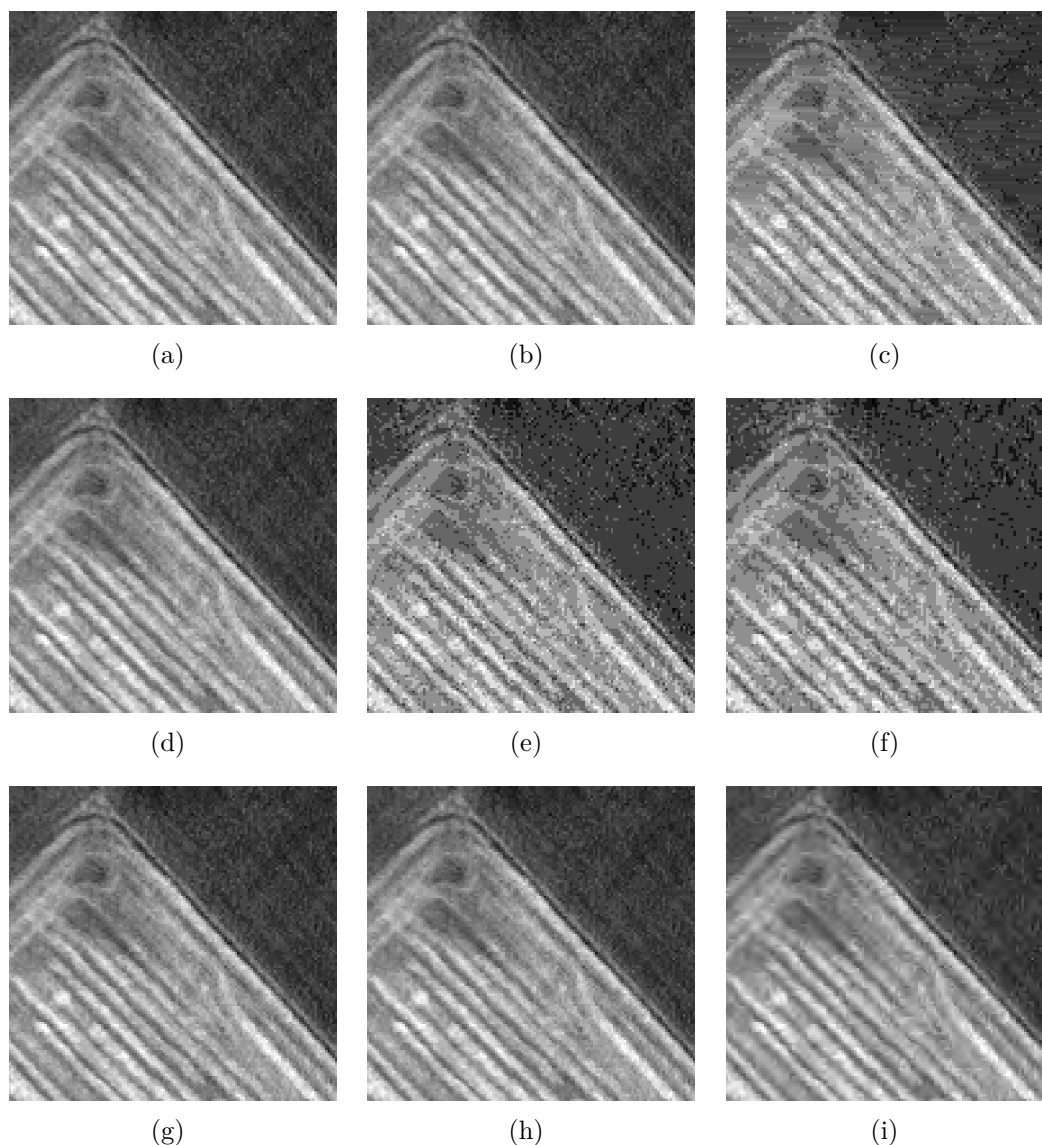


Fig. 5.23. *Análisis visual de imágenes P_Rur comprimidas, recortes de 128×128 píxeles de OV3_P_06. (a), (d) y (g) Original. Compresión con JPEG-LS con (b) NEAR = 2 y (c) NEAR = 8. Compresión con JPEG-Baselines con (e) Quality = 100 y (f) Quality = 92. Compresión con CCSDS-IDC con (h) CR = 3,5 e (i) CR = 8.*

Para ambos sistemas basados en DWT las calidades observadas fueron iguales, por lo tanto en las Figs. 5.22 y 5.23 sólo se muestran ejemplos para CCSDS-IDC. Con $CR = 3,5$ la calidad observada tanto en imágenes urbanas como rurales fue excelente, incluso para algunas no se observaron diferencias comparando con la imagen original, como se ejemplifica en las Figs. 5.22(h) y 5.23(h). Con $CR = 8$ para las diferentes imágenes observadas algunas se vieron más degradadas que otras y las calidades pueden clasificarse en buena y aceptable. Se apreciaron evidentes artefactos y borrono en algunas zonas de la imagen como

se puede ver en las Figs. 5.22(i) y 5.23(i). Además se identificaron fácilmente discontinuidades entre subimágenes (o teselas) adyacentes.

En conclusión, para JPEG-Baselines se percibió la peor calidad en la mayoría de los casos, mientras que JPEG-LS, JPEG2000 y CCSDS-IDC presentaron calidades excelentes para $CR = 3,5$ y para tasas de 8 los algoritmos basados en DWT presentaron los mejores resultados.

5.9. Comparación entre sistemas y selección

Tasa de compresión

Respecto a la tasa de compresión, en los sistemas JPEG-LS y JPEG-Baselines, ésta no se puede especificar de manera directa sino que es consecuencia del valor que tome el parámetro de control de calidad (*NEAR* o *Quality* según corresponda) y de las características de la imagen comprimida, como se ha mostrado. Ésto plantea inconvenientes desde el punto de vista de la operación del instrumento ya que no se conoce con anticipación cuanta memoria ocupará cada imagen. No obstante, asociado a cada valor del parámetro de control de calidad se tiene una tasa de compresión promedio que puede servir como un indicador de la CR alcanzada para cada valor de *NEAR* o *Quality*. Por lo contrario, los sistemas de compresión basados en DWT son de CR fija (en los casos con pérdidas), no se tienen los inconvenientes antes mencionados y presentan ventajas desde el punto de vista de la operación del instrumento ya que se conoce cuanto espacio de memoria ocupará cada imagen.

De aquí en adelante el desempeño de los sistemas JPEG-LS y JPEG-Baselines se analiza en función de la tasa de compresión promedio para poder hacer comparaciones con los restantes.

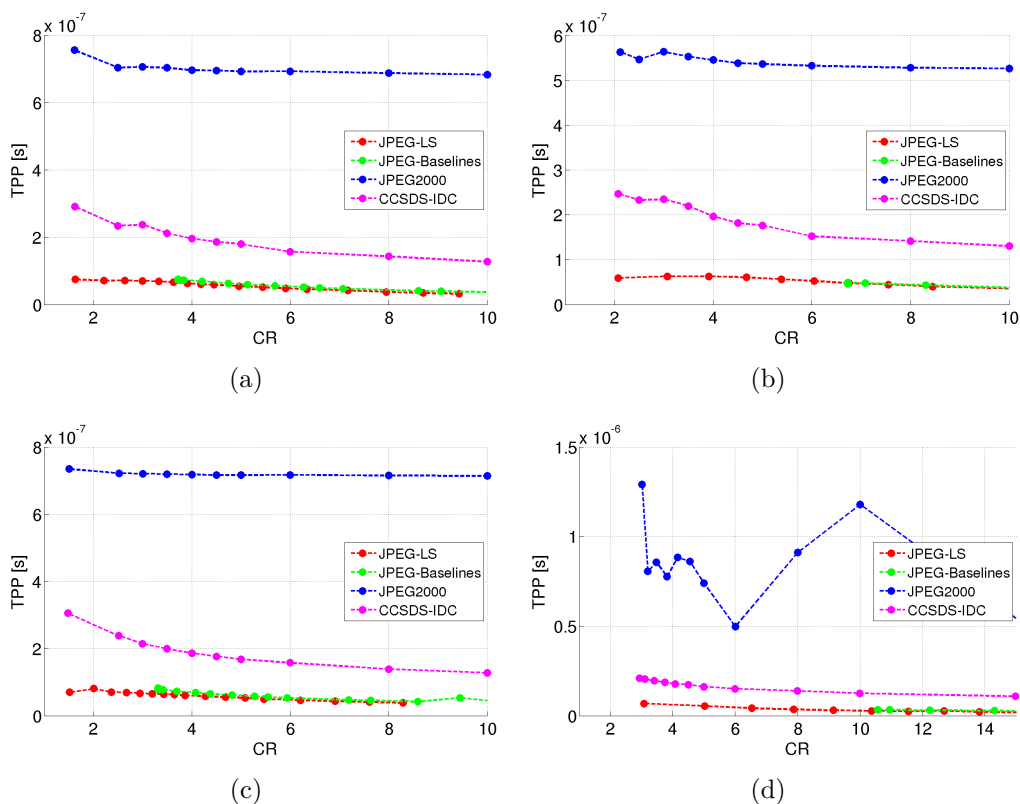
Complejidad

Respecto a la complejidad, en la Tabla 5.14 se compara la cantidad de filas que deben almacenarse en la memoria de entrada del sistema de compresión, M_{in} , para cada uno de los sistemas. Cabe destacar que para los sistemas de compresión basados en DWT se muestra el valor que fue encontrado como óptimo, no obstante se mostró que es posible lograr sistemas donde M_{in} se minimiza a 32 filas, pero debería evaluarse la posibilidad de que franjas adyacentes presenten diferentes calidades de imagen y las discontinuidades sean visibles.

Tabla 5.14 – Cantidad de filas de la imagen que deben almacenarse en la memoria de entrada del compresor para cada algoritmo evaluado.

Algoritmo	M_{in}
JPEG-LS	1
JPEG-Baselines	8
JPEG2000	512
CCSDS-IDC	512

Como segundo indicador de complejidad, en la Fig. 5.24 se muestra el TPP de cada algoritmo de compresión para los cuatro conjuntos de datos de prueba. De manera general se aprecia que JPEG-Baselines y JPEG-LS son los que requieren menor tiempo, mientras que CCSDS-IDC necesita un tiempo aproximadamente 4 veces mayor y JPEG2000 un tiempo un orden de magnitud mayor a los primeros.


Fig. 5.24. Comparación de TPP vs. CR, entre los cuatro sistemas de compresión, para el conjunto de datos de prueba (a) *P_Urb*, (b) *P_Rur*, (c) *M_Urb* y (d) *M_Rur*.

Degradación de la calidad de imagen

Respecto a la calidad de imagen, en términos de la degradación del NIIRS, se ha mostrado que éste se degrada principalmente debido al borroneo en la imagen, es decir la disminución de RER. En la Fig. 5.25 se muestra el

$\Delta NIIRS$ de los cuatro sistemas de compresión para cada uno de los conjuntos de datos de prueba. Observando el $\Delta NIIRS$ para el conjunto de datos P_Urb, ilustrado en la Fig. 5.25(a), puede verse que JPEG-LS presenta el mejor comportamiento ya que prácticamente no degrada el NIIRS, en segundo lugar se tiene JPEG-Baselines, luego JPEG2000 y por último CCSDS-IDC que difiere poco del antes mencionado.

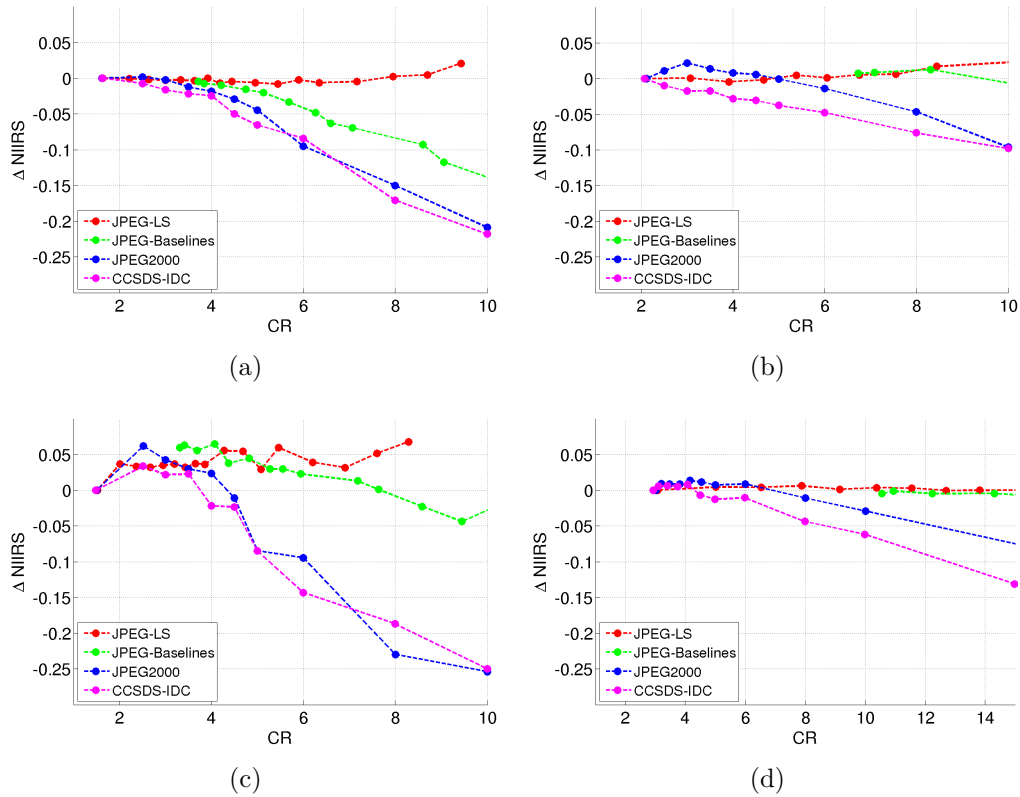


Fig. 5.25. Comparación de $\Delta NIIRS$ vs. CR, entre los cuatro sistemas de compresión, para el conjunto de datos de prueba (a) P-Urb, (b) P-Rur, (c) M-Urb y (d) M-Rur.

Para el resto de los conjuntos de datos pueden observarse comportamientos que conducen a las mismas conclusiones, aunque en algunos casos se observan aumentos en el NIIRS a pesar de la compresión. Si bien estudios adicionales deberían realizarse para determinar la causa, estos aumentos podrían deberse a dos motivos: 1) artefactos de compresión o 2) defectos en el algoritmo de estimación.

Como segundo indicador de calidad de imagen, en la Fig. 5.26 se muestra la SNR_C de acuerdo a la tasa de compresión. Para todos los conjuntos de datos de prueba, puede observarse que a bajas tasas de compresión JPEG-LS presenta el mejor comportamiento; mientras que para $CR \geq 6$, para las imágenes PAN, los sistemas basados en DWT pasan a tener un mejor desempeño. En general se ve

que JPEG-Baselines da como resultado las imágenes comprimidas que presentan el error más grande con la imagen original.

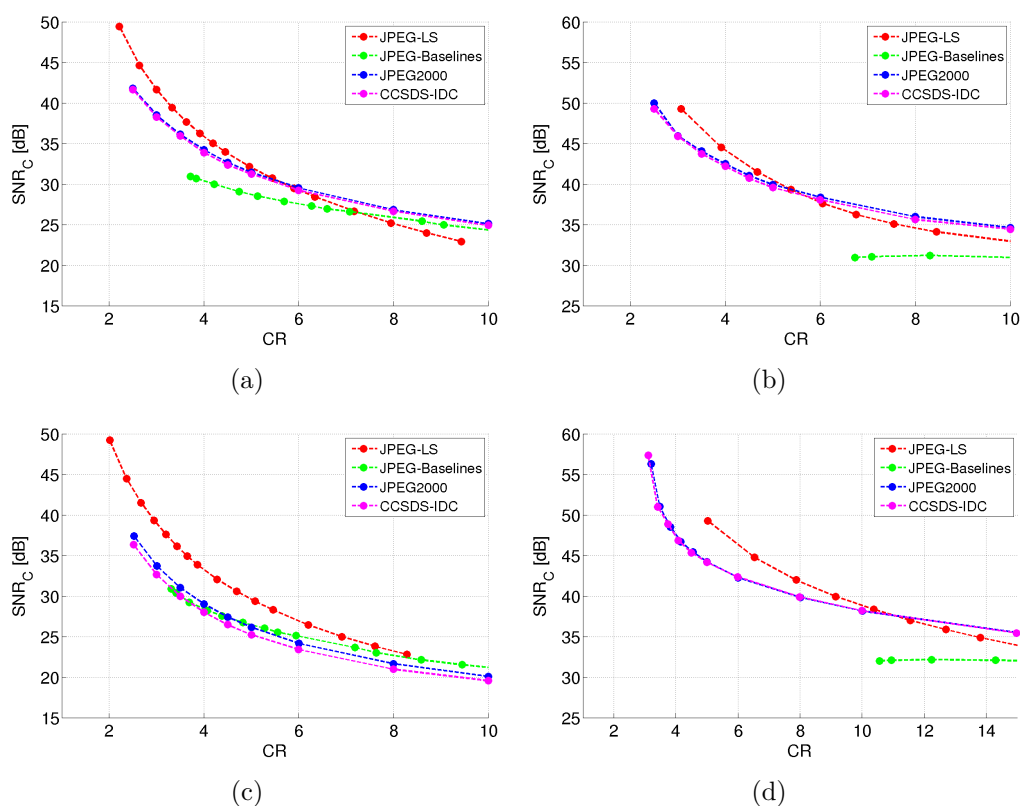


Fig. 5.26. Comparación de SNR_C vs. CR , entre los cuatro sistemas de compresión, para el conjunto de datos de prueba (a) $P-Urb$, (b) $P-Rur$, (c) $M-Urb$ y (d) $M-Rur$.

Como último indicador cualitativo de la degradación de la calidad de imagen se tiene el MSSIM que se ilustra en la Fig. 5.27. Nuevamente se aprecia que JPEG-LS presenta los mejores resultados a tasas de compresión bajas, mientras que en las imágenes PAN los sistemas basados en DWT son los que menos degradan la estructura para $CR \geq 6$. Por otro lado, JPEG-Baselines es en general el sistema que más degrada la estructura de la imagen.

Selección

El sistema de compresión JPEG-LS es el de menor complejidad y presenta el mejor desempeño de calidad de imagen en base a todas las métricas cuantitativas empleadas. Por lo tanto, se selecciona como primera opción.

El algoritmo JPEG-Baselines, si bien presenta buenos resultados en términos del NIIRS, es el que peor se comporta observando las métricas clásicas de calidad de imagen y el análisis visual mostró gran cantidad de artefactos visibles. Por lo tanto si la selección de un sistema de compresión se basa sólo

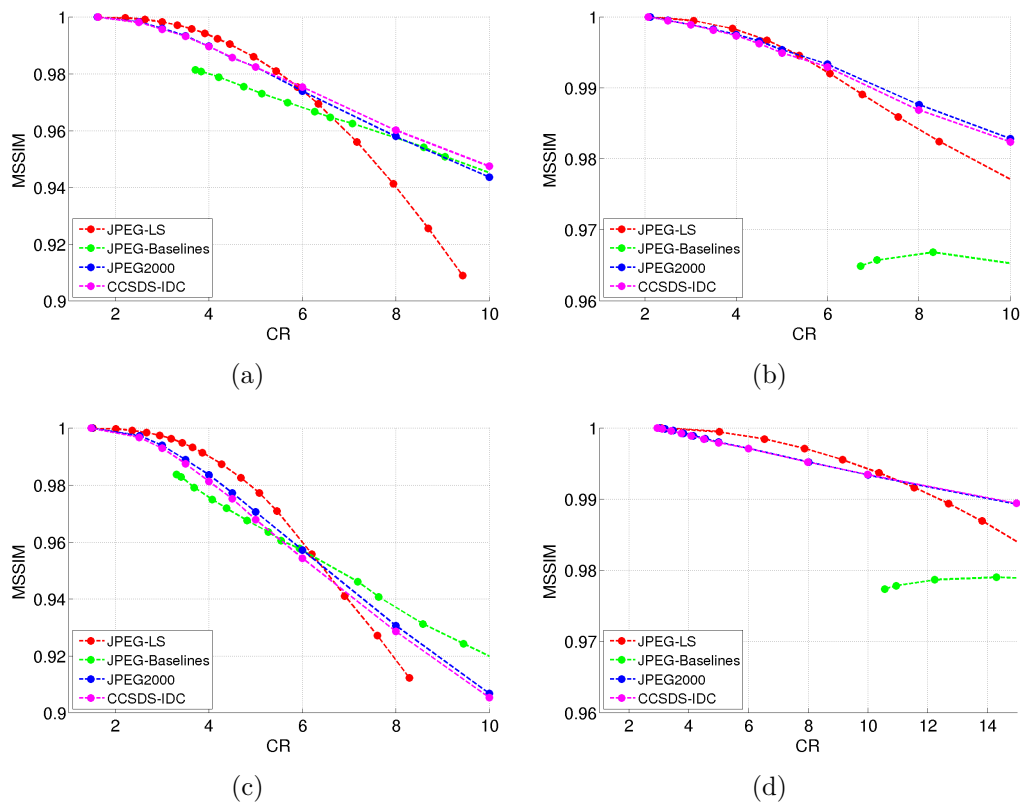


Fig. 5.27. Comparación de MSSIM vs. CR, entre los cuatro sistemas de compresión, para el conjunto de datos de prueba (a) *P_Urb*, (b) *P_Rur*, (c) *M_Urb* y (d) *M_Rur*.

en el NIIRS, JPEG-Baselines debe ser considerado dada su baja complejidad. No obstante como en esta tesis se aplica un criterio general, evaluado todas las métricas, este sistema se descarta.

Respecto a los sistemas basados en DWT, se ha observado que JPEG2000 presenta una calidad levemente mejor que CCSDS-IDC a costa de al menos duplicar la complejidad computacional. En consecuencia como segunda alternativa se propone CCSDS-IDC con la configuración adoptada en este trabajo, es decir, DWT entera, subimágenes de 512 filas y un único segmento por subimagen. Incluso si se desea mejorar la calidad de imagen, a costa de un leve aumento en complejidad que practicante no altera el TPP, puede utilizarse la DWT flotante.

Tanto para el sistema de compresión JPEG-LS, como para el CCSDS-IDC, con la configuración adoptada y con una tasa de compresión promedio de 3,5, que permitiría transmisión en tiempo real para la CAR montada en un satélite con las características del SARE 2A, el NIIRS no se degrada una cantidad mayor al valor mínimo visualmente perceptible (0,1) para ninguna de las imágenes con las que se realizaron ensayos.

Conclusiones y trabajos a futuro

6.1. Conclusiones

En esta tesis se presentaron todas las actividades que condujeron a la selección de un sistema de compresión óptimo para una cámara satelital de alta resolución contemplando el desempeño del algoritmo en términos de su tasa de compresión, complejidad y degradación de la calidad de imagen, donde esta última se evaluó con métricas clásicas y en términos de la degradación del NIIRS.

La revisión del estado del arte de la compresión de imágenes a bordo de misiones satelitales, contemplando sistemas basados en predicción, DCT y DWT, permitió realizar una preselección de cuatro sistemas de compresión compatibles con la CAR. Los algoritmos JPEG-LS, JPEG-Baselines, JPEG2000 y CCSDS-IDC fueron elegidos ya que se utilizaron en misiones espaciales, son capaces de lograr la tasa de compresión deseada para la CAR y dado que están estandarizados se dispone de la información completa del algoritmo. En otro orden, la revisión mostró que, si se clasifican los sistemas de compresión respecto a su base teórica, los más utilizados fueron los basados en predicción; si se clasifican en base a la pérdida de información, los más utilizados fueron los con pérdidas; y si se clasifican de acuerdo a si son algoritmos estandarizados o no, los pertenecientes al primer grupo fueron los más utilizados. También se observó que en misiones espaciales se realizó compresión con tasas que van desde la compresión sin pérdidas hasta tasas que logran una reducción del tamaño de la imagen en un factor de 10.

El estudio de las métricas de calidad de imagen indicó que la mejor alternativa para la estimación del NIIRS en imágenes satelitales es hacerlo basándose en la información contenida en bordes objetivos naturales presentes en imágenes satelitales.

El desarrollo del algoritmo de caracterización de la respuesta al borde de un sistema de imagen, que contempló analizar el desempeño de métodos basados en detectores de bordes y métodos basados en el ajuste funciones cúbicas, spline y sigmoide; condujo a seleccionar el método basado en el ajuste de funciones spline ya que fue el más rápido y exacto. Finalmente, estos algoritmos se aplicaron para la estimación del NIIRS de imágenes satelitales y su validación con imágenes de NIIRS conocido mostró resultados satisfactorios.

Para realizar las pruebas del desempeño de los sistemas de compresión se deseaba utilizar imágenes de zonas rurales y zonas urbanas que tuvieran las mismas características de los datos crudos que llegarían al compresor de la CAR. Debido a la inexistencia de instrumentos locales similares previos a la CAR, la diferencia de características entre los instrumentos de distintas misiones espaciales extranjeras, y el difícil o acotado acceso a imágenes crudas, no fue posible satisfacer totalmente lo deseado. Sin embargo, con las imágenes disponibles, se logró seleccionar un conjunto de ellas con la mayor cantidad de aspectos similares a la CAR.

Mediante la implementación de los sistemas de compresión preseleccionados en MATLAB, utilizando librerías existentes en C/C++, se realizó la evaluación del desempeño de los mismos. En primer lugar, la aplicación de la relación de compromiso perseguida en este trabajo a los sistemas JPEG-Baselines, JPEG2000 y CCSDS-IDC; permitió ajustar sus sus parámetros de control para obtener un desempeño óptimo y si bien se ha adoptado una determinada configuración, los datos obtenidos permiten aplicar otros criterios que conducirían a configuraciones diferentes. En segundo lugar, la comparación entre los cuatro sistemas preseleccionados, condujo a la conclusión de que los algoritmos JPEG-LS y CCSDS-IDC presentan las mejores opciones para la CAR. El primero, requiere una menor complejidad computacional y menor memoria, proporcionando una tasa de compresión variable de acuerdo a la imagen; y el segundo, por el contrario, requiere una mayor complejidad de implementación y mayor memoria pero su tasa de compresión es constante. Respecto a la degradación de la calidad de imagen, en general se observó que JPEG-LS presenta un mejor desempeño que CCSDS-IDC para tasas de compresión bajas mientras que los resultados se invierten para tasas de compresión cercanas a 10. Además, con ambos sistemas de compresión es posible comprimir las imágenes generadas por la CAR a una tasa de 3,5, que en una plataforma con las características del SARE 2A permitiría la transmisión en tiempo real, sin degradar el NIIRS en una cantidad visualmente perceptible.

6.2. Trabajos a futuro

Si bien se considera que en la presente tesis se han cumplido las expectativas en cuanto a los alcances de la misma, quedan aspectos pendientes a ser profundizados en el algoritmo de estimación del NIIRS, los datos de prueba utilizados para ensayar los sistemas de compresión y el diseño del sistema de compresión de la CAR.

En lo que concierne a lo primero, con la finalidad de eliminar la asistencia de una persona para seleccionar los bordes objetivos naturales para la estimación del NIIRS y reducir el tiempo que toma dicha tarea, es necesario desarrollar técnicas de detección automática de bordes que consideren las características que éstos deben tener para ser aptos para la estimación. Otro aspecto a considerar es la validación del algoritmo de estimación del NIIRS con mayor cantidad de imágenes y contra valores de NIIRS estimados subjetivamente por analistas de imagen. Ésto no sólo daría una validación con datos más confiables sino que además permitiría introducir correcciones en el algoritmo para optimizar la estimación.

Respecto al conjunto de datos de prueba, con la finalidad de poder ajustar los parámetros de configuración de los sistemas de compresión de manera más exacta y predecir con mayor certeza la tasa de compresión de los sistemas de tasa de compresión variable, como el JPEG-LS y JPEG-Baselines, es menester formar nuevos conjuntos de datos con imágenes de características idénticas o casi idénticas a la CAR. Dicho objetivo puede ser logrado utilizando imágenes de la misma CAR para diseñar el sistema de compresión para futuras misiones que lleven la misma, realizando adquisiciones con la CAR montada en un avión, o simulando imágenes con las características de la CAR a partir de imágenes de un GSD menor. En la Sección 5.1 se proponen algunos lineamientos para la última opción. También, es necesario considerar la posibilidad de contar con imágenes crudas para evaluar la diferencia entre, corregir las imágenes por no uniformidad de los píxeles y viñeteo y luego comprimir a bordo, o comprimir a bordo y luego corregir en tierra.

En tercer lugar, aunque se evaluaron los sistemas de compresión para diversas configuraciones, quedó un grupo menor de parámetros de configuración que no han sido modificados, tales como:

- en JPEG-LS: los niveles de umbral de la cuantificación de gradientes, $[T_1, T_2, T_3]$ y la cantidad de píxeles cada la cual se dividen a la mitad los acumuladores para el cálculo de la predicción adaptativa, *RESET*;
- en JPEG2000: el tamaño de los bloques de código y recintos;
- en CCSDS-IDC: la cantidad de bloques por segmento y factores de peso de cada una de las sub-bandas cuando se aplica DWT entera.

Dichos parámetros, pueden ser ajustados durante el diseño detallado del sistema de compresión del instrumento con el objetivo de lograr un desempeño óptimo para la aplicación o adecuarlos al hardware disponible. Otro aspecto pendiente, que involucra a los sistemas de compresión, es realizar evaluaciones de la complejidad del algoritmo de acuerdo los recursos de hardware concretos que éste necesite.

Anexos

Cálculo del tiempo de línea

En la Fig. A.1 se ilustra un satélite que órbita a la tierra a una altura H , para calcular el tiempo de línea es necesario conocer la velocidad del satélite proyectada sobre la superficie terrestre, v_{st} .

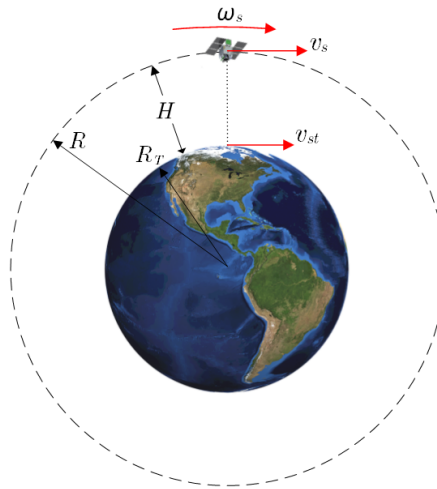


Fig. A.1. Órbita y velocidades (la imagen no está en escala).

Para la CAR, donde se tiene una órbita heliosincrónica de altura $H = 550 \text{ Km}$, que puede suponerse circular sin cometer grandes errores, la tercera ley de Kepler permite calcular el período de una órbita como

$$T = \sqrt{\frac{4 \cdot \pi^2 \cdot (R_T + H)^3}{\mu}}, \quad (\text{A.1})$$

donde $R_T = 6371 \text{ Km}$ es el radio medio de la tierra y $\mu = 3,986 \times 10^8 \text{ m}^3/\text{s}^2$ es el parámetro de gravitación universal estándar para la tierra. La velocidad angular

del satélite en dicha órbita es

$$\omega = \frac{2.\pi}{T} = \frac{2.\pi}{\sqrt{\frac{4.\pi^2.(R_T+H)}{\mu}}}. \quad (\text{A.2})$$

Lo que finalmente permite calcular la velocidad del satélite proyectada sobre la tierra, es decir, la velocidad con la que la proyección de los píxeles en la tierra recorren el terreno

$$v_{st} = \omega.R_T = \frac{2.\pi}{\sqrt{\frac{4.\pi^2.(R_T+H)}{\mu}}}.R_T = 6986 \text{ m/s}. \quad (\text{A.3})$$

El tiempo de línea, es el tiempo que le toma a la línea de píxeles proyectada en la tierra recorrer una distancia igual al GSD. Recordando que $GSD_{PAN} = 1 \text{ m}$ y $GSD_{MS} = GSD_B = GSD_G = GSD_R = GSD_{NIR} = 4 \text{ m}$, se obtienen los tiempos de línea para el canal PAN y el MS respectivamente,

$$T_{l_{PAN}} = \frac{GSD_{PAN}}{v_{st}} = 143,16 \mu s, \quad (\text{A.4})$$

$$T_{l_{MS}} = \frac{GSD_{MS}}{v_{st}} = 572,58 \mu s. \quad (\text{A.5})$$

Relevamiento de compresión de imágenes a bordo

En las Tablas B.1 a B.5 se resume la información relevada para cada uno de los casos en los que se aplicó compresión de imágenes a bordo de misiones espaciales. La información se encuentra ordenada alfabéticamente, por el nombre del satélite o misión, que se encuentra en la primera columna. En algunos casos, junto al nombre de satélite o misión, se indica entre paréntesis el nombre del instrumento. En la segunda y tercera columna se indica la empresa, organización o país propietario del mismo y su año de lanzamiento respectivamente. En la columna “Uso” se indica si es un satélite de observación terrestre de alta resolución (EO-HR), de media resolución (EO-MR) o de otros usos (que incluye misiones de observación de otros cuerpos celeste y misiones de demostración tecnológica). Los casos donde el GSD de la banda PAN es menor o igual 10 metros se consideraron como EO-HR de lo contrario se consideraron como EO-MR. En las columnas 5 a 9 se muestran las características del sistema de compresión y en la última columna las fuentes de las que se extrajo la información.

Tabla B.1 – Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (1/5).

Nombre	Propietario	Año	Uso	Algoritmo de compresión	Estándar	Base	Con / Sin pérdida	CR	Referencias
ALOS (AVNIR-2)	JAXA	2006	EO-HR	Lossless JPEG	Si	Predicción	Sin Pérd.	1,33	[97]
ALOS (PRISM)	JAXA	2006	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	4,5 a 9	[97]
Beijing-1 (CMT)	MoST / China	2005	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.		[3]
BilSAT-1	TUBITAK / Turquía	2003	EO-MR	JPEG2000	Si	DWT	Opcional		[98–100]
Cartosat-1	ISRO	2005	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	hasta 3,2	[3]
Cartosat-2	ISRO	2007	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	hasta 3,3	[3]
Cassini Probe (CDA)	NASA-JPL	1997	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[101]
Cassini Probe	NASA-JPL	1997	Otro	ICM	No	DCT	Con Pérd.		[1, 2]
Clementine	NASA	1994	Otro	ICM	No	DCT	Con Pérd.		[1, 2, 102]
DMC-3 a	DMCii	2015	EO-HR	JPEG-LS	Si	Predicción	Opcional	2	[30]
DMC-3 b	DMCii	2015	EO-HR	JPEG-LS	Si	Predicción	Opcional	2	[30]
DMC-3 c	DMCii	2015	EO-HR	JPEG-LS	Si	Predicción	Opcional	2	[30]
EnMAP	DLR	2017+	EO-MR	CCSDS-IDC	Si	DWT	Opcional	1,6	[35, 38]
EO-1	NASA	2000	EO-MR	CCSDS-LDC	Si	Predicción	Sin Pérd.		[23, 103]
ETS 7	JAXA	1997	Otro	ICM	No	DCT	Con Pérd.		[1]
FedSAT	Australia	2002	Otro	SAICS	No	Predicción	Con Pérd.	1,6 a 3,4	[31]
FormoSat-5	NSPO	2016	EO-HR	CCSDS-IDC	Si	DWT	Opcional	1,5 ^a a 7,5 ^b	[37, 104, 105]

^a Para compresión sin pérdidas.^b Para compresión con pérdidas.

Tabla B.2 – Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (2/5).

Nombre	Propietario	Año	Uso	Algoritmo de compresión	Estándar	Base	Con / Sin pérdida	CR	Referencias
Ikonos-2	Digital Globe	1999	EO-HR	ADPCM (Kodak)	No	Predicción	Con Pérd.	4,23	[11, 72, 106]
IMAGE / MIDEX-1	NASA-JPL	2000	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[101]
IMS-1 / TWSat	ISRO	2008	EO-MR	JPEG2000	Si	DWT	Opcional	3,4	[3, 107]
KazEOSat-1	Airbus Group Astrium / Kazakhstan	2014	EO-HR	MRCPB-CORECI	No	DWT	Opcional		[108]
Kompsat-1	KARI	1999	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[101]
Landsat 8	USGS	2013	EO-MR	CCSDS-LDC	Si	Predicción	Sin Pérd.	1,55	[22, 109]
LEWIS	NASA	1997	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[101]
MAP / MIDEX-2	NASA-JPL	2001	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[101]
Mars Odyssey	NASA	2001	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[3]
Mars Probe	CNES	1996	Otro	ICM	No	DCT	Con Pérd.		[1, 2]
MER-A Spirit ^a	NASA-JPL	2003	Otro	LOCO-I	No	Predicción	Sin Pérd.		[29]
MER-A Spirit ^b	NASA-JPL	2003	Otro	ICER	No	DWT	Con Pérd.		[29]
MER-B Opportunity ^a	NASA-JPL	2003	Otro	LOCO-I	No	Predicción	Sin Pérd.		[29]
MER-B Opportunity ^b	NASA-JPL	2003	Otro	ICER	No	DWT	Con Pérd.		[29]
Meteosat-8 (Canal HRV)	ESA	2002	Otro	JPEG-Baselines	Si	DCT	Con Pérd.		[3, 110]

^a Para compresión sin pérdidas.

^b Para compresión con pérdidas.

Tabla B.3 – *Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (3/5).*

Nombre	Propietario	Año	Uso	Algoritmo de compresión	Estándar	Base	Con / Sin pérdida	CR	Referencias
Meteosat-8 (Canal VIS-IR)	ESA	2002	Otro	Lossless JPEG	Si	Predicción	Sin Pérd.		[3, 110]
MicroLabSat ^a	JAXA	2003	Otro	Lossless JPEG	Si	Predicción	Sin Pérd.		[111]
MicroLabSat ^a	JAXA	2002	Otro	JPEG-Baselines	Si	DCT	Con Pérd.		[111]
MTI	DoE	2000	EO-HR	CCSDS-LDC	Si	Predicción	Sin Pérd.	2,5	[3, 112]
1 OrbView-3	Digital Globe	2003	EO-HR				Con Pérd.	5,5	[113]
PHOBOS	URSS / Fracia	1988	Otro	DCT-DE	No	DCT	Con Pérd.		[32]
PICARD	CNES	2010	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[3]
PLANET B / Nozomi	JAXA	1998	Otro	ICM	No	DCT	Con Pérd.		[1]
Pleiades-HR 1A	CNES	2011	EO-HR	DWT-DE	No	DWT	Con Pérd.	PAN 4,8 MS 4,3	[2]
Pleiades-HR 1B	CNES	2012	EO-HR	DWT-DE	No	DWT	Con Pérd.	PAN 4,8 MS 4,3	[2]
PoSAT-1	Portugal	1993		AMPBTC	No	BTC	Con Pérd.		[3]
Proba-1	ESA	2001	Otro	JPEG-Baselines	Si	DCT	Con Pérd.		[3]
Proba-2 (SWAP)	ESA	2009	Otro	JPEG-Baselines	Si	DCT	Con Pérd.	2,5 a 3	[3, 114]
Proba-2 (X-CAM)	ESA	2009	Otro	LZW	Si	Diccionario	Sin Pérd.		[3, 114]
Proba-V	ESA	2013	EO-MR	CCSDS-IDC	Si	DWT	Opcional		[35, 36, 115]
QuickBird-2	Digital Globe	2001	EO-HR	ADPCM (Kodak)	No	Predicción	Con Pérd.		[11, 20]

^a Para compresión sin pérdidas.

^b Para compresión con pérdidas.

Tabla B.4 – Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (4/5).

Nombre	Propietario	Año	Uso	Algoritmo de compresión	Estándar	Base	Con / Sin pérdida	CR	Referencias
RapidEye-1	Planet Labs	2008	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	2 a 10	[3, 12]
RapidEye-2	Planet Labs	2008	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	2 a 10	[3, 12]
RapidEye-3	Planet Labs	2008	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	2 a 10	[3, 12]
RapidEye-4	Planet Labs	2008	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	2 a 10	[3, 12]
RapidEye-5	Planet Labs	2008	EO-HR	JPEG-Baselines	Si	DCT	Con Pérd.	2 a 10	[3, 12]
RASAT 1	TUBITAK / Turquía	2009	EO-HR	JPEG2000	Si	DWT	Opcional		[98, 99]
Resourcesat-II (LISS-IV)	ISRO	2011	EO-HR	DPCM (ISRO)	No	Predicción	Con Pérd.	1,4	[20]
SAC-C (MMRS)	CONAE	2000	EO-MR	JPEG-Baselines	Si	DCT	Con Pérd.	4 a 10	[43]
Sentinel-2A	ESA	2015	EO-MR	MRCPB- CORECI	No	DWT	Opcional		[116]
SERTS-97	NASA-GSFC	1997	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[101]
SOLAR-B	JAXA	2006	Otro	JPEG-Baselines	Si	DCT	Con Pérd.		[3]
SPOT 1	CNES	1986	EO-HR	DPCM (SPOT)	No	Predicción	Con Pérd.	1,33	[1, 2]
SPOT 2	CNES	1990	EO-HR	DPCM (SPOT)	No	Predicción	Con Pérd.	1,33	[1, 2]
SPOT 3	CNES	1993	EO-HR	DPCM (SPOT)	No	Predicción	Con Pérd.	1,33	[1, 2]
SPOT 4	CNES	1998	EO-HR	DPCM (SPOT)	No	Predicción	Con Pérd.	1,33	[1, 2]
SPOT 5	CNES	2002	EO-HR	DCT-DE	No	DCT	Con Pérd.	2,81	[2, 4]
SPOT 6	CNES / Airbus Group Astrium	2012	EO-HR	MRCPB- CORECI	No	DWT	Opcional		[40, 117]

^a Para compresión sin pérdidas.^b Para compresión con pérdidas.

Tabla B.5 – Lista de misiones espaciales que aplicaron compresión de imágenes a bordo (5/5).

Nombre	Propietario	Año	Uso	Algoritmo de compresión	Estándar	Base	Con / Sin pérdida	CR	Referencias
SPOT 7	CNES / Airbus Group Astrium	2014	EO-HR	MRCPB-CORECI	No	DWT	Opcional		[40, 117]
SUNSAT	Sudafrica	1999	Otro	JPEG-Baselines	Si	DCT	Con Pérd.		[3]
SWAS	NASA-GSFC	1999	Otro	CCSDS-LDC	Si	Predicción	Sin Pérd.		[101]
TEAMSAT	ESA	1997	Otro	JPEG-Baselines	Si	DCT	Con Pérd.		[3]
TiungSAT	Malasia	2000		Improved AMPBTC	No	BTC	Con Pérd.		[3]
TRACE	NASA-GSFC	1998	Otro	JPEG-Baselines	Si	DCT	Con Pérd.		[118]
Tsinghua-1	Universidad Tsinghua	2000		AMPBTC	No	BTC	Con Pérd.		[3]
WorldView-1	Digital Globe	2007	EO-HR	ADPCM (Kodak)	No	Predicción	Con Pérd.	2,75	[3, 11, 20]
Yaogan 21	China	2014		JPEG2000	Si	DWT	Opcional		[119]
Yaogan 8	China	2009	EO-HR	JPEG2000	Si	DWT	Opcional		[119]
Ziuyan 3	China	2012	EO-HR	JPEG2000	Si	DWT	Opcional		[119]

^a Para compresión sin pérdidas.^b Para compresión con pérdidas.

Metodología de simulación de un borde objetivo

En la Fig. C.1 se puede ver el diagrama de flujo del algoritmo implementado en MATLAB para la simulación de un borde objetivo. Cabe destacar que no todos los procesos de la cadena de detección han sido tenidos en cuenta, ésta es una simulación reducida que permite obtener imágenes degradadas en frecuencia acorde a lo descrito en [60, 71] y con ruido acorde a lo descrito en [120]. No obstante se tuvieron en cuenta los factores más importantes y son útiles a fines de desarrollar el algoritmo para la estimación del NIIRS.

En la Sección C.1 se describe cada uno de los pasos del algoritmo de simulación del borde objetivo. Y en la Sección C.2 se describe cómo se calculan los valores que se consideran reales y son usados para contrastar con los estimados.

C.1. Algoritmo de simulación de un borde objetivo

C.1.1. Generación del borde objetivo ideal

En este paso se genera una imagen del tamaño, $l_x \times l_y$, inclinación de borde respecto a la vertical, θ_{borde} , y reflectancia de la zona oscura, ρ_{low} , y de la zona clara, ρ_{high} deseadas, es decir,

$$se_0(x, y) \begin{cases} \rho_{low} & \text{si } y < a \cdot x + b; \\ \rho_{high} & \text{si } y \geq a \cdot x + b; \end{cases} \quad \begin{cases} \text{para } 0 \leq x \leq l_x; \\ 0 \leq y \leq l_y; \end{cases} \quad (C.1)$$

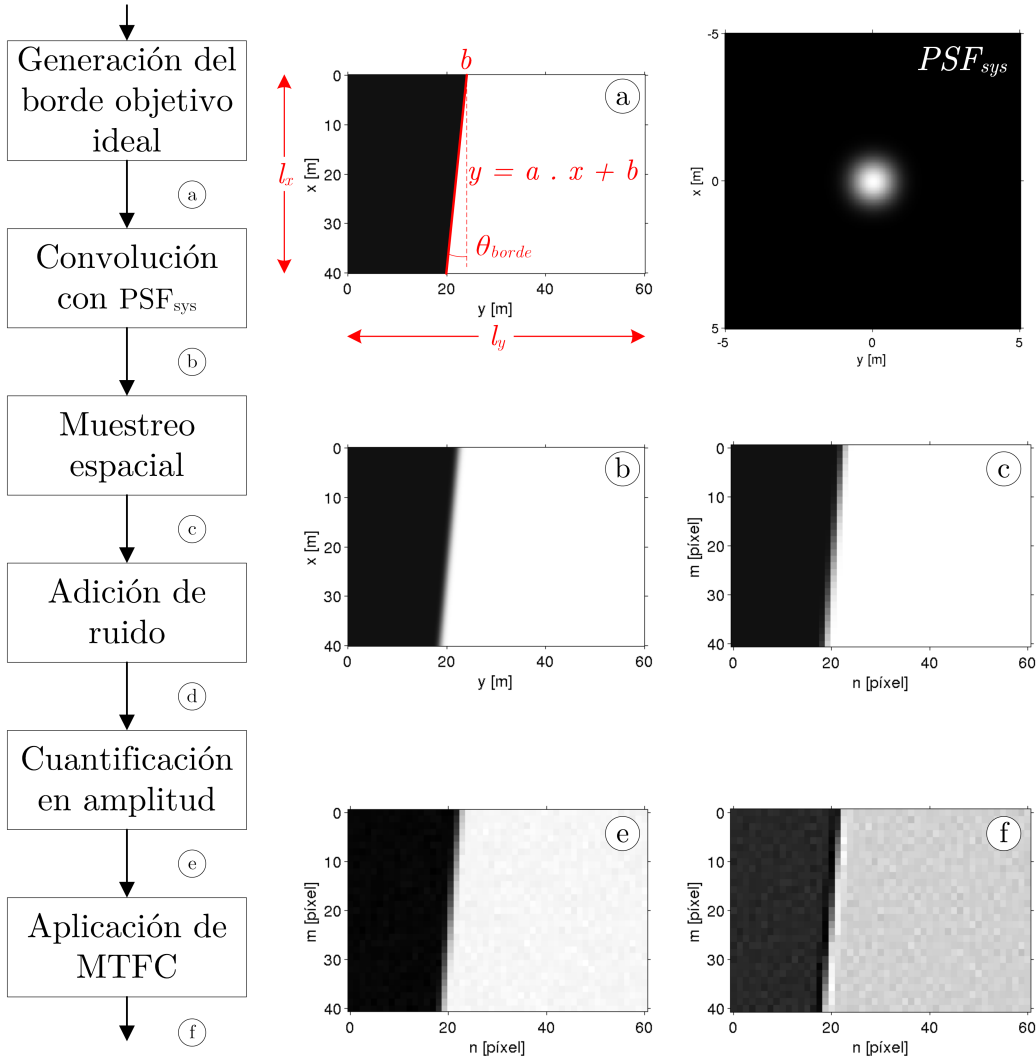


Fig. C.1. Diagrama de bloques de simulación de borde objetivo.

donde $y = a \cdot x + b$ es la recta que define el borde, $a = \tan(\theta_{borde})$ y b se elige arbitrariamente. Luego, mediante un cálculo radiométrico que contempla las características del instrumento, se obtienen las intensidades en unidades de fotones que inciden sobre cada píxel para cada zona. Entonces, la imagen ideal sin degradaciones queda dada por

$$se_1(x, y) \begin{cases} I_{ph} | \rho_{low} & \text{si } y < a \cdot x + b; \\ I_{ph} | \rho_{high} & \text{si } y \geq a \cdot x + b; \end{cases} \quad \begin{cases} \text{para } 0 \leq x \leq l_x; \\ 0 \leq y \leq l_y; \end{cases} \quad (C.2)$$

donde $I_{ph} | \rho_{low}$ y $I_{ph} | \rho_{high}$ son las intensidades en la zona oscura y clara respectivamente. En la Fig. C.1(a) se muestra un ejemplo que corresponden a un borde objetivo de $l_x = 40m$, $l_y = 60m$, con un borde inclinado $\theta_{borde} = 6^\circ$ y con reflectancias $\rho_{low} = 3,5\%$ y $\rho_{high} = 52\%$.

A continuación se resume el cálculo radiométrico para la obtención de $I_{ph}|_{\rho_{low}}$ y $I_{ph}|_{\rho_{high}}$ a partir de las reflectancias y las características del instrumento [62]. Se realizó utilizando especificaciones de: ancho de banda, BW , longitud de onda media la banda, λ_{med} , número f del sistema óptico ¹, $F\#$, transmitancia del sistema óptico, τ_{op} , tamaño del píxel, p , y tiempo de integración, T_i ; de la banda PAN de una cámara satelital de alta resolución.

Se utilizó un modelo de transmisión simplificado [67] (que supone la ausencia de atmósfera) en el que la radiancia espectral que llega al instrumento está dada por

$$L_{inst,\lambda} = \rho \cdot L_{sol,\lambda}, \quad (C.3)$$

donde $L_{sol,\lambda}$ es la radiancia espectral del sol que llega a la tierra y ρ la reflectancia de la superficie terrestre, que puede ser ρ_{low} o ρ_{high} . Considerando el ancho de banda BW , se calcula la radiancia por

$$L_{inst} = L_{inst,\lambda} \cdot BW. \quad (C.4)$$

Conociendo el número f del sistema óptico, $F\#$, y la transmitancia del sistema óptico, τ_{op} , se calcula la irradiancia en el plano focal

$$I_{FPA} = \frac{\pi \cdot L_{inst}}{(4 \cdot F\#^2 + 1)} \cdot \tau_{op}. \quad (C.5)$$

Dado el tamaño del píxel, p , y la energía media de una fotón, U_{ph} , para la longitud de onda λ_{med} , se obtiene la cantidad de fotones colectados por unidad de tiempo por cada píxel

$$\Phi_{ph} = \frac{I_{FPA} \cdot p^2}{U_{ph}} = \frac{I_{FPA} \cdot p^2}{\frac{h \cdot c}{\lambda_{med}}}, \quad (C.6)$$

donde h es la constante de Plank y c la velocidad de la luz en el vacío. Considerando el tiempo de integración, T_i , la cantidad de fotones colectados por cada píxel esta dado por

$$I_{ph} = round(\Phi_{ph} \cdot T_i), \quad (C.7)$$

donde el redondeo surge ya que la cantidad de fotones debe ser una cantidad discreta [120].

¹dado por la relación entre la distancia focal y el diámetro de la apertura

C.1.2. Convolución con PSF_{sys}

La PSF_{sys} se modeló a partir de datos genéricos para un instrumento de alta resolución de $GSD = 1m$ considerando la contribución del sistema óptico, el detector, movimiento de la imagen y efectos de la electrónica; más detalles pueden encontrarse en [60, 71]. Matemáticamente la PSF_{sys} se expresa como la convolución de sus componentes, es decir,

$$PSF_{sys}(x, y) = PSF_{op}(x, y) * PSF_{det}(x, y) * PSF_{mov}(x, y) * PSF_{elec}(x, y). \quad (C.8)$$

Dada la $PSF_{sys}(x, y)$, se realiza la convolución de ésta con el borde objetivo ideal

$$se_2(x, y) = PSF_{sys}(x, y) * se_1(x, y), \quad (C.9)$$

y se obtiene el borde objetivo filtrado, donde la transición deja de ser abrupta como se ilustra en la Fig. C.1(b).

C.1.3. Muestreo espacial

Siguiendo en la cadena de detección, se tiene la discretización espacial que produce el detector. Considerando que llega una señal continua, ésta se muestrea cada una distancia de $GSD = 1m$,

$$se_3[n, m] = se_2(m.GSD, n.GSD); \quad m = \{0, 1, \dots, M-1\}; \quad M = \lfloor l_x / GSD \rfloor; \quad (C.10) \\ n = \{0, 1, \dots, N-1\}; \quad N = \lfloor l_y / GSD \rfloor;$$

donde m y n corresponden a la posición de píxel en el sentido de las filas y las columnas respectivamente. En la Fig. C.1.(c) puede verse el borde objetivo muestreado.

C.1.4. Adición de ruido

Una vez que los fotones llegan al detector se produce la detección mediante la cual el éste convierte los fotones a electrones. En este proceso se añade ruido a la señal. Para estas simulaciones sólo se consideraron las fuentes de ruido más importantes, el ruido *shot* debido a la señal (*photon shot noise*), el ruido *shot* debido a la corriente oscura (*dark current shot noise*), y el ruido de lectura (*readout noise*). El ruido *shot* tiene una distribución de Poisson y el ruido de lectura una distribución Gaussiana [121]. En esta etapa se emplearon especificaciones de eficiencia cuántica, QE , corriente oscura, I_{DC} , y desviación

estándar del ruido de lectura, $\sigma_{readout}$; de un detector usado para cámaras de alta resolución.

El ruido *shot* debido a la señal se modeló adicionando a la señal un ruido con distribución de Poisson con media igual al valor de la señal

$$se_{4ph}[n, m] = \mathcal{P}(se_3[n, m]), \quad (C.11)$$

donde $\mathcal{P}(\Lambda)$ representa un generador de ruido con distribución de Poisson con media Λ y se_{4ph} es la señal más este ruido en unidad de fotones. Para obtener la señal equivalente en electrones basta multiplicar a esta última por la eficiencia cuántica del detector, QE , para la cual se asumió un valor medio para todas las longitudes de onda de la banda. Entonces, se obtiene

$$se_{4e}[n, m] = QE \cdot se_{4ph}[n, m]. \quad (C.12)$$

Dada la especificación de corriente oscura del detector, I_{DC} , el ruido por corriente oscura se modeló como un ruido con distribución de Poisson con media igual $I_{DC} \cdot T_i$, es decir,

$$N_{shotDC}[n, m] = \mathcal{P}(I_{DC} \cdot T_i). \quad (C.13)$$

El ruido de lectura se modeló adicionando un ruido con distribución Gaussiana de media cero y desviación estándar $\sigma_{readout}$, es decir,

$$N_{readout}[n, m] = \mathcal{G}(0, \sigma_{readout}), \quad (C.14)$$

donde $\mathcal{G}(\mu, \sigma)$ representa un generador de ruido con distribución Gaussiana con media μ y desviación estándar σ .

Finalmente se suman todas las componentes y se obtiene la señal que representa a el borde objetivo con los ruidos que agrega el detector, es decir,

$$se_4[n, m] = se_{4pe}[n, m] + N_{shotDC}[n, m] + N_{readout}[n, m]. \quad (C.15)$$

Más detalles sobre la simulación de la cadena de ruido del detector pueden verse en [120].

C.1.5. Cuantificación en amplitud

La tensión a la salida del detector se convierte a números digitales (DN) mediante una conversión analógica a digital. Esto naturalmente añade un ruido de cuantificación con distribución uniforme con media $\mu_Q = 0$ y desviación estándar $\sigma_Q = \frac{2^B}{\sqrt{12}}$, donde B es la cantidad de bits utilizados para la cuantificación, para este caso $B = 10bits$. Luego de este paso, se obtiene la imagen tal como la adquiriría el instrumento, que se denomina $se_5[n, m]$ y puede verse en la Fig. C.1(e).

C.1.6. Aplicación de MTFC

Finalmente, si se desea considerar el posprocesamiento aplicado a la imagen en tierra, puede aplicarse la compensación de la MTF (MTFC) por medio de la convolución de la imagen con el núcleo del filtro, es decir,

$$se_6[x, y] = se_5[n, m] * Kernel_{MTFC}. \quad (C.16)$$

Para éstas simulaciones se aplicó la MTFC cuyo núcleo está dado por

$$Kernel_{MTFC} = \begin{bmatrix} c & b & c \\ b & a & b \\ c & b & c \end{bmatrix} \quad (C.17)$$

y su respuesta en frecuencia por

$$MTFC(u, v) = a + 2.b.[\cos(2.\pi.u) + \cos(2.\pi.v)] + 4.c.\cos(2.\pi.u).\cos(2.\pi.v), \quad (C.18)$$

con $a = 2,707$, $b = -0,3536$ y $c = -0,0732$ [62]. La imagen resultante puede verse en la Fig. C.1(f) donde gracias al realce aplicado para mostrar la imagen se hace evidente el rebasamiento de borde.

C.2. Obtención de posición del borde, RER, H, y SNR reales

La posición real del borde queda determinada por la recta que se utilizó para su construcción en (C.2).

Dada la PSF del sistema de imagen (C.8) y la función de transferencia del posprocesamiento (C.18), la PSF neta está dada por la convolución entre

éstas (3.21). Las ESF en cada una de las direcciones, $ESF_x(x)$ y $ESF_y(y)$, resultan de integrar la PSF como indica la ecuación (3.18) en la dirección x e y respectivamente. Finalmente el RER y H en cada dirección se calculan a partir de $ESF_x(x)$ y $ESF_y(y)$ acorde a lo mencionado en la Sección 3.2.3.

Dado el borde objetivo final, que puede contar o no con el posprocesamiento, la relación señal a ruido se calcula aplicando la métrica (3.31) basada en la imagen, que se repite aquí por conveniencia,

$$SNR = \frac{S_{target}}{\sigma_{noise}} = \frac{\overline{DN_{\rho_{high}}} - \overline{DN_{\rho_{low}}}}{(\sigma_{\rho_{high}} + \sigma_{\rho_{low}})/2}. \quad (C.19)$$

Dada la posición real del borde, la zona de ρ_{low} se define como todos los píxeles a la izquierda del borde que distan al menos 5 píxeles de éste y la zona de ρ_{high} se define como todos los píxeles a la derecha del borde que distan al menos 5 píxeles de éste, como se ilustra en la Fig. C.2.

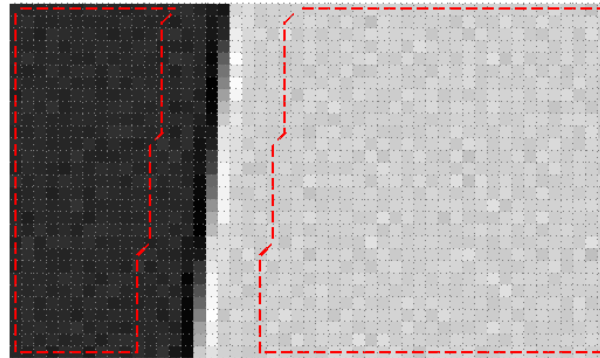


Fig. C.2. Zonas para el cálculo de SNR real en al borde objetivo simulado.

Desarrollo de (4.15)

Dada la función de transferencia de la MTFC (C.18), en una única dirección,

$$\begin{aligned}MTFC(0, v) &= a + 2.b.[1 + \cos(2.\pi.v)] + 4.c.\cos(2.\pi.v), \\ &= (a+2.b) + (2.b+4.c).\cos(2.\pi.v),\end{aligned}\tag{D.1}$$

se desea encontrar su transformada inversa de Fourier. La transformada directa de Fourier del coseno, esta dada por

$$x(t) = \cos(t) \xrightarrow{\mathcal{F}} X^F(\omega) = \pi.[\delta(\omega-1) + \delta(\omega+1)],\tag{D.2}$$

y aplicando la propiedad de dualidad

$$X^F(t) = \pi.[\delta(t-1) + \delta(t+1)] \xrightarrow{\text{dualidad}} 2.\pi.x(-\omega) = 2.\pi.\cos(-\omega),\tag{D.3}$$

puede encontrarse la transformada inversa de Fourier de una función coseno

$$Y^F(\omega) = \cos(-\omega) \xrightarrow{\mathcal{F}^{-1}} y(t) = \frac{1}{2}.[\delta(t-1) + \delta(t+1)].\tag{D.4}$$

Entonces, por analogía y considerando que $\omega \equiv 2.\pi.v$ y $t \equiv y$, la transformada inversa de Fourier de (D.1) queda dada por,

$$\begin{aligned}\mathcal{F}^{-1}\{MTFC(u, 0)\} &= (a+2.b).\mathcal{F}^{-1}\{1\} + (2.b+4.c).\mathcal{F}^{-1}\{\cos(2.\pi.v)\} \\ &= (a+2.b).\delta(y) + \frac{(2.b+4.c)}{2}.[\delta(y-1) + \delta(y+1)] \\ &= k_1.\delta(y) + k_2.\delta(y-1) + k_2.\delta(y+1),\end{aligned}\tag{D.5}$$

donde $k_1 = (a+2.b)$ y $k_2 = (b+2.c)$.

Histogramas de RER, H y SNR para imágenes de validación

En las Figs. E.1 a E.11, pueden verse los histogramas de los parámetros RER, H y SNR para las imágenes de la tabla Tabla 4.3 utilizadas para la validación del algoritmo de estimación del NIIRS. Donde μ y σ son la media y desviación estándar antes de eliminar los bordes atípicos, y μ' es la media luego de eliminarlos.

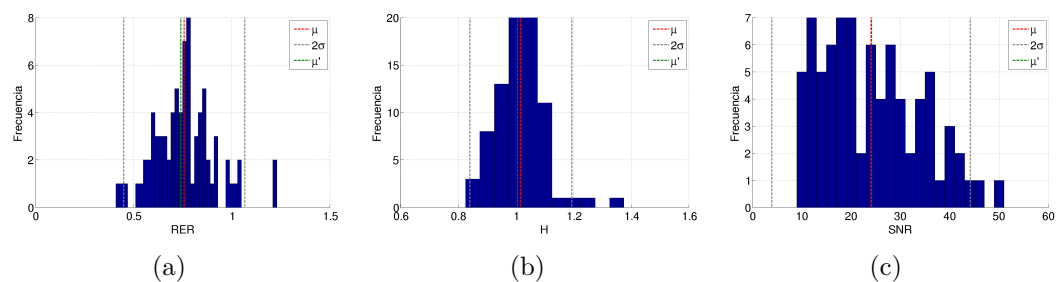


Fig. E.1. Estimación de NIIRS en imagen GE1-01-P, histograma de (a) RER, (b) H y (c) SNR.

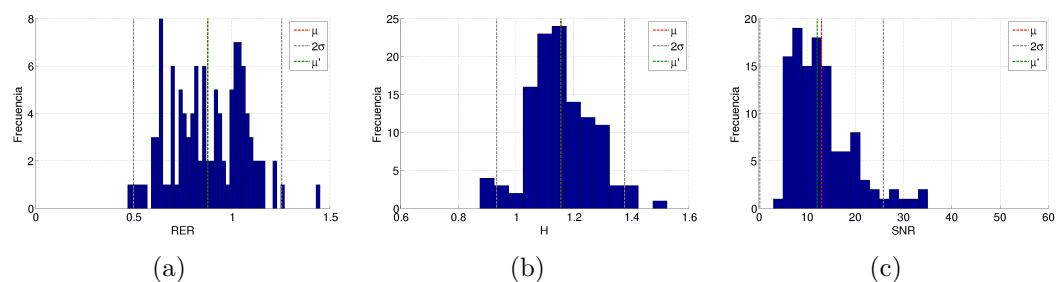


Fig. E.2. Estimación de NIIRS en imagen GE1-01-M, histograma de (a) RER, (b) H y (c) SNR.

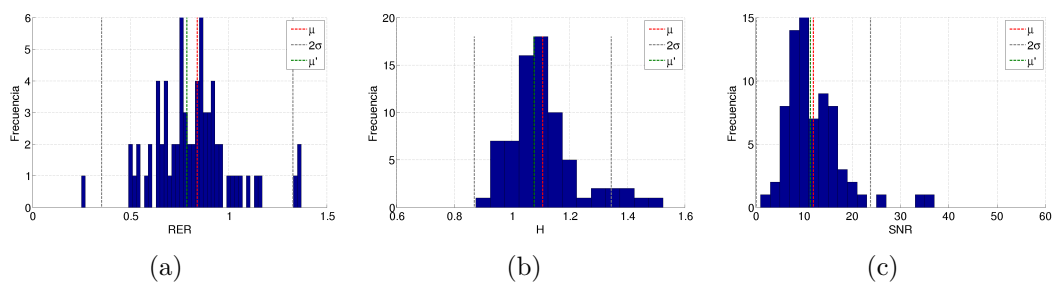


Fig. E.3. Estimación de NIIRS en imagen QB2-01-P, histograma de (a) RER, (b) H y (c) SNR.

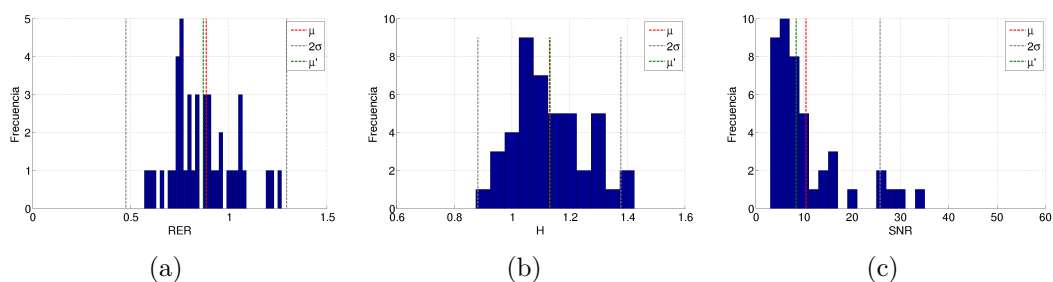


Fig. E.4. Estimación de NIIRS en imagen QB2-01-M, histograma de (a) RER, (b) H y (c) SNR.

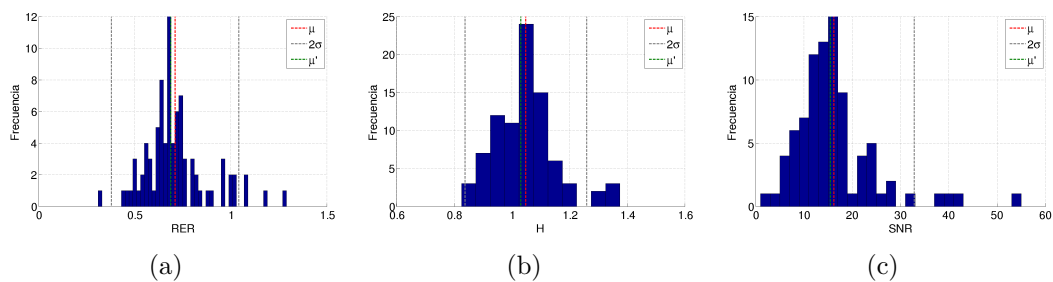


Fig. E.5. Estimación de NIIRS en imagen IK2-01-P, histograma de (a) RER, (b) H y (c) SNR.

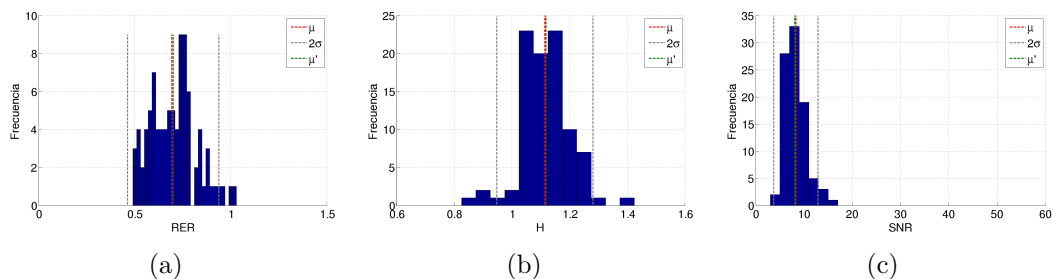


Fig. E.6. Estimación de NIIRS en imagen IK2-02-P, histograma de (a) RER, (b) H y (c) SNR.

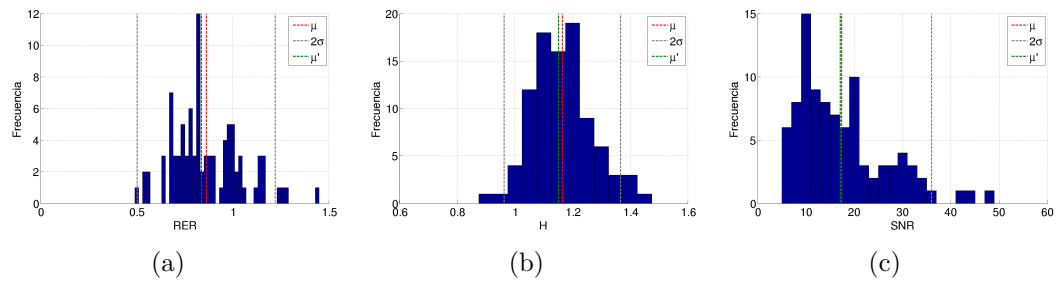


Fig. E.7. Estimación de NIIRS en imagen WV1-01-P, histograma de (a) RER, (b) H y (c) SNR.

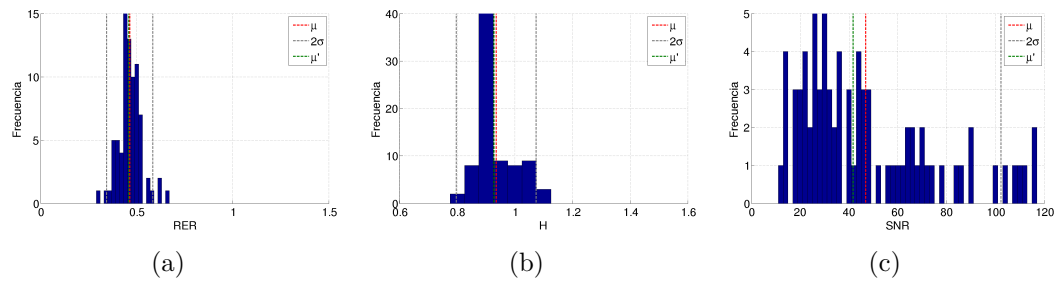


Fig. E.8. Estimación de NIIRS en imagen WV2-01-P, histograma de (a) RER, (b) H y (c) SNR.

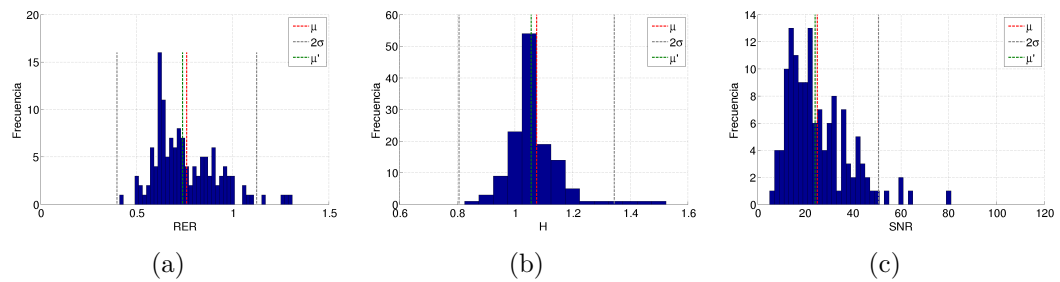


Fig. E.9. Estimación de NIIRS en imagen WV3-01-P, histograma de (a) RER, (b) H y (c) SNR.

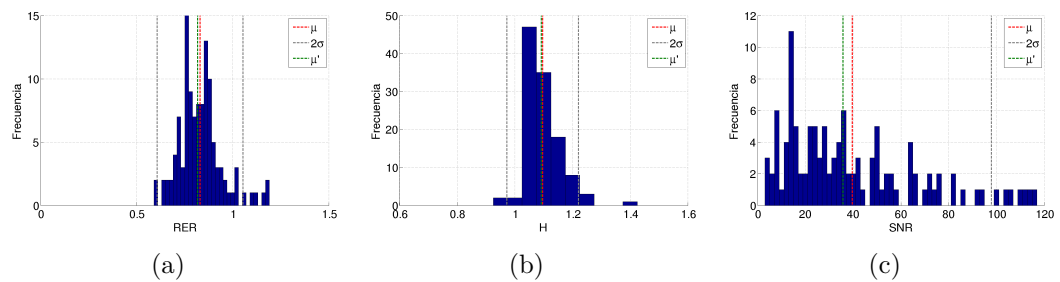


Fig. E.10. Estimación de NIIRS en imagen WV3-01-M, histograma de (a) RER, (b) H y (c) SNR.

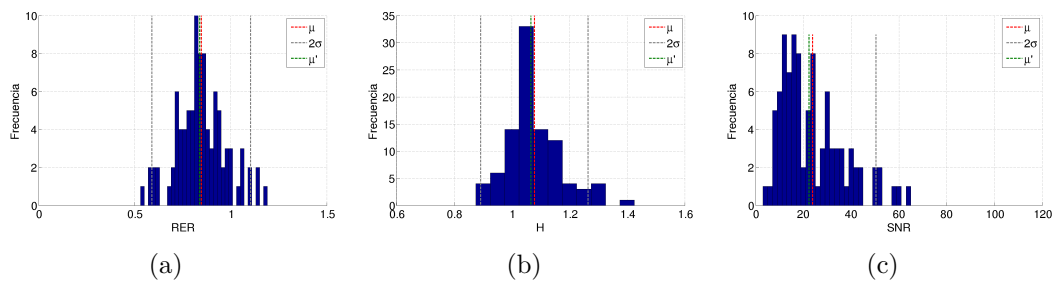


Fig. E.11. Estimación de NIIRS en imagen ERB-01-P, histograma de (a) RER, (b) H y (c) SNR.

Estimación de NIIRS con método 2 para reconstrucción de ESF

Tabla F.1 – Resultados de validación de NIIRS (método 2 para la reconstrucción de la ESF).

Código	$NIIRS_R$	N_b	GSD [in]	\overline{RER}_E	\overline{H}_E	\overline{SNR}_E	$NIIRS_E$	$\Delta NIIRS$
GE1-01-P	5,40	79	17,09	0,7771	1,01	24,2	5,37	-0,03
GE1-01-M	3,10	116	96,54	0,9372	1,02	12,2	2,87	-0,23
QB2-01-P	4,70	73	24,53	0,8695	1,04	11,3	4,98	0,28
QB2-01-M	2,70	44	98,03	0,9217	1,07	8,4	2,84	0,14
IK2-01-P	4,28	86	32,69	0,7162	1,01	15,5	4,37	0,09
IK2-02-P	3,81	91	45,96	0,7281	1,07	8,1	3,87	0,06
WV1-01-P	4,90	93	21,57	0,8932	1,13	17,1	5,14	0,24
WV2-01-P	5,00	79	18,66	0,4522	0,91	41,8	4,66	-0,34
WV3-01-P	5,60	135	12,64	0,7851	1,04	24,1	5,78	0,18
WV3-01-M	3,6	116	50,55	0,8718	1,07	35,7	3,98	0,38
ERB-01-P	4,60	95	26,38	0,8768	1,06	22,4	4,89	0,29

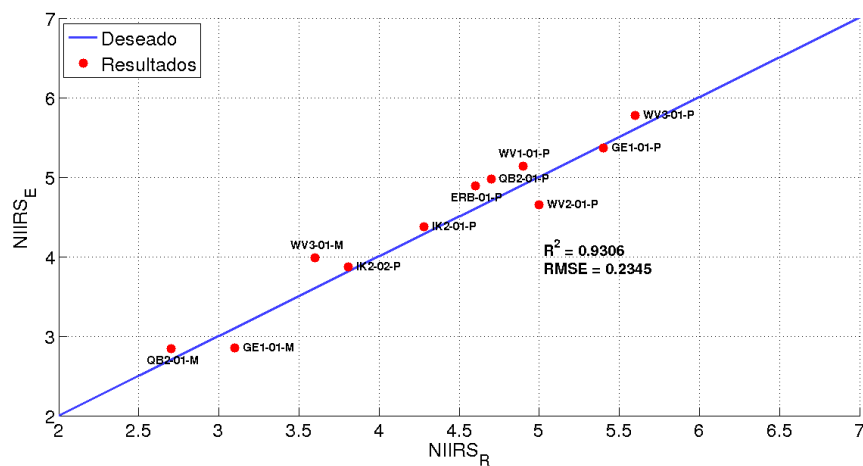


Fig. F.1. Resultados de estimación de NIIRS vs. NIIRS Real (método 2 para la reconstrucción de la ESF).

Resultados de evaluación de JPEG-LS

A continuación se presentan todos los resultados de los casos de prueba de JPEG-LS mostrados en la Tabla 5.8. Aunque la información de cada caso está en la misma aquí se indica el conjunto de datos de prueba.

Caso de prueba JLS_01: Imágenes PAN urbanas.

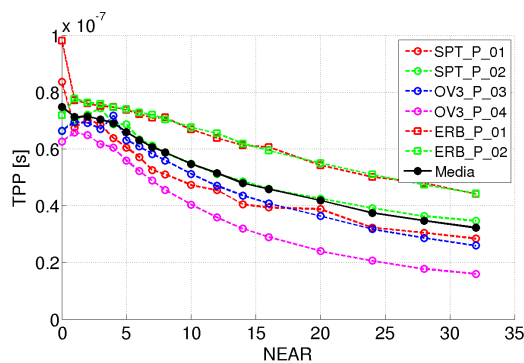


Fig. G.1. TPP vs. $NEAR$ para JLS_01 .

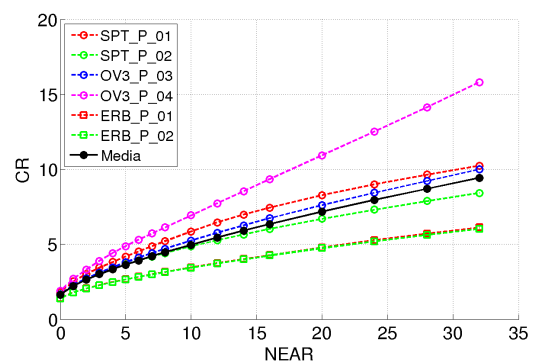


Fig. G.2. CR vs. $NEAR$ para JLS_01 .

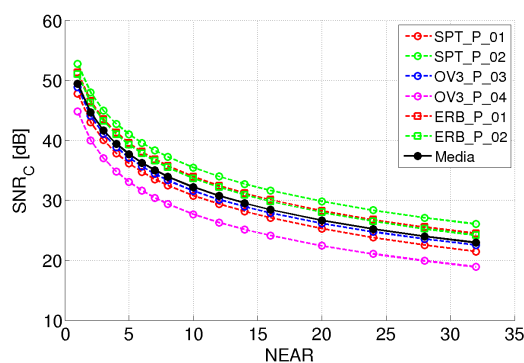


Fig. G.3. SNR_C vs. $NEAR$ para JLS_01 .

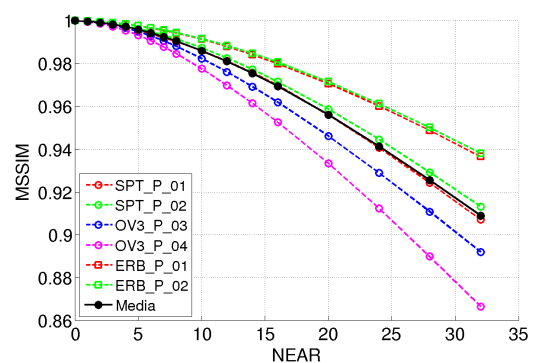


Fig. G.4. $MSSIM$ vs. $NEAR$ para JLS_01 .

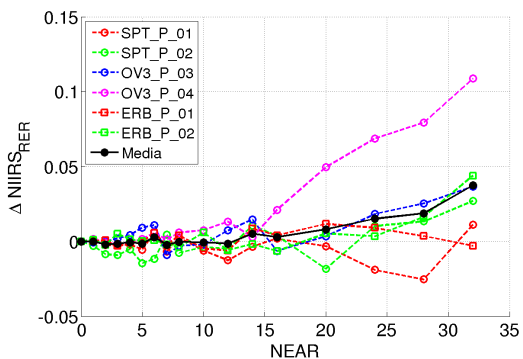


Fig. G.5. $\Delta NIIRS_{RER}$ vs. NEAR para JLS_01.

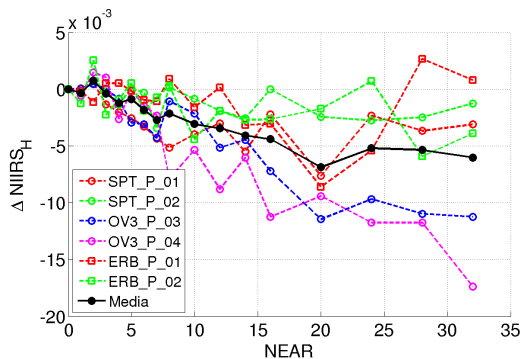


Fig. G.6. $\Delta NIIRS_H$ vs. NEAR para JLS_01.

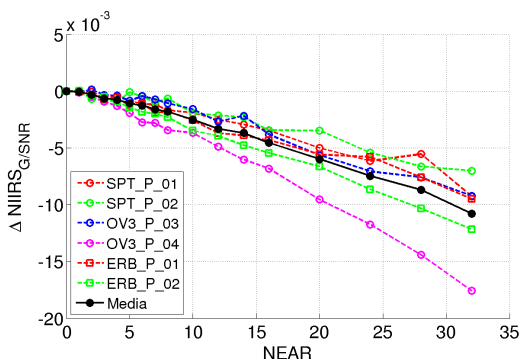


Fig. G.7. $\Delta NIIRS_{SNR}$ vs. NEAR para JLS_01.

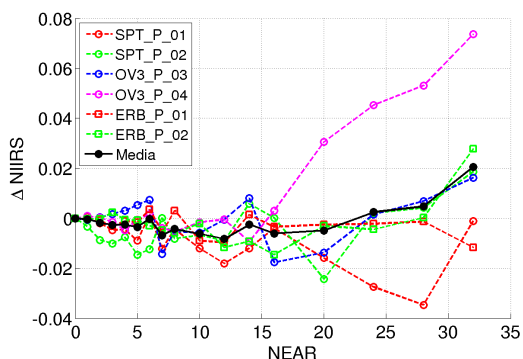


Fig. G.8. $\Delta NIIRS$ vs. NEAR para JLS_01.

Caso de prueba JLS_21: Imágenes PAN rurales.

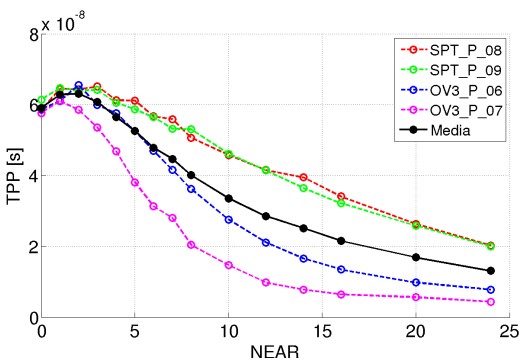


Fig. G.9. TPP vs. NEAR para JLS_21.

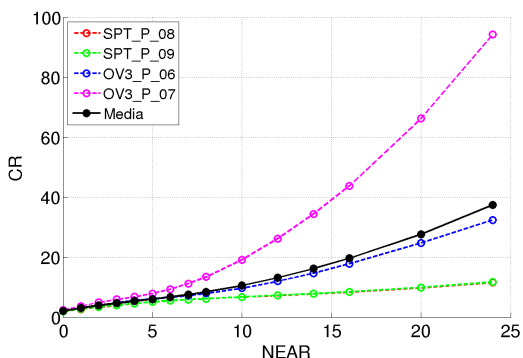


Fig. G.10. CR vs. NEAR para JLS_21.

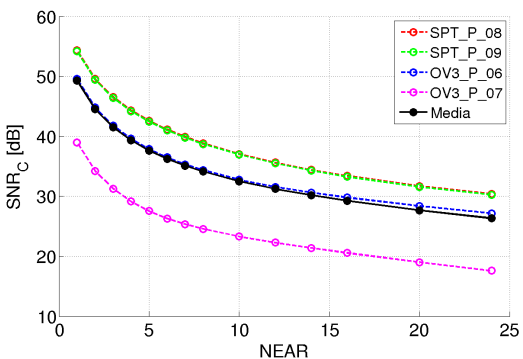


Fig. G.11. SNR_C vs. NEAR para JLS_21.

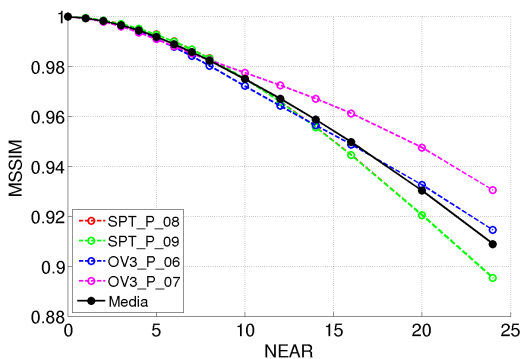


Fig. G.12. MSSIM vs. NEAR para JLS_21.

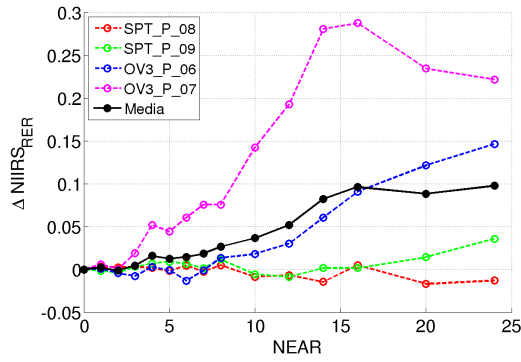


Fig. G.13. $\Delta NIIRS_{RER}$ vs. $NEAR$ para JLS_{21} .

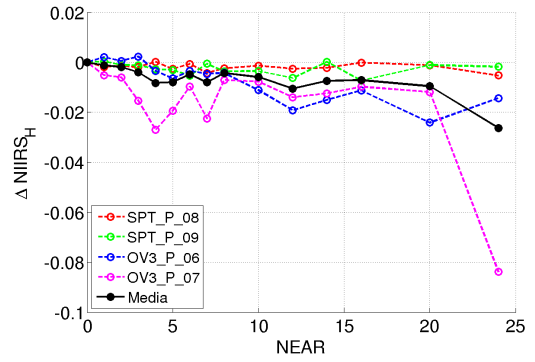


Fig. G.14. $\Delta NIIRS_H$ vs. $NEAR$ para JLS_{21} .

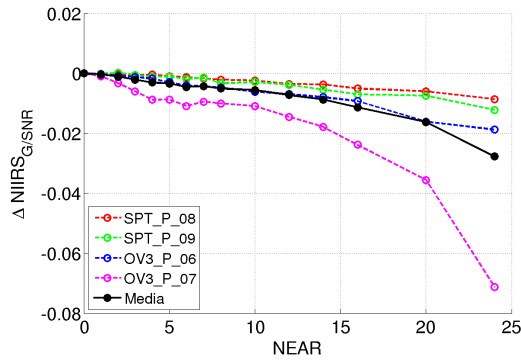


Fig. G.15. $\Delta NIIRS_{SNR}$ vs. $NEAR$ para JLS_{21} .

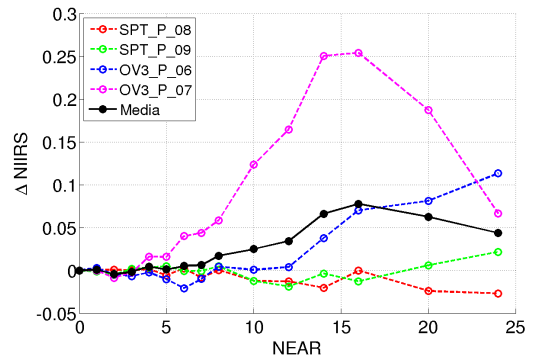


Fig. G.16. $\Delta NIIRS$ vs. $NEAR$ para JLS_{21} .

Caso de prueba JLS_31: Imágenes MS urbanas.

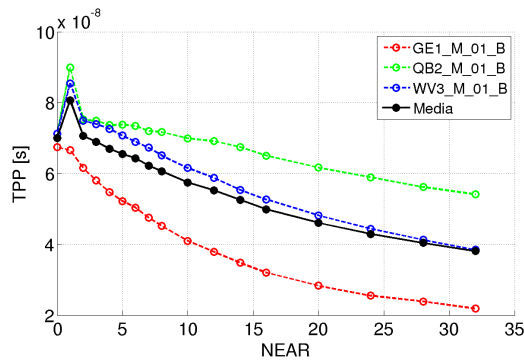


Fig. G.17. TPP vs. $NEAR$ para JLS_{31} .

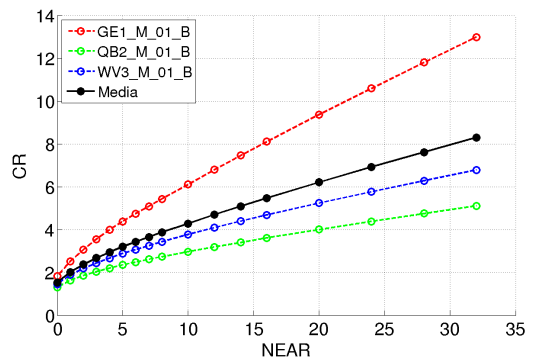


Fig. G.18. CR vs. $NEAR$ para JLS_{31} .

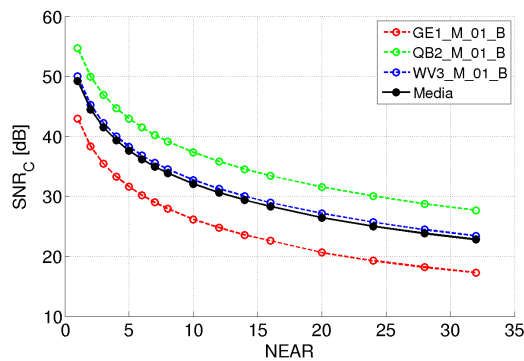


Fig. G.19. SNR_C vs. $NEAR$ para JLS_{31} .

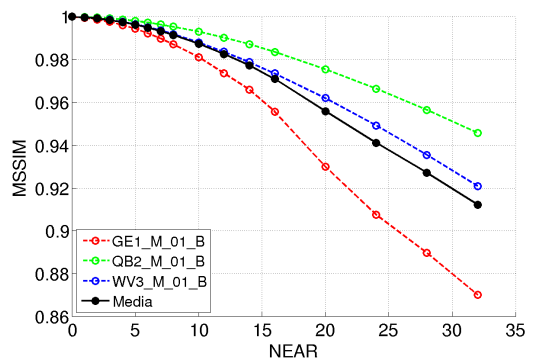


Fig. G.20. $MSSIM$ vs. $NEAR$ para JLS_{31} .

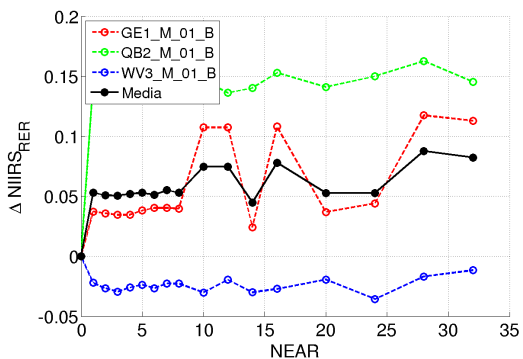


Fig. G.21. $\Delta NIIRS_{RER}$ vs. $NEAR$ para JLS_{31} .

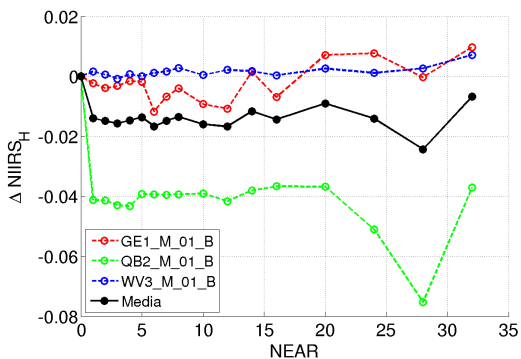


Fig. G.22. $\Delta NIIRS_H$ vs. $NEAR$ para JLS_{31} .

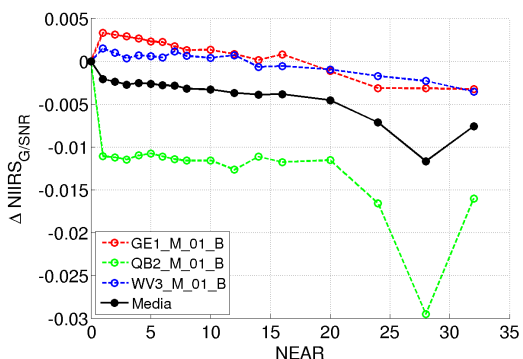


Fig. G.23. $\Delta NIIRS_{SNR}$ vs. $NEAR$ para JLS_{31} .

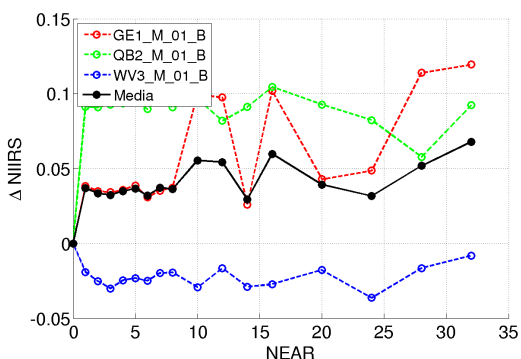


Fig. G.24. $\Delta NIIRS$ vs. $NEAR$ para JLS_{31} .

Caso de prueba JLS_41: Imágenes MS rurales.

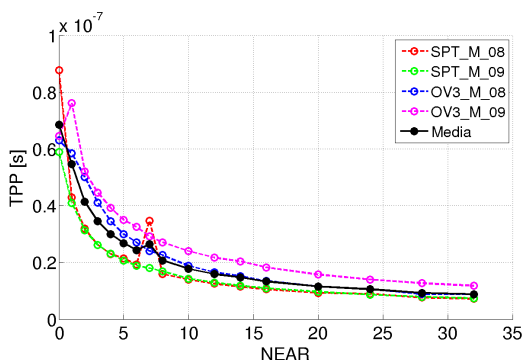


Fig. G.25. TPP vs. $NEAR$ para JLS_{41} .

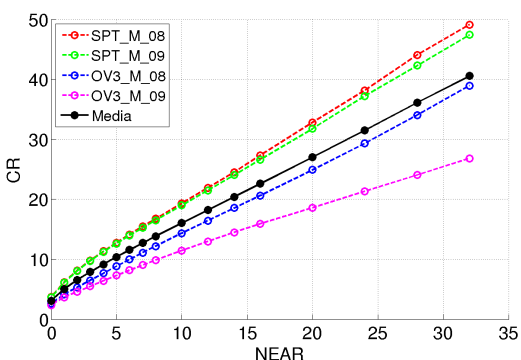


Fig. G.26. CR vs. $NEAR$ para JLS_{41} .

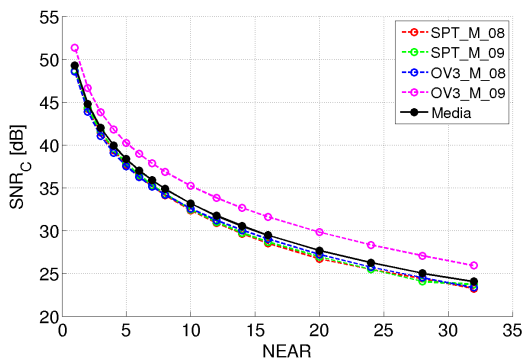


Fig. G.27. SNR_C vs. $NEAR$ para JLS_{41} .

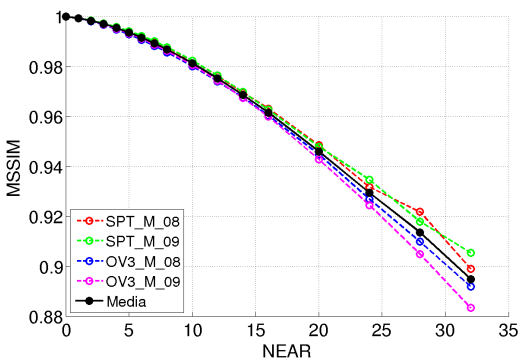


Fig. G.28. $MSSIM$ vs. $NEAR$ para JLS_{41} .

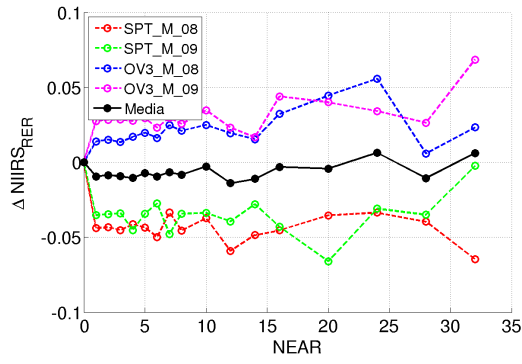


Fig. G.29. $\Delta NIIRS_{RER}$ vs. $NEAR$ para JLS_{41} .

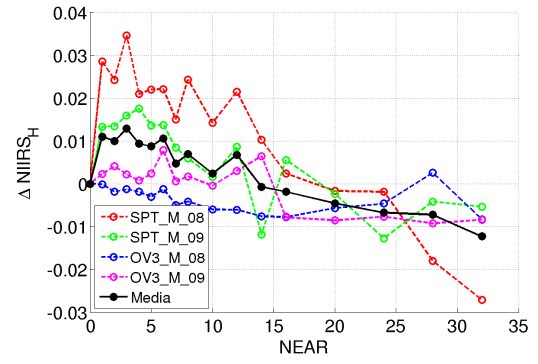


Fig. G.30. $\Delta NIIRS_H$ vs. $NEAR$ para JLS_{41} .

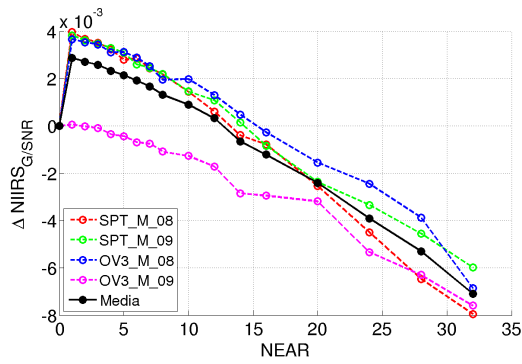


Fig. G.31. $\Delta NIIRS_{GSNR}$ vs. $NEAR$ para JLS_{41} .

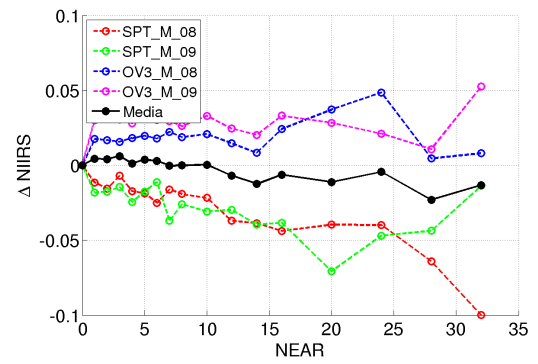


Fig. G.32. $\Delta NIIRS$ vs. $NEAR$ para JLS_{41} .

Resultados de evaluación de JPEG-Baselines

A continuación se presentan todos los resultados de los casos de prueba de JPEG-Baselines mostrados en la Tabla 5.10. Aunque la información de cada caso está en la misma aquí se indica la tabla de cuantificación usada y el conjunto de datos de prueba.

Caso de prueba JPG_01: Tabla *Default*, imágenes PAN urbanas.

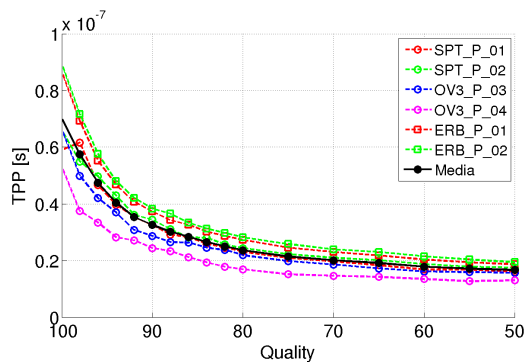


Fig. H.1. *TPP vs. Quality para JPG_01.*

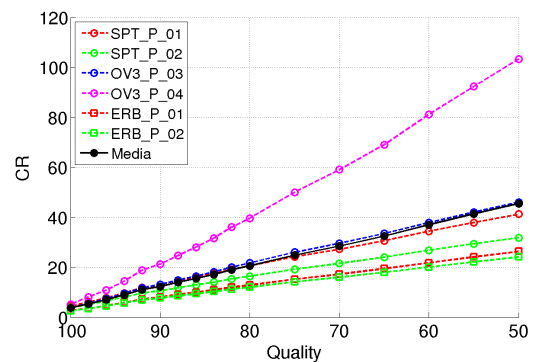


Fig. H.2. *CR vs. Quality para JPG_01.*

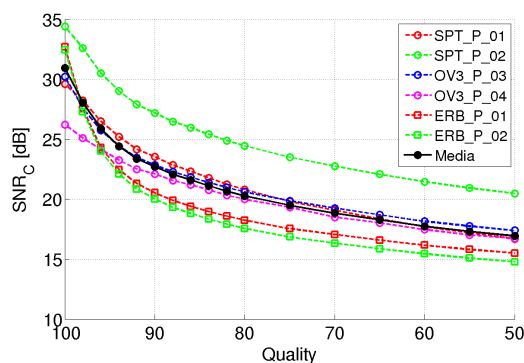


Fig. H.3. *SNR_C vs. Quality para JPG_01.*

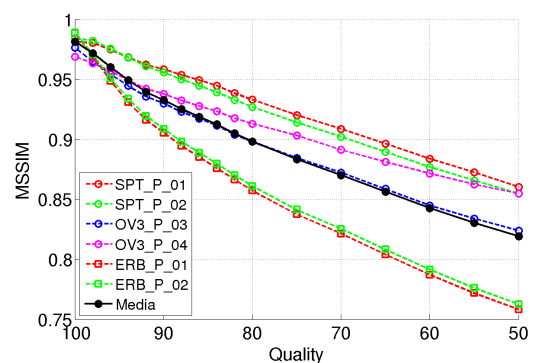


Fig. H.4. *MSSIM vs. Quality para JPG_01.*

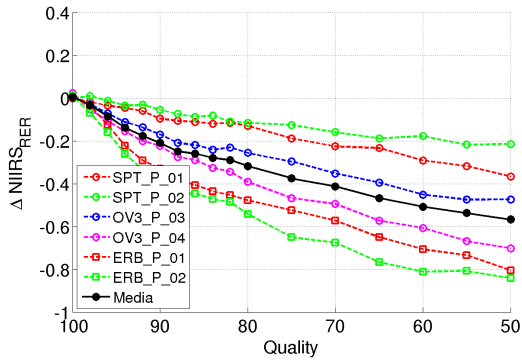


Fig. H.5. $\Delta NIIRS_{RER}$ vs. Quality para JPG_01.

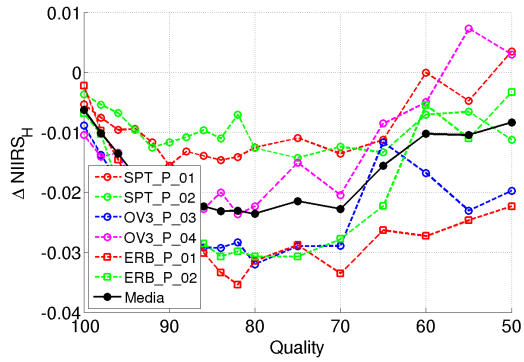


Fig. H.6. $\Delta NIIRS_H$ vs. Quality para JPG_01.

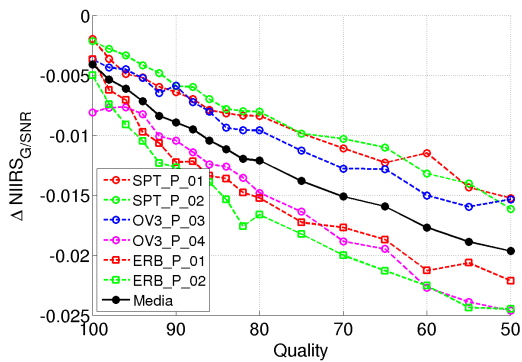


Fig. H.7. $\Delta NIIRS_{G/SNR}$ vs. Quality para JPG_01.

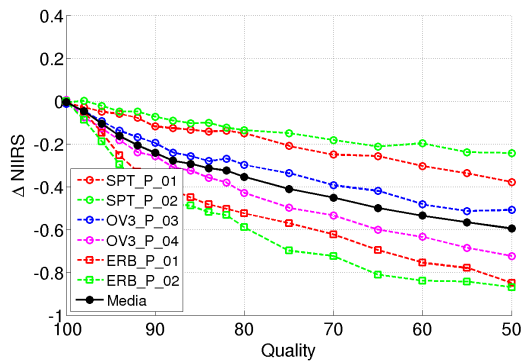


Fig. H.8. $\Delta NIIRS$ vs. Quality para JPG_01.

Caso de prueba JPG_02: Tabla GWG, imágenes PAN urbanas.

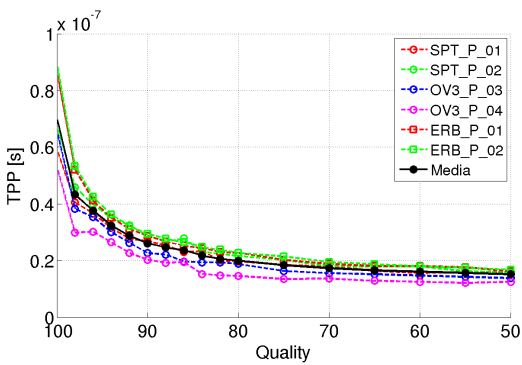


Fig. H.9. TPP vs. Quality para JPG_02.

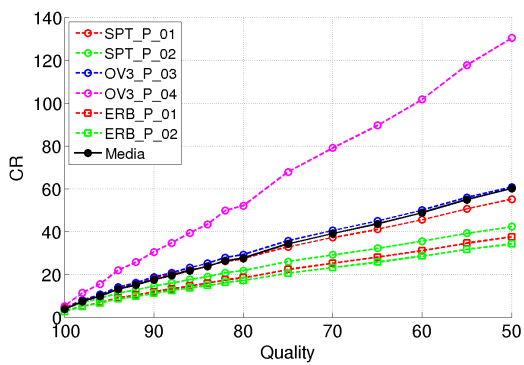


Fig. H.10. CR vs. Quality para JPG_02.

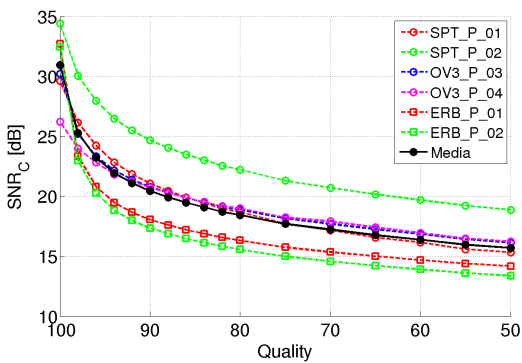


Fig. H.11. SNR_C vs. Quality para JPG_02.

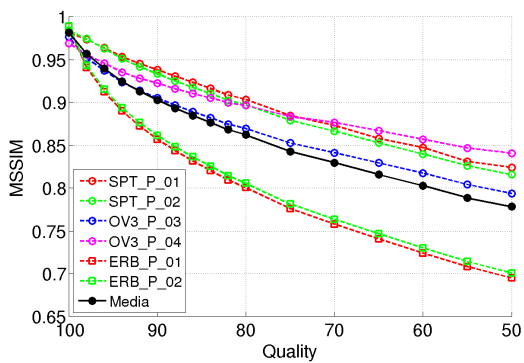


Fig. H.12. MSSIM vs. Quality para JPG_02.

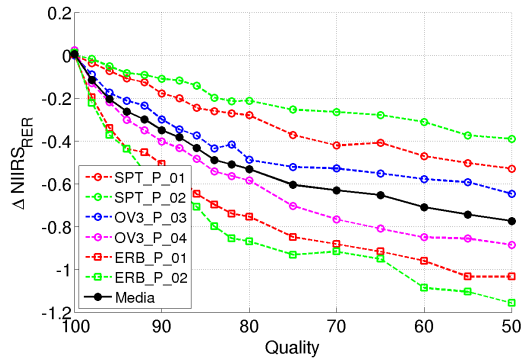


Fig. H.13. $\Delta NIIRS_{RER}$ vs. Quality para JPG_02.

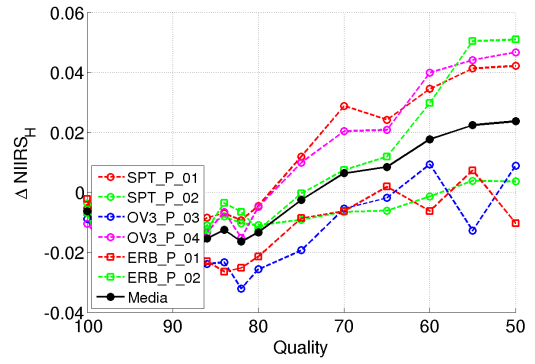


Fig. H.14. $\Delta NIIRS_H$ vs. Quality para JPG_02.

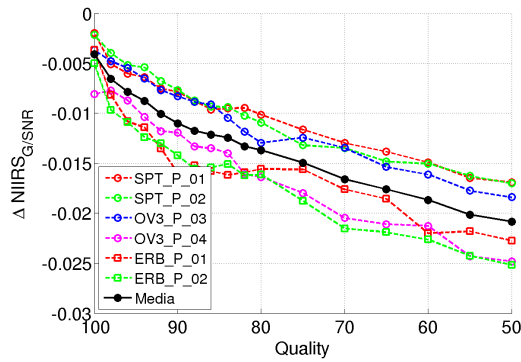


Fig. H.15. $\Delta NIIRS_{SNR}$ vs. Quality para JPG_02.

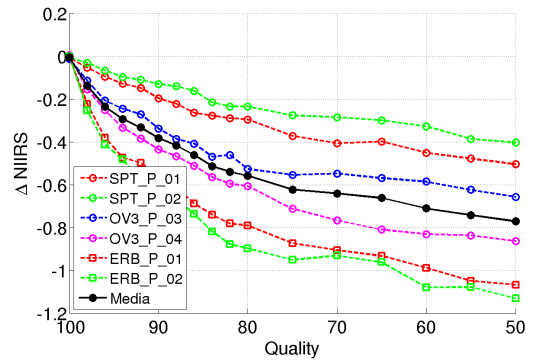


Fig. H.16. $\Delta NIIRS$ vs. Quality para JPG_02.

Caso de prueba JPG_03: Tabla R=1, imágenes PAN urbanas.

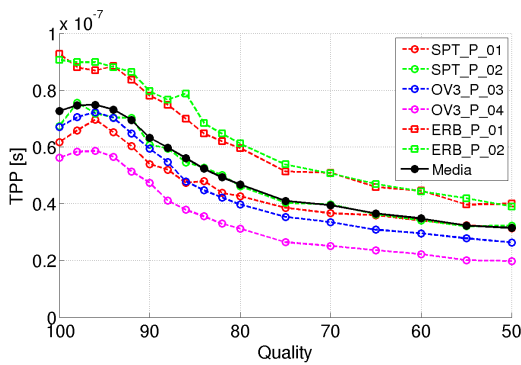


Fig. H.17. TPP vs. Quality para JPG_03.

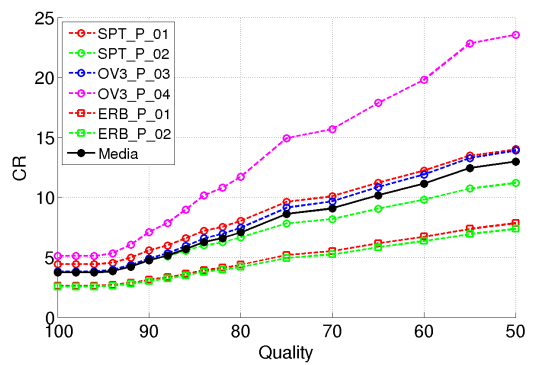


Fig. H.18. CR vs. Quality para JPG_03.

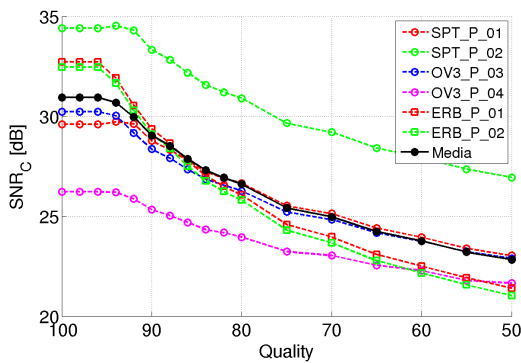


Fig. H.19. SNR_C vs. Quality para JPG_03.

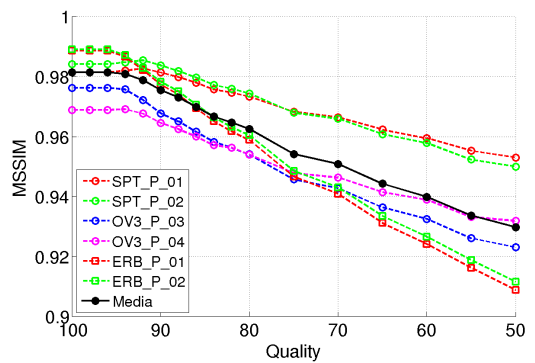


Fig. H.20. MSSIM vs. Quality para JPG_03.

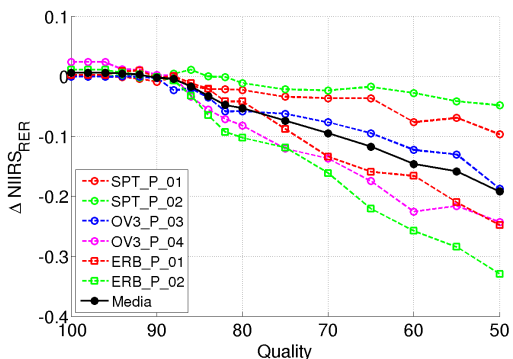


Fig. H.21. $\Delta NIIRS_{RER}$ vs. Quality para *JPG_03*.

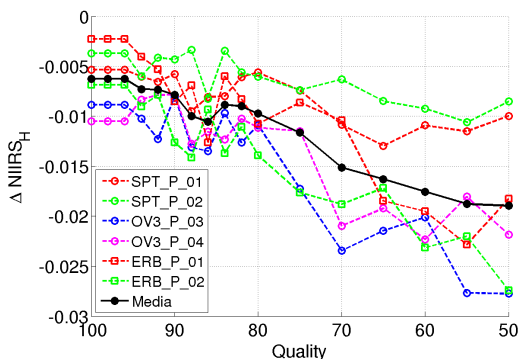


Fig. H.22. $\Delta NIIRS_H$ vs. Quality para *JPG_03*.

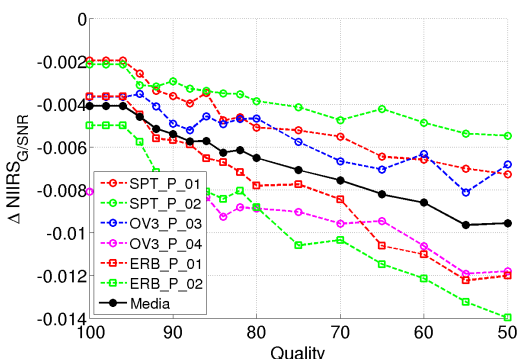


Fig. H.23. $\Delta NIIRS_{SNR}$ vs. Quality para *JPG_03*.

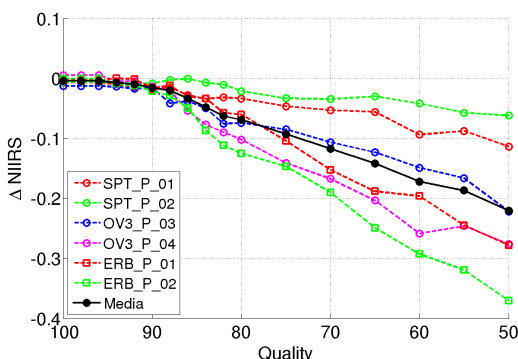


Fig. H.24. $\Delta NIIRS$ vs. Quality para *JPG_03*.

Caso de prueba *JPG_21*: Tabla *Default*, imágenes PAN rurales.

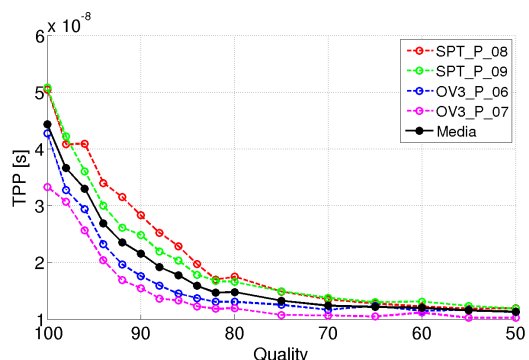


Fig. H.25. TPP vs. Quality para *JPG_21*.

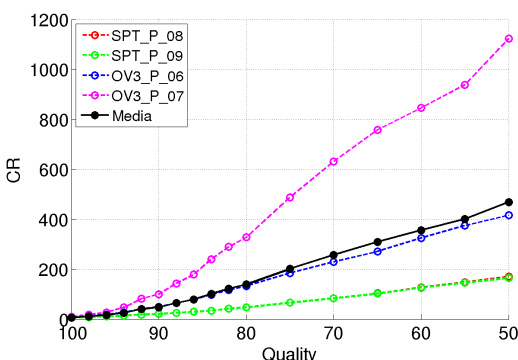


Fig. H.26. CR vs. Quality para *JPG_21*.

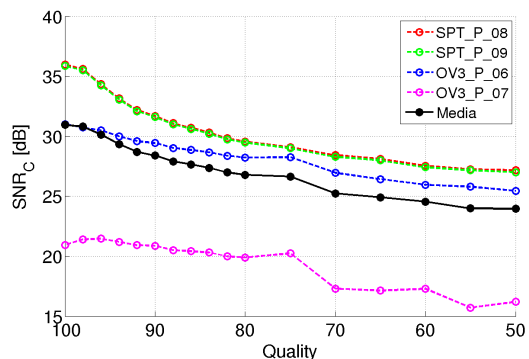


Fig. H.27. SNR_C vs. Quality para *JPG_21*.

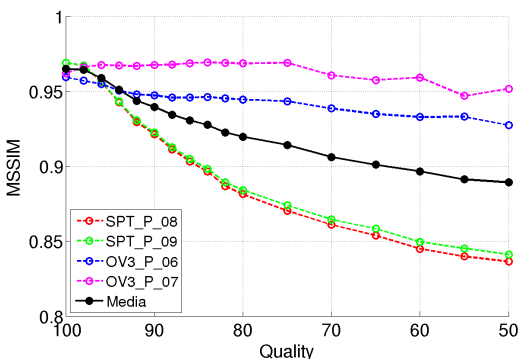


Fig. H.28. MSSIM vs. Quality para *JPG_21*.

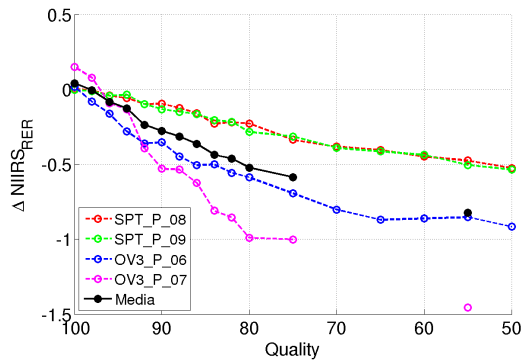


Fig. H.29. $\Delta NIIRS_{RER}$ vs. Quality para JPG_21.

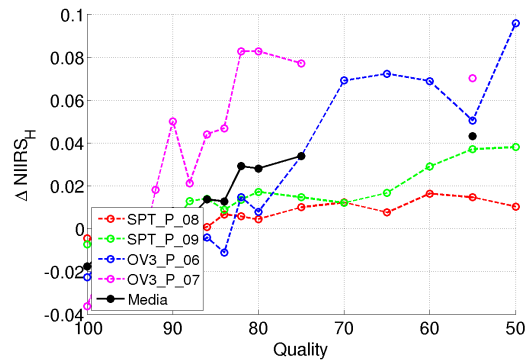


Fig. H.30. $\Delta NIIRS_H$ vs. Quality para JPG_21.

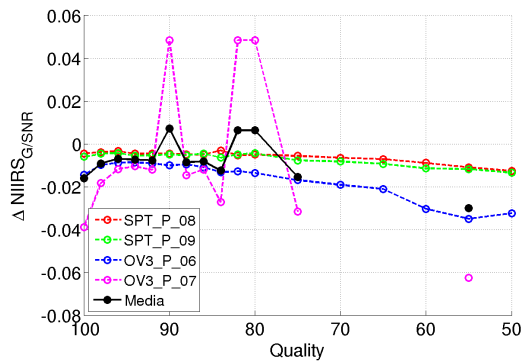


Fig. H.31. $\Delta NIIRS_{SNR}$ vs. Quality para JPG_21.

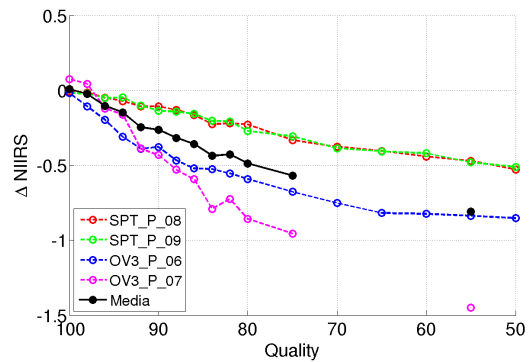


Fig. H.32. $\Delta NIIRS$ vs. Quality para JPG_21.

Caso de prueba JPG_22: Tabla GWG, imágenes PAN rurales.

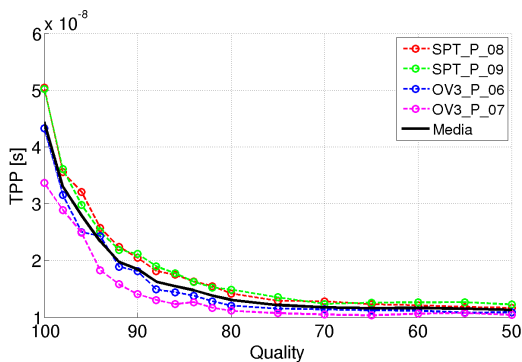


Fig. H.33. TPP vs. Quality para JPG_22.

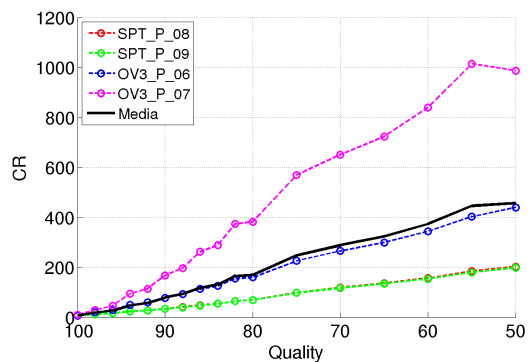


Fig. H.34. CR vs. Quality para JPG_22.

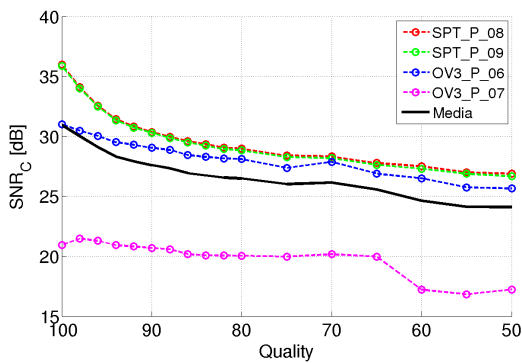


Fig. H.35. SNR_C vs. Quality para JPG_22.

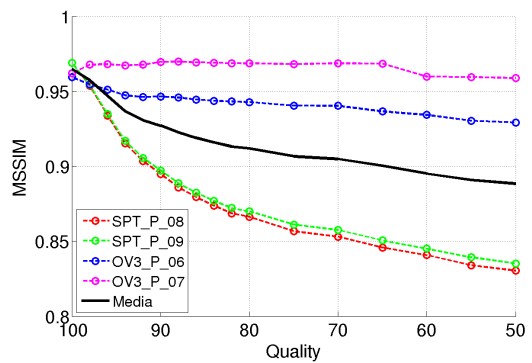


Fig. H.36. MSSIM vs. Quality para JPG_22.

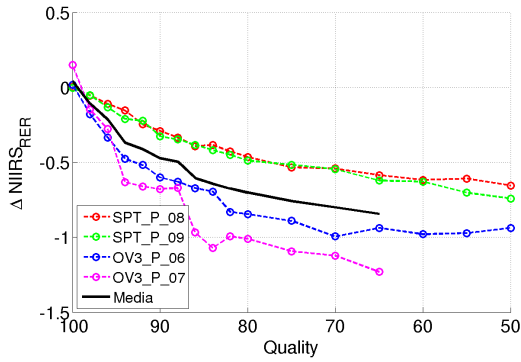


Fig. H.37. $\Delta NIIRS_{RER}$ vs. Quality para JPG_22.

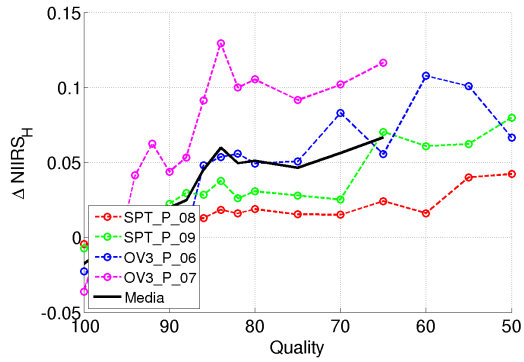


Fig. H.38. $\Delta NIIRS_H$ vs. Quality para JPG_22.

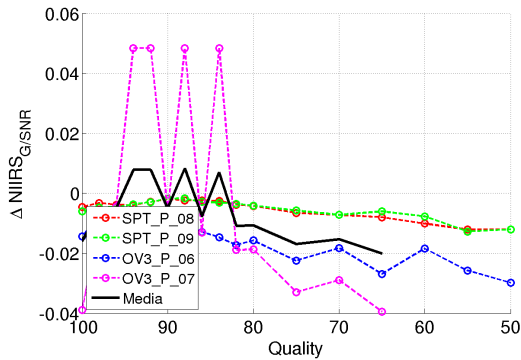


Fig. H.39. $\Delta NIIRS_{SNR}$ vs. Quality para JPG_22.

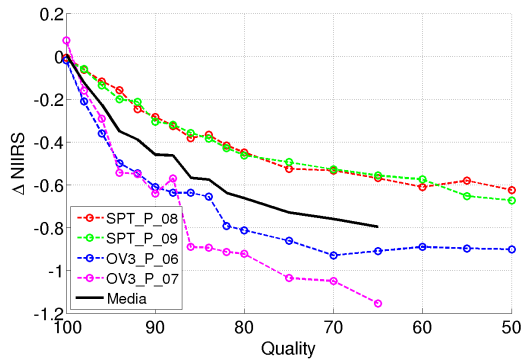


Fig. H.40. $\Delta NIIRS$ vs. Quality para JPG_22.

Caso de prueba JPG_23: Tabla $R=1$, imágenes PAN rurales.

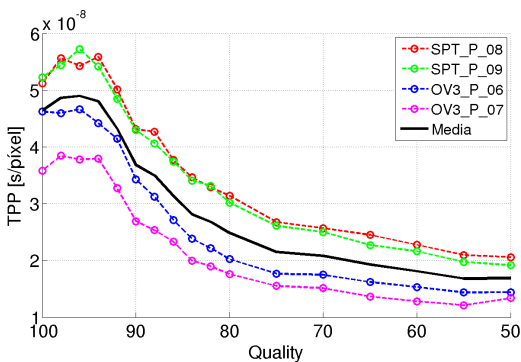


Fig. H.41. TPP vs. Quality para JPG_23.

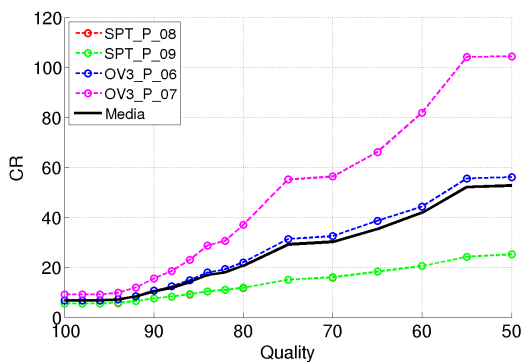


Fig. H.42. CR vs. Quality para JPG_23.

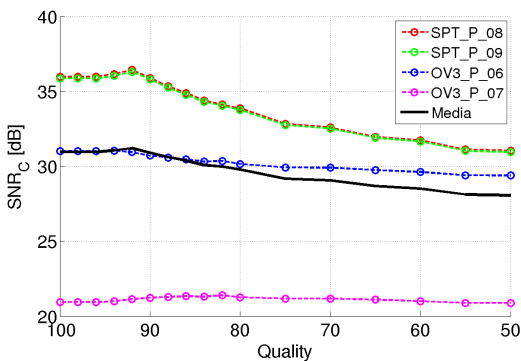


Fig. H.43. SNR_C vs. Quality para JPG_23.

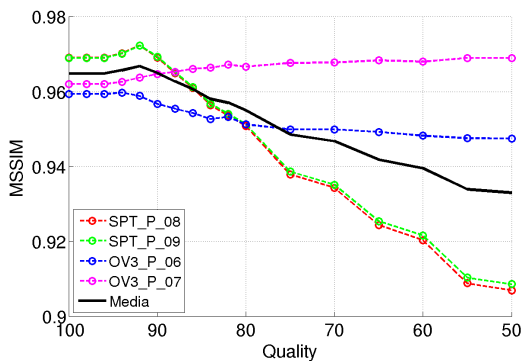


Fig. H.44. MSSIM vs. Quality para JPG_23.

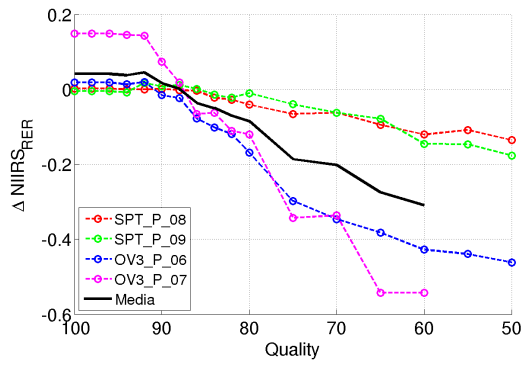


Fig. H.45. $\Delta NIIRS_{RER}$ vs. Quality para JPG_23.

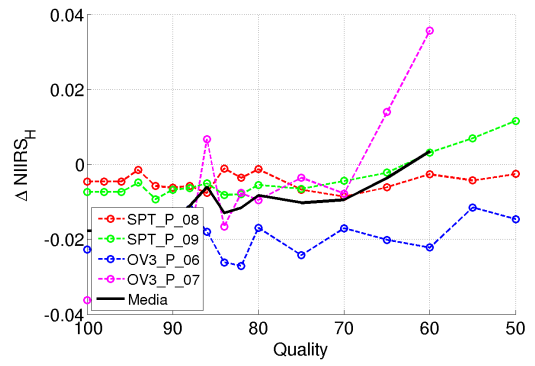


Fig. H.46. $\Delta NIIRS_H$ vs. Quality para JPG_23.

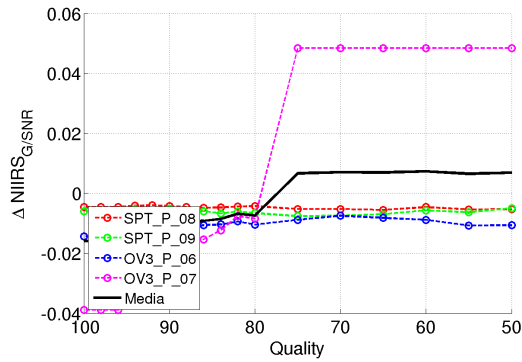


Fig. H.47. $\Delta NIIRS_{SNR}$ vs. Quality para JPG_23.

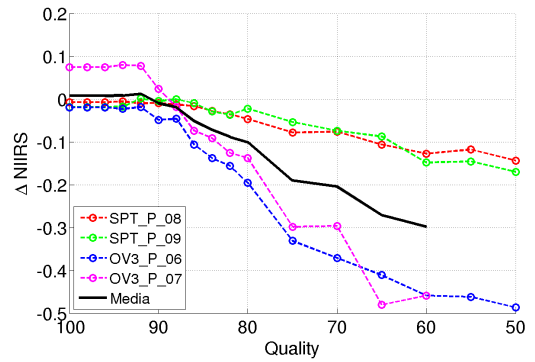


Fig. H.48. $\Delta NIIRS$ vs. Quality para JPG_23.

Caso de prueba JPG_31: Tabla R=1, imágenes MS urbanas.

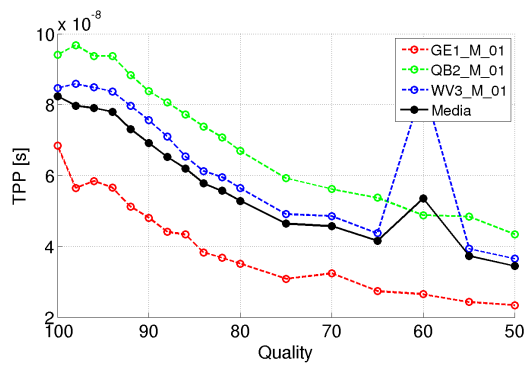


Fig. H.49. TPP vs. Quality para JPG_31.

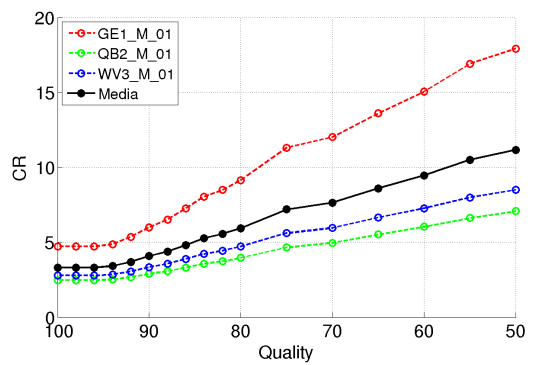


Fig. H.50. CR vs. Quality para JPG_31.

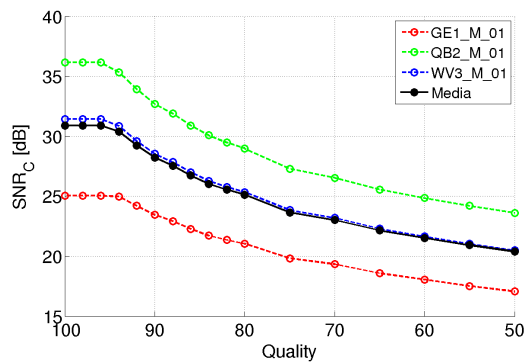


Fig. H.51. SNR_C vs. Quality para JPG_31.

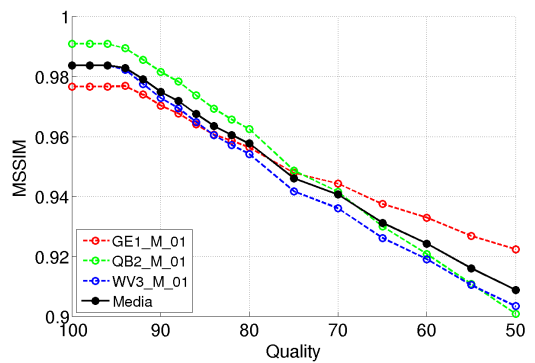


Fig. H.52. MSSIM vs. Quality para JPG_31.

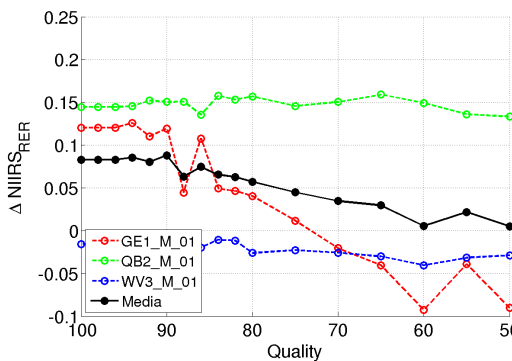


Fig. H.53. $\Delta NIIRS_{RER}$ vs. Quality para *JPG_31*.

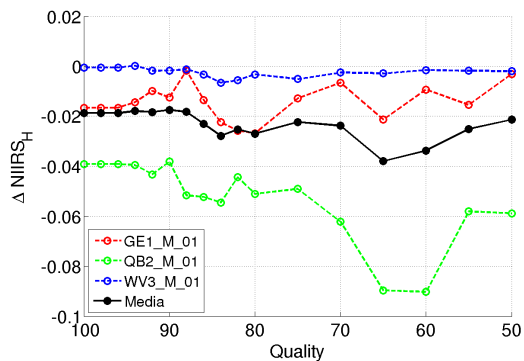


Fig. H.54. $\Delta NIIRS_H$ vs. Quality para *JPG_31*.

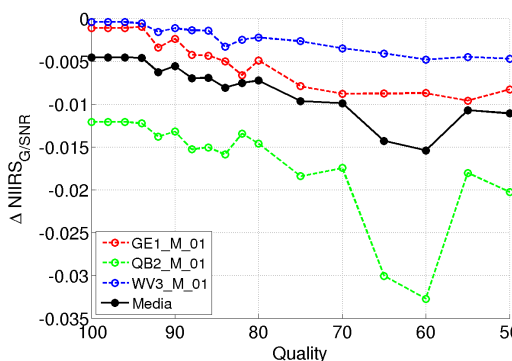


Fig. H.55. $\Delta NIIRS_{SNR}$ vs. Quality para *JPG_31*.

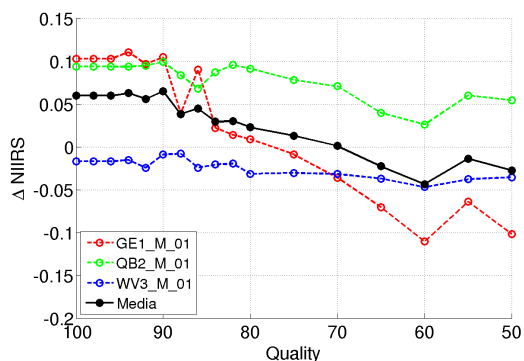


Fig. H.56. $\Delta NIIRS$ vs. Quality para *JPG_31*.

Caso de prueba *JPG_41*: Tabla $R=1$, imágenes MS rurales.

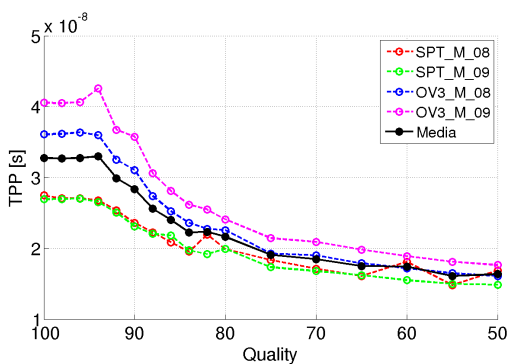


Fig. H.57. TPP vs. Quality para *JPG_41*.

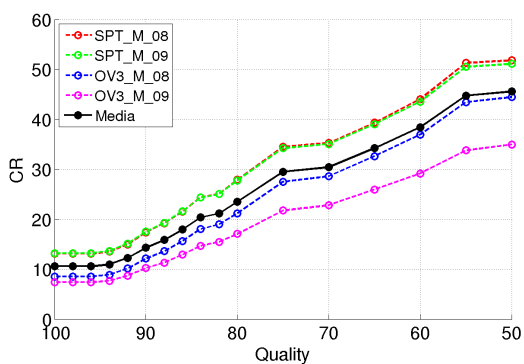


Fig. H.58. CR vs. Quality para *JPG_41*.

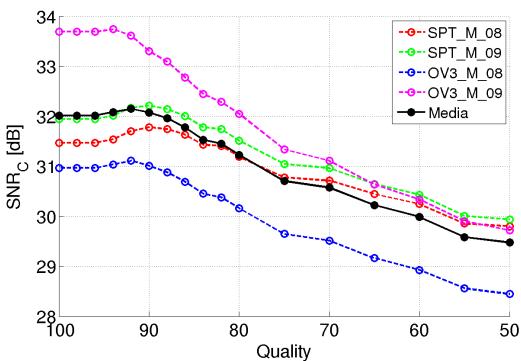


Fig. H.59. SNR_C vs. Quality para *JPG_41*.

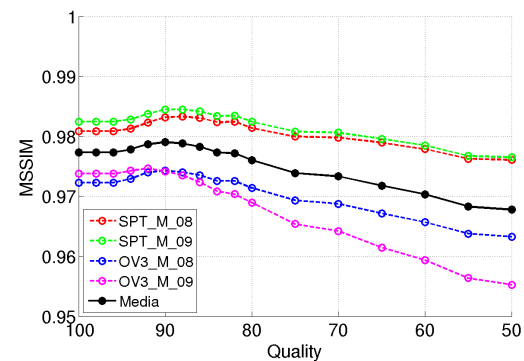


Fig. H.60. MSSIM vs. Quality para *JPG_41*.

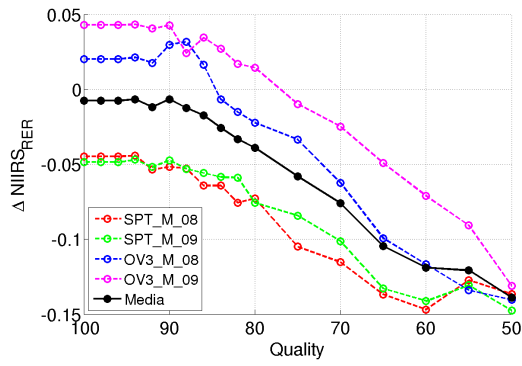


Fig. H.61. $\Delta NIIRS_{RER}$ vs. Quality para JPG-41.

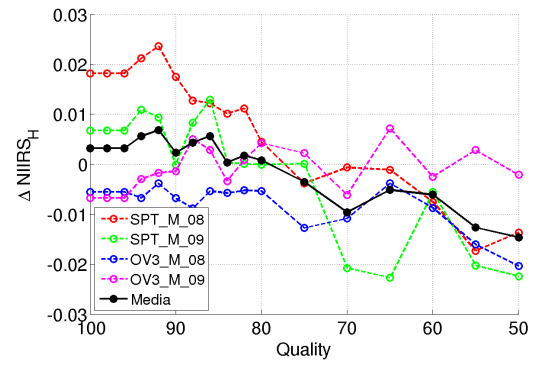


Fig. H.62. $\Delta NIIRS_H$ vs. Quality para JPG-41.

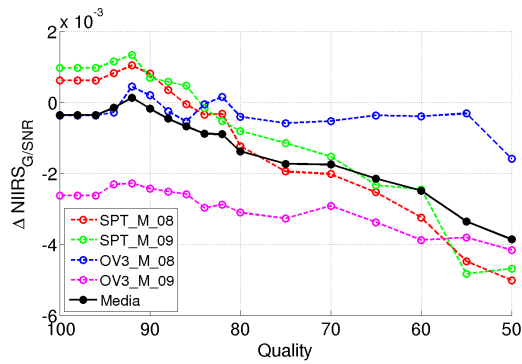


Fig. H.63. $\Delta NIIRS_{GSNR}$ vs. Quality para JPG-41.

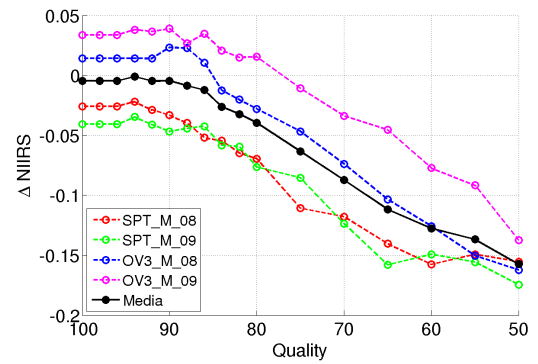


Fig. H.64. $\Delta NIIRS$ vs. Quality para JPG-41.

Resultados de evaluación de JPEG2000

A continuación se presentan todos los resultados de los casos de prueba de JPEG2000 mostrados en la Tabla 5.11. Aunque la información de cada caso está en la misma aquí se indican algunos parámetros de configuración y el conjunto de datos de prueba de cada caso.

Caso de prueba JP2.01: DWT entera, $N_{DWT}=5$, $M_T=M$, imágenes PAN urbanas.

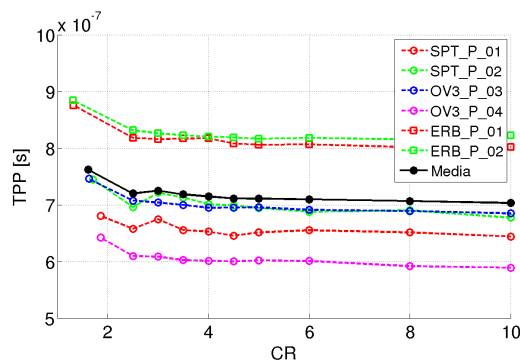


Fig. I.1. TPP vs. CR para JP2.01.

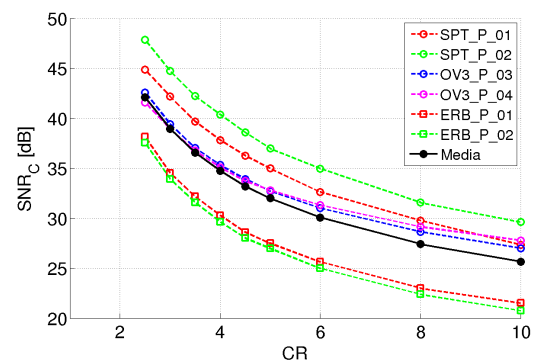


Fig. I.2. SNR_C vs. CR para JP2.01.

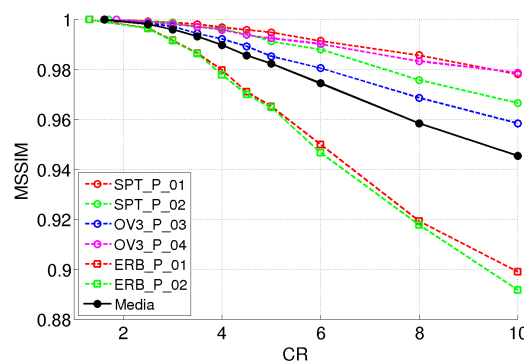


Fig. I.3. MSSIM vs. CR para JP2.01.

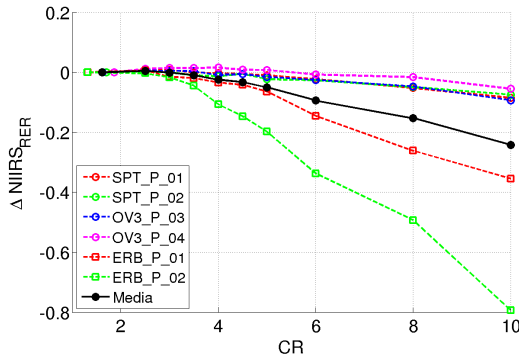


Fig. I.4. $\Delta NIIRS_{RER}$ vs. CR para JP2_01.

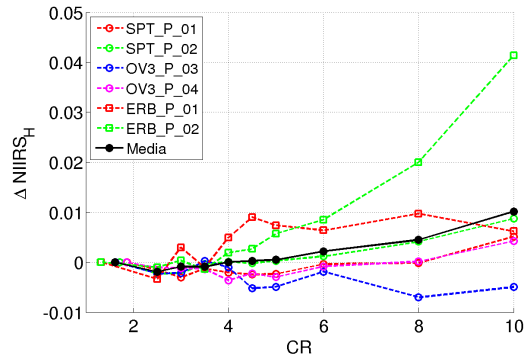


Fig. I.5. $\Delta NIIRS_H$ vs. CR para JP2_01.

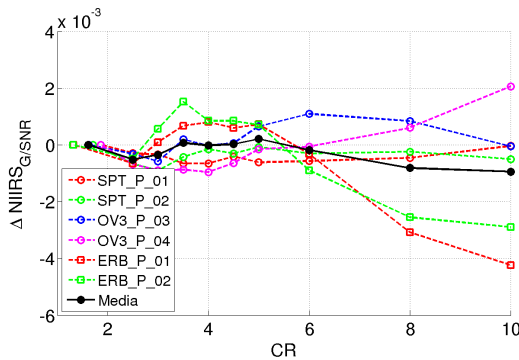


Fig. I.6. $\Delta NIIRS_{SNR}$ vs. CR para JP2_01.

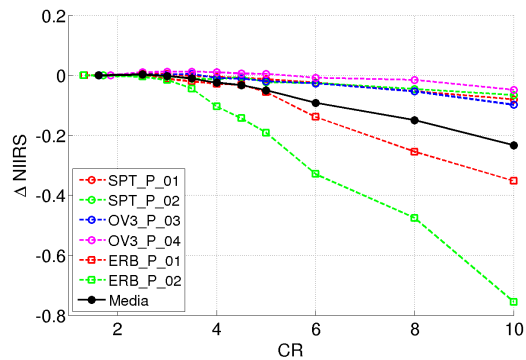


Fig. I.7. $\Delta NIIRS$ vs. CR para JP2_01.

Caso de prueba JP2_02: DWT flotante, $N_{DWT}=5$, $M_T=M$, imágenes PAN urbanas.

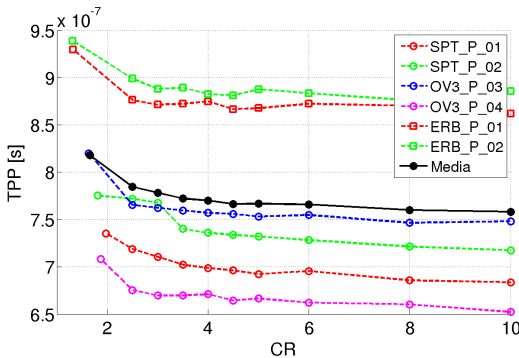


Fig. I.8. TPP vs. CR para JP2_02.

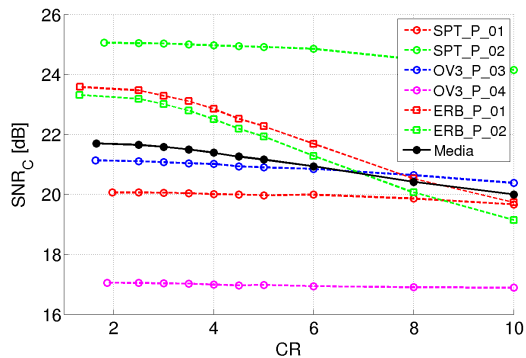


Fig. I.9. SNR_C vs. CR para JP2_02.

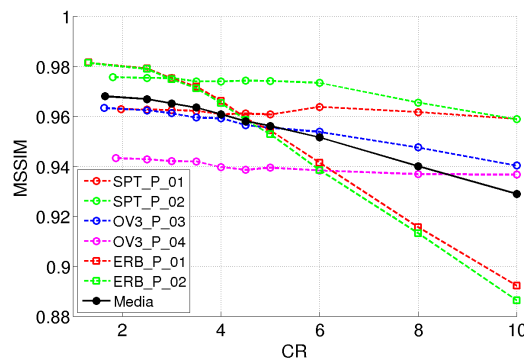


Fig. I.10. MSSIM vs. CR para JP2_02.

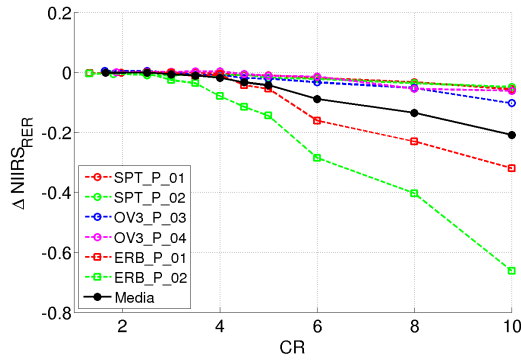


Fig. I.11. $\Delta NIIRS_{RER}$ vs. CR para $JP2_{02}$.

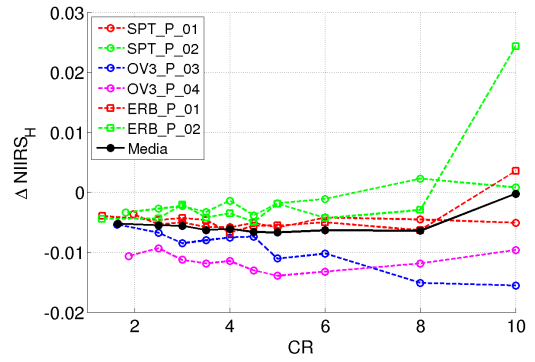


Fig. I.12. $\Delta NIIRS_H$ vs. CR para $JP2_{02}$.

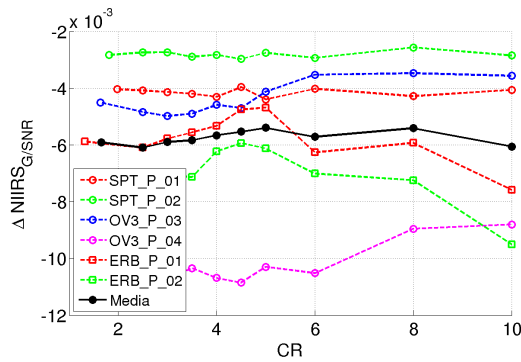


Fig. I.13. $\Delta NIIRS_{SNR}$ vs. CR para $JP2_{02}$.

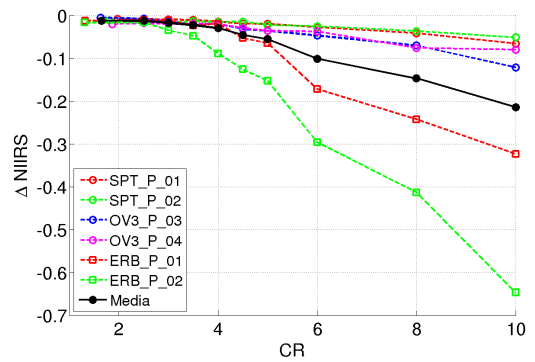


Fig. I.14. $\Delta NIIRS$ vs. CR para $JP2_{02}$.

Caso de prueba JP2_03: Barrido para cantidad de DWT 2D (N_{DWT}), DWT entera, $M_T=M$, $CR=0$, imágenes PAN urbanas.

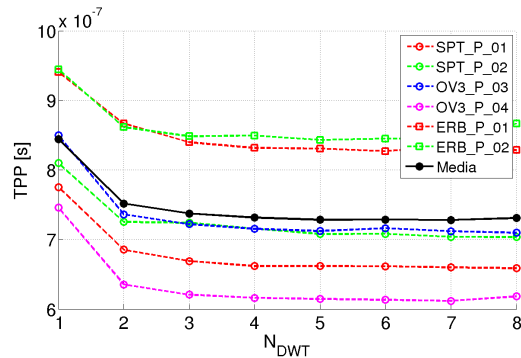


Fig. I.15. TPP vs. N_{DWT} para $JP2_{03}$.

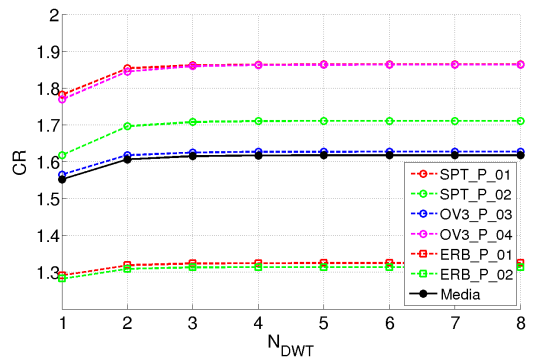


Fig. I.16. CR vs. N_{DWT} para $JP2_{03}$.

Caso de prueba JP2_04: Barrido para cantidad de DWT 2D (N_{DWT}), DWT entera, $M_T=M$, $CR=10$, imágenes PAN urbanas.

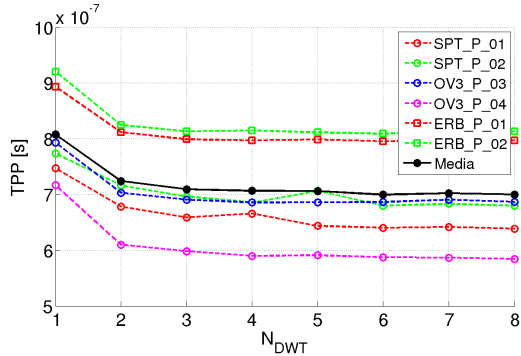


Fig. I.17. TPP vs. N_{DWT} para JP2_04.

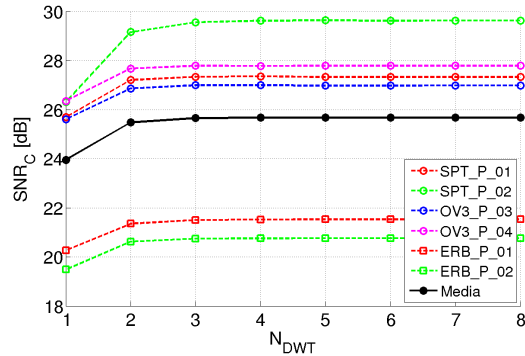


Fig. I.18. SNR_C vs. N_{DWT} para JP2_04.

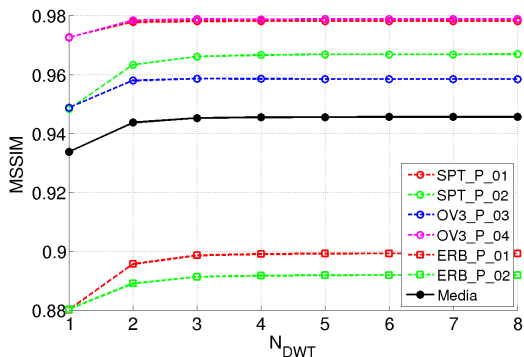


Fig. I.19. MSSIM vs. N_{DWT} para JP2_04.

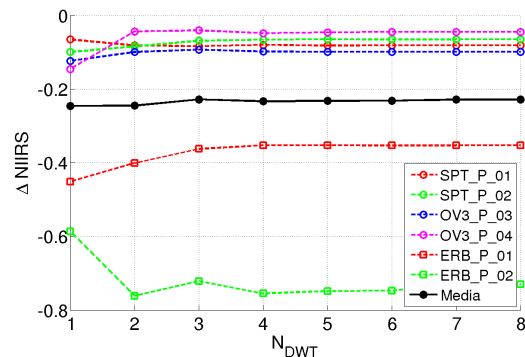


Fig. I.20. $\Delta NIIRS$ vs. N_{DWT} para JP2_04.

Caso de prueba JP2_05: Barrido para cantidad filas por tesela (M_T), DWT entera, $N_{DWT}=4$, $CR=0$, imágenes PAN urbanas.

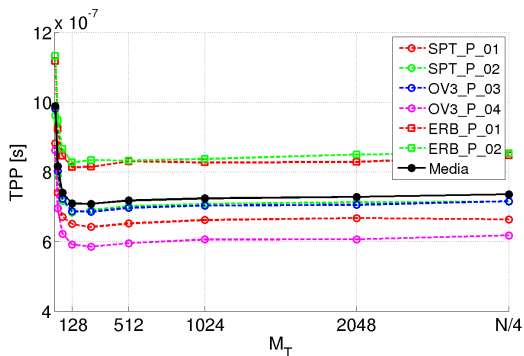


Fig. I.21. TPP vs. M_T para JP2_05.

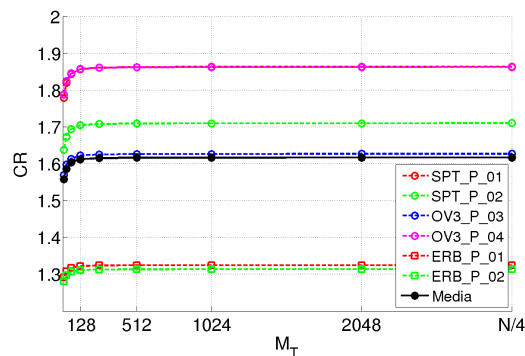


Fig. I.22. CR vs. M_T para JP2_05.

Caso de prueba JP2_06: Barrido para cantidad filas por tesela (M_T), DWT entera, $N_{DWT}=4$, $CR=10$, imágenes PAN urbanas.

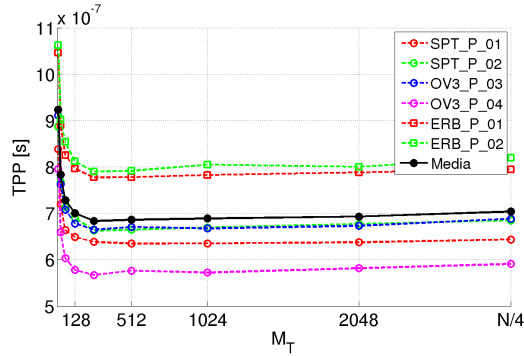


Fig. I.23. TPP vs. M_T para JP2_06.

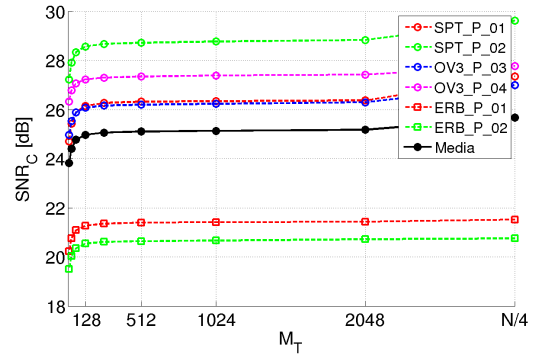


Fig. I.24. SNR_C vs. M_T para JP2_06.

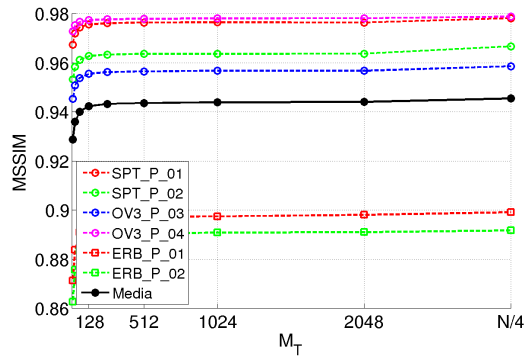


Fig. I.25. $MSSIM$ vs. M_T para JP2_06.

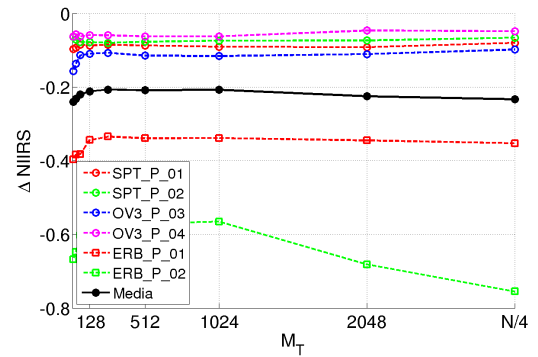


Fig. I.26. $\Delta NIIRS$ vs. M_T para JP2_06.

Caso de prueba JP2_07: DWT entera, $N_{DWT}=4$, $M_T=32$, imágenes PAN urbanas.

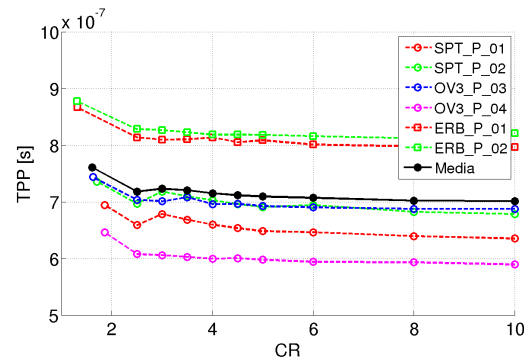


Fig. I.27. TPP vs. CR para JP2_07.

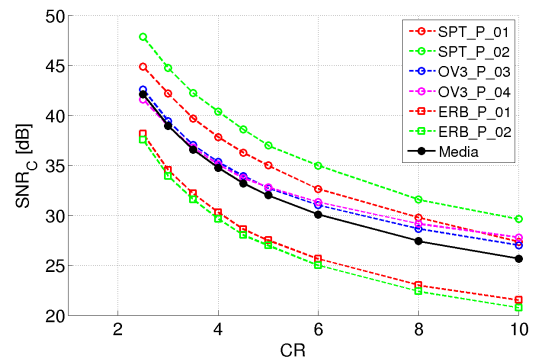


Fig. I.28. SNR_C vs. CR para JP2_07.

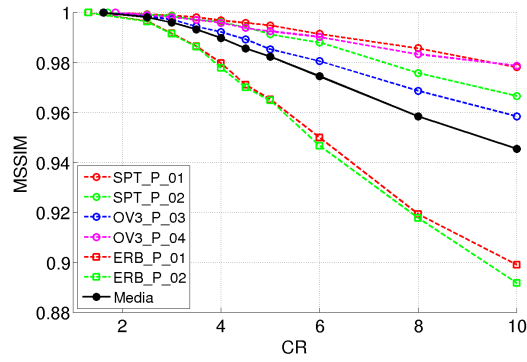


Fig. I.29. $MSSIM$ vs. CR para $JP2_07$.

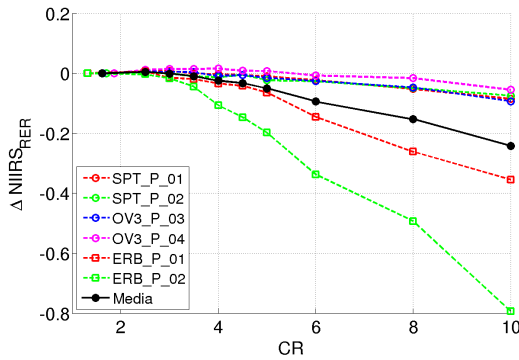


Fig. I.30. $\Delta NIIRS_{RER}$ vs. CR para $JP2_07$.

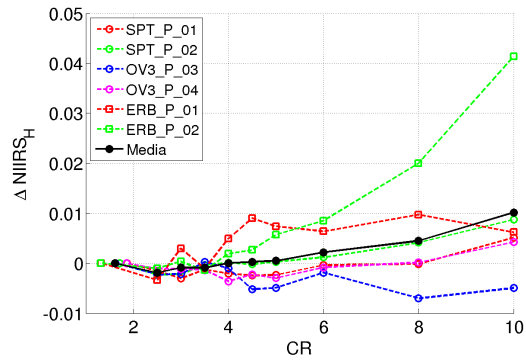


Fig. I.31. $\Delta NIIRS_H$ vs. CR para $JP2_07$.

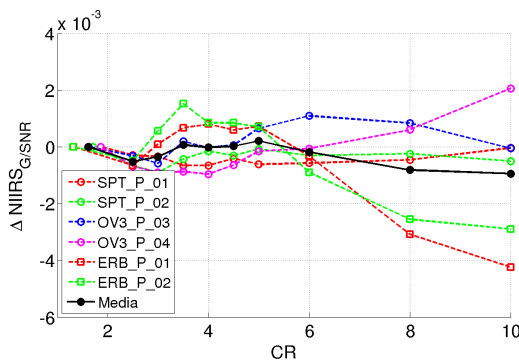


Fig. I.32. $\Delta NIIRS_{SNR}$ vs. CR para $JP2_07$.

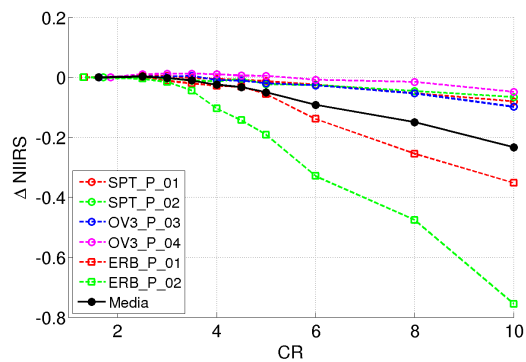


Fig. I.33. $\Delta NIIRS$ vs. CR para $JP2_07$.

Caso de prueba $JP2_08$: DWT entera, $N_{DWT}=4$, $M_T=512$, imágenes PAN urbanas.

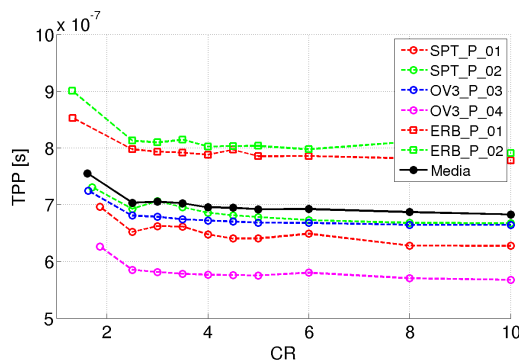


Fig. I.34. TPP vs. CR para $JP2_08$.

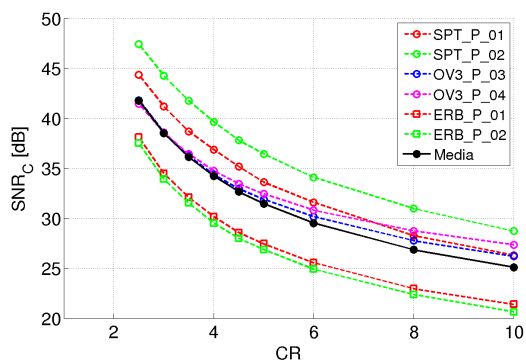


Fig. I.35. SNR_C vs. CR para $JP2_08$.

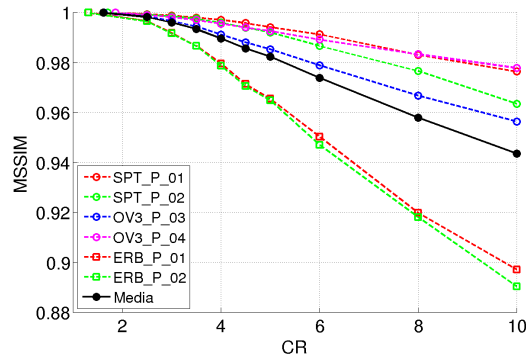


Fig. I.36. MSSIM vs. CR para JP2_08.

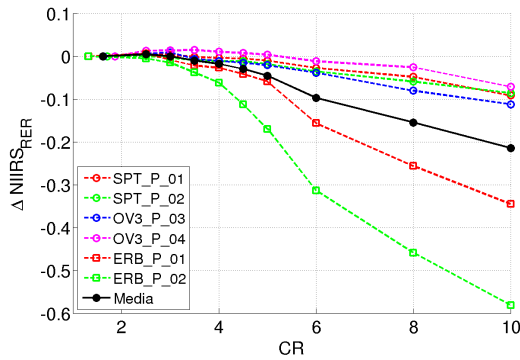


Fig. I.37. $\Delta NIIRS_{RER}$ vs. CR para JP2_08.

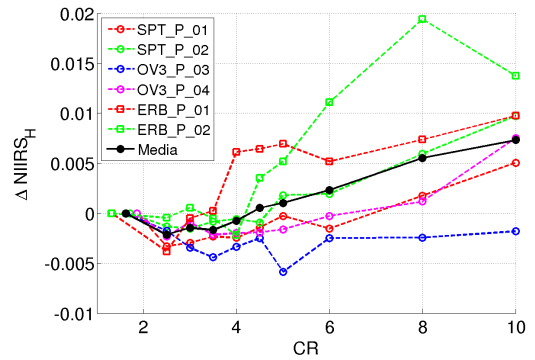


Fig. I.38. $\Delta NIIRS_H$ vs. CR para JP2_08.

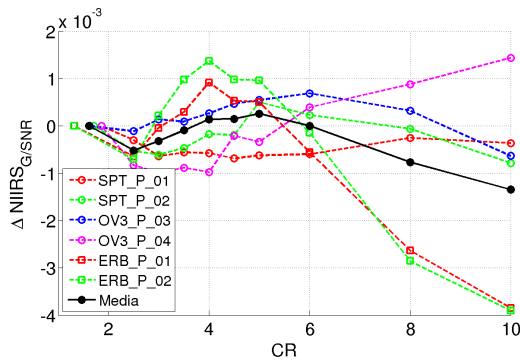


Fig. I.39. $\Delta NIIRS_{SNR}$ vs. CR para JP2_08.

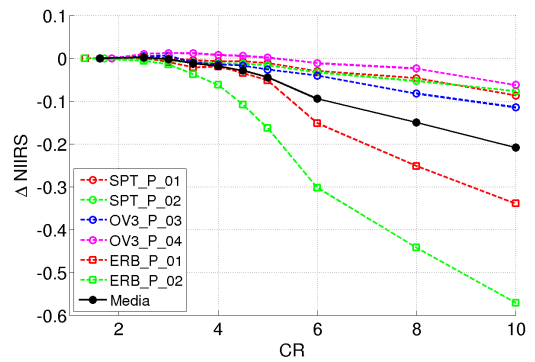


Fig. I.40. $\Delta NIIRS$ vs. CR para JP2_08.

Caso de prueba JP2_09: DWT entera, $N_{DWT}=4$, $M_T=M$, imágenes PAN urbanas.

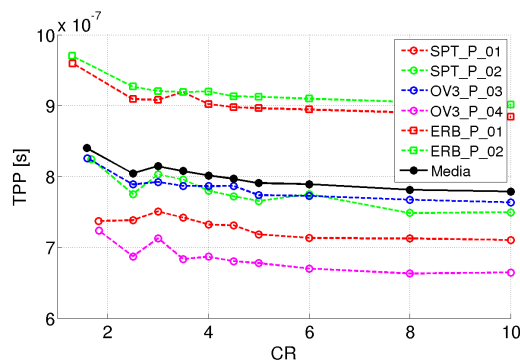


Fig. I.41. TPP vs. CR para JP2_09.

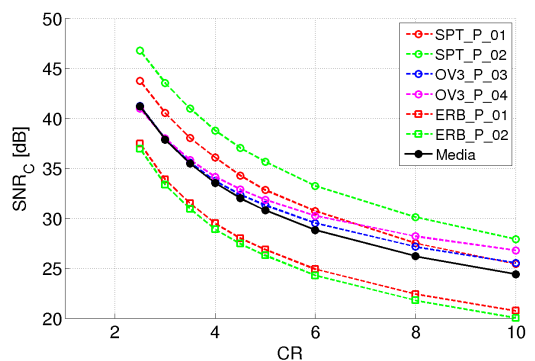


Fig. I.42. SNR_C vs. CR para JP2_09.

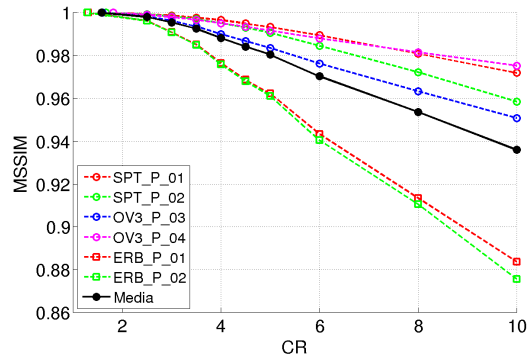


Fig. I.43. $MSSIM$ vs. CR para $JP2_{09}$.

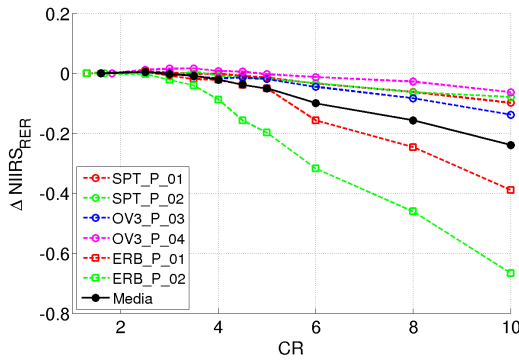


Fig. I.44. $\Delta NIIRS_{RER}$ vs. CR para $JP2_{09}$.

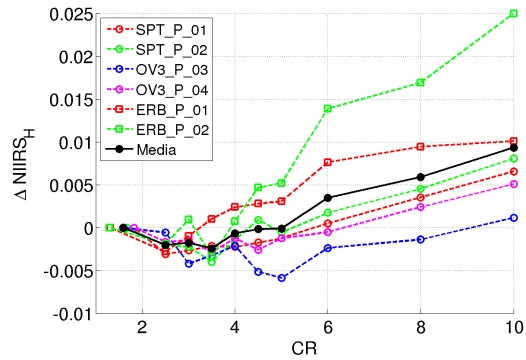


Fig. I.45. $\Delta NIIRS_H$ vs. CR para $JP2_{09}$.

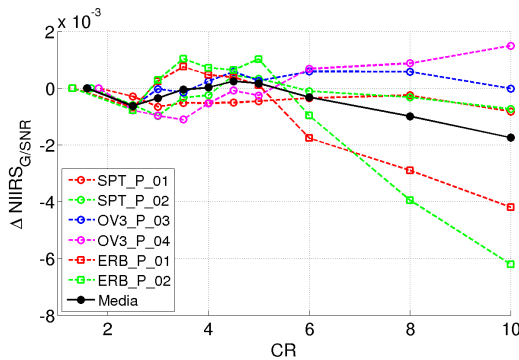


Fig. I.46. $\Delta NIIRS_{SNR}$ vs. CR para $JP2_{09}$.

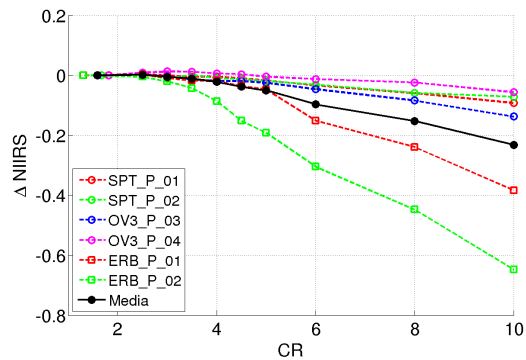


Fig. I.47. $\Delta NIIRS$ vs. CR para $JP2_{09}$.

Caso de prueba $JP2_{21}$: DWT entera, $N_{DWT}=4$, $M_T=512$, imágenes PAN rurales.

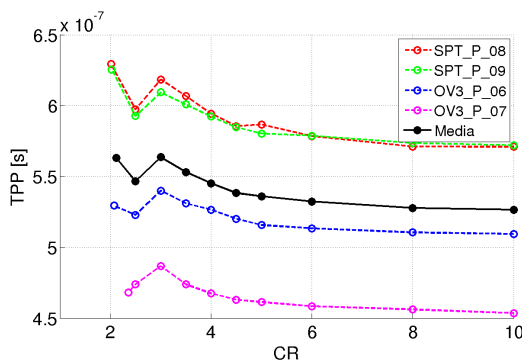


Fig. I.48. TPP vs. CR para $JP2_{21}$.

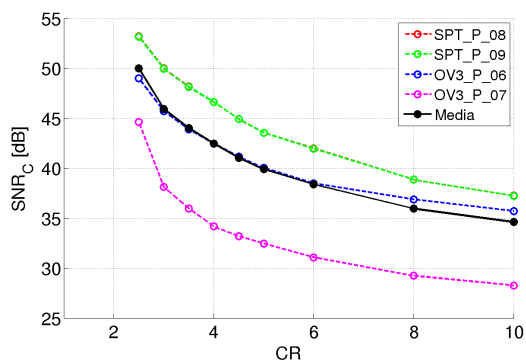


Fig. I.49. SNR_C vs. CR para $JP2_{21}$.

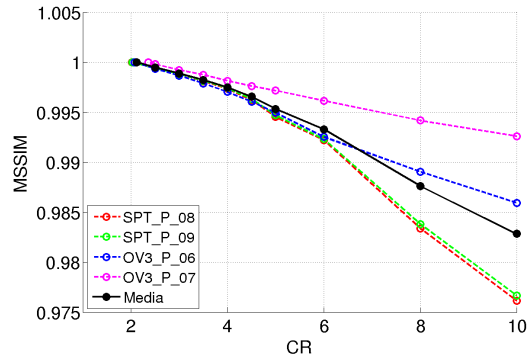


Fig. I.50. $MSSIM$ vs. CR para $JP2_{21}$.

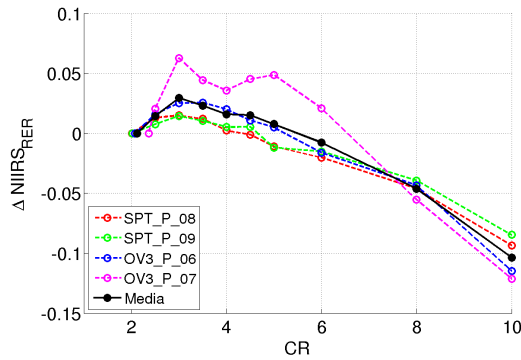


Fig. I.51. $\Delta NIIRS_{RER}$ vs. CR para $JP2_{21}$.

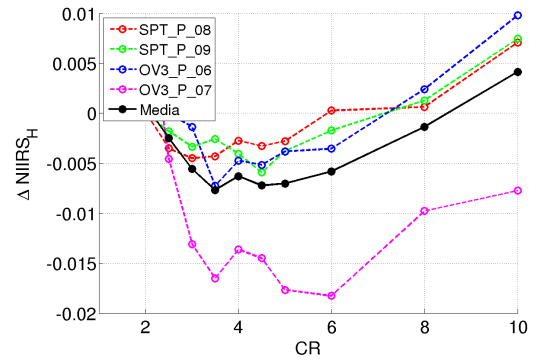


Fig. I.52. $\Delta NIIRS_H$ vs. CR para $JP2_{21}$.

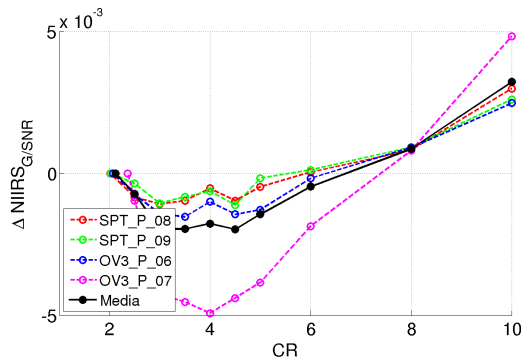


Fig. I.53. $\Delta NIIRS_{SNR}$ vs. CR para $JP2_{21}$.

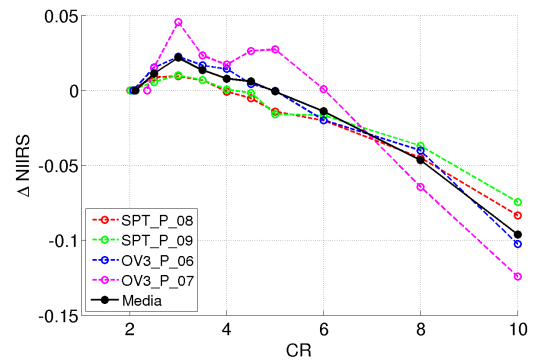


Fig. I.54. $\Delta NIIRS$ vs. CR para $JP2_{21}$.

Caso de prueba JP2_23: Barrido para cantidad de DWT 2D (N_{DWT}), DWT entera, $M_T=M$, $CR=0$, imágenes PAN rurales.

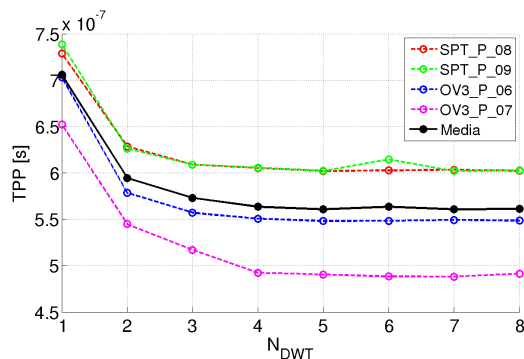


Fig. I.55. TPP vs. N_{DWT} para $JP2_{23}$.

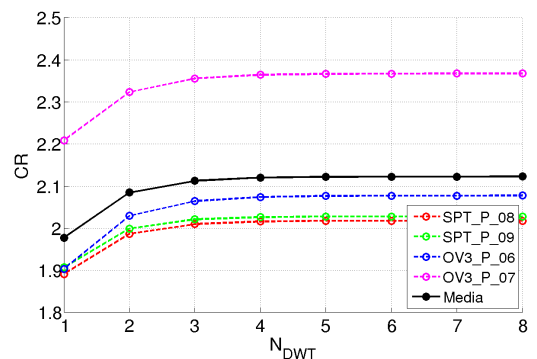


Fig. I.56. CR vs. N_{DWT} para $JP2_{23}$.

Caso de prueba JP2_24: Barrido para cantidad de DWT 2D (N_{DWT}), DWT entera, $M_T=M$, $CR=10$, imágenes PAN rurales.

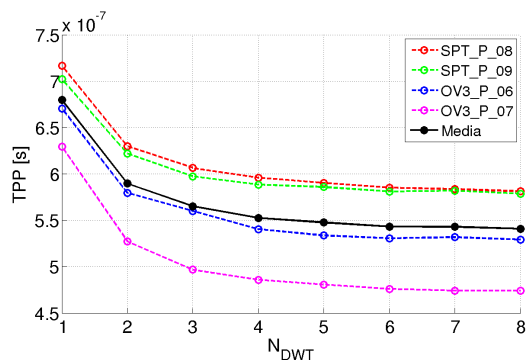


Fig. I.57. TPP vs. N_{DWT} para JP2_24.

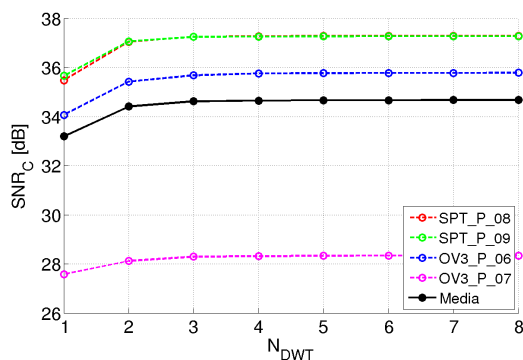


Fig. I.58. SNR_C vs. N_{DWT} para JP2_24.

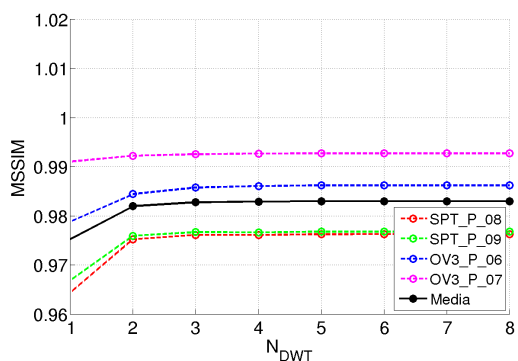


Fig. I.59. $MSSIM$ vs. N_{DWT} para JP2_24.

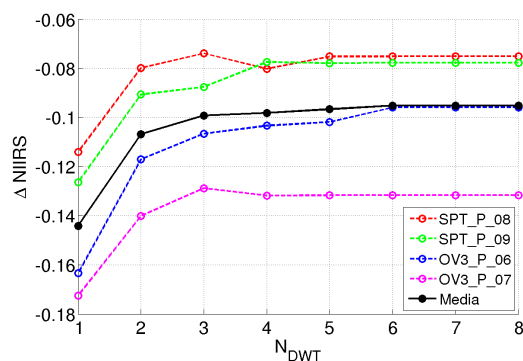


Fig. I.60. $\Delta NIIRS$ vs. N_{DWT} para JP2_24.

Caso de prueba JP2_25: Barrido para cantidad filas por tesela (M_T), DWT entera, $N_{DWT}=4$, $CR=0$, imágenes PAN rurales.

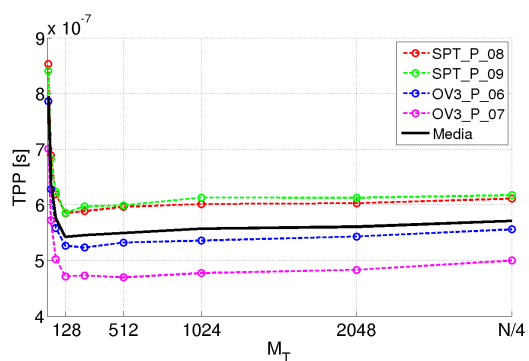


Fig. I.61. TPP vs. M_T para JP2_25.

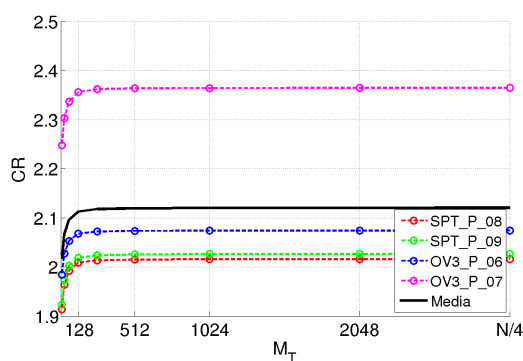


Fig. I.62. CR vs. M_T para JP2_25.

Caso de prueba JP2_26: Barrido para cantidad filas por tesela (M_T), DWT entera, $N_{DWT}=4$, $CR=10$, imágenes PAN rurales.

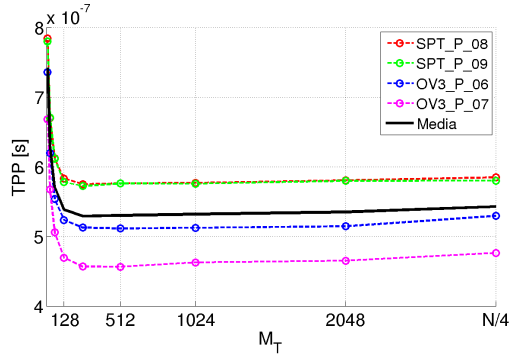


Fig. I.63. TPP vs. M_T para JP2_26.

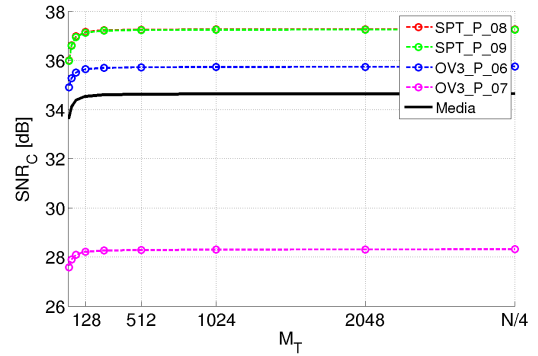


Fig. I.64. SNR_C vs. M_T para JP2_26.

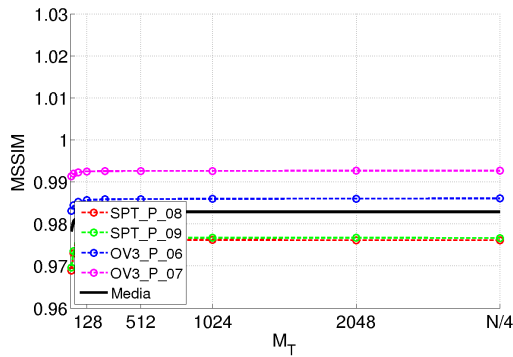


Fig. I.65. $MSSIM$ vs. M_T para JP2_26.

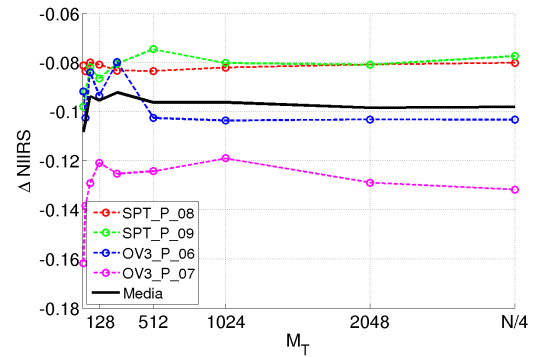


Fig. I.66. $\Delta NIIRS$ vs. M_T para JP2_26.

Caso de prueba JP2_31: DWT entera, $N_{DWT}=4$, $M_T=512$, imágenes MS urbanas.

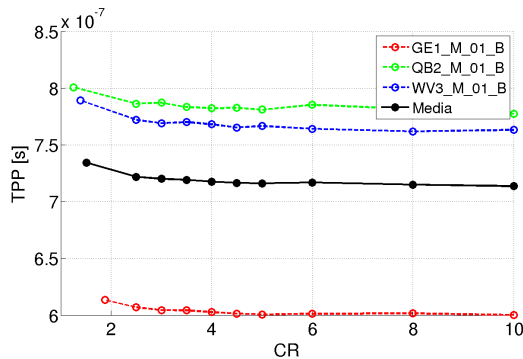


Fig. I.67. TPP vs. CR para JP2_31.

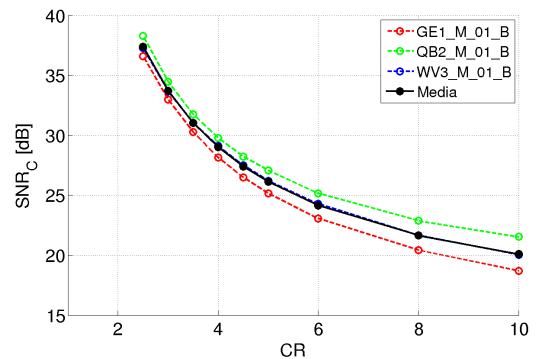


Fig. I.68. SNR_C vs. CR para JP2_31.

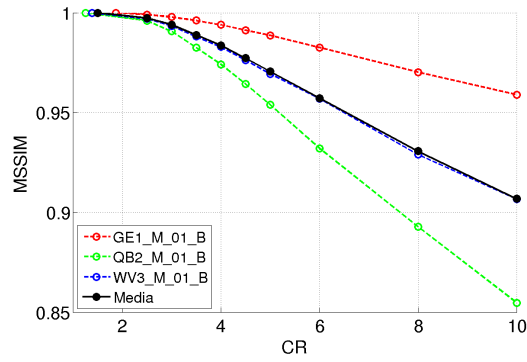


Fig. I.69. $MSSIM$ vs. CR para $JP2_{31}$.

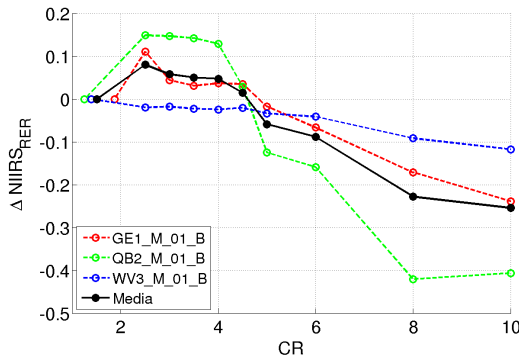


Fig. I.70. $\Delta NIIRS_{RER}$ vs. CR para $JP2_{31}$.

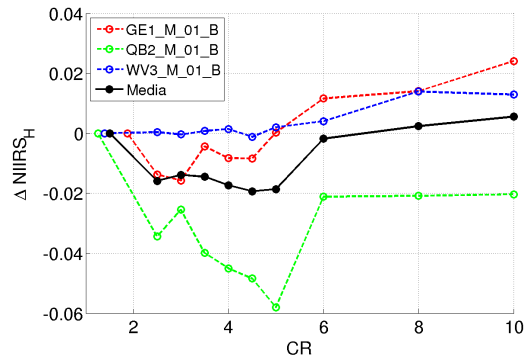


Fig. I.71. $\Delta NIIRS_H$ vs. CR para $JP2_{31}$.

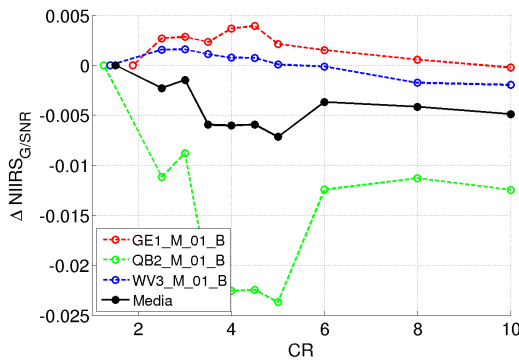


Fig. I.72. $\Delta NIIRS_{SNR}$ vs. CR para $JP2_{31}$.

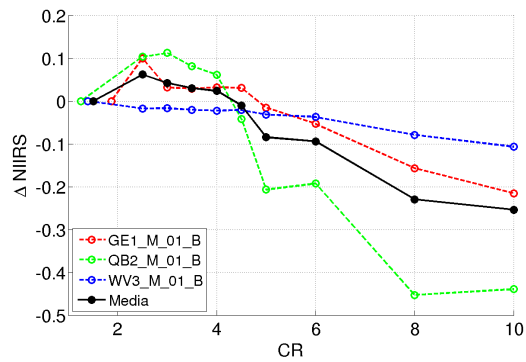


Fig. I.73. $\Delta NIIRS$ vs. CR para $JP2_{31}$.

Caso de prueba $JP2_{41}$: DWT entera, $N_{DWT}=4$, $M_T=512$, imágenes MS rurales.

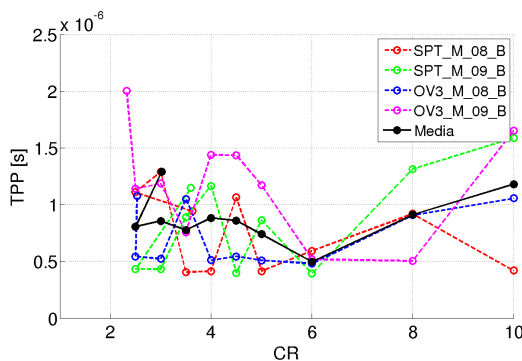


Fig. I.74. TPP vs. CR para $JP2_{41}$.

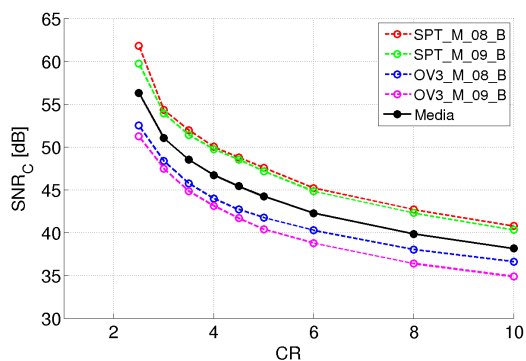


Fig. I.75. SNR_C vs. CR para $JP2_{41}$.

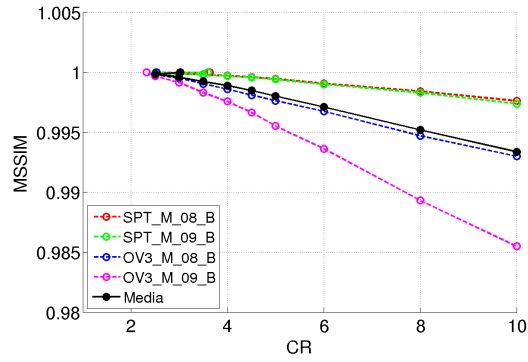


Fig. I.76. $MSSIM$ vs. CR para $JP2_{41}$.

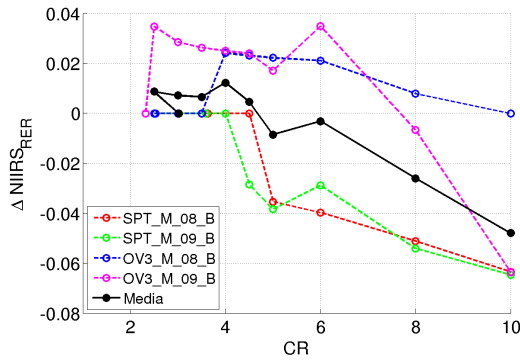


Fig. I.77. $\Delta NIIRS_{RER}$ vs. CR para $JP2_{41}$.

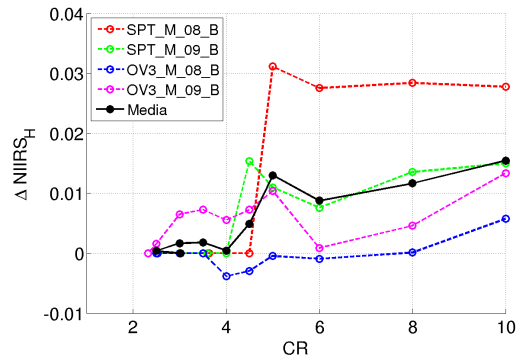


Fig. I.78. $\Delta NIIRS_H$ vs. CR para $JP2_{41}$.

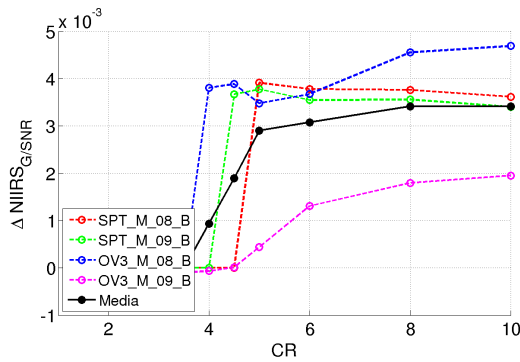


Fig. I.79. $\Delta NIIRS_{GSNR}$ vs. CR para $JP2_{41}$.

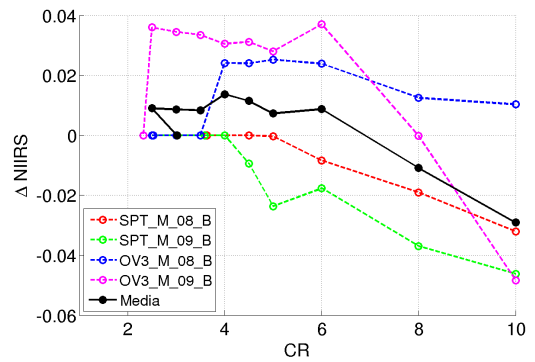


Fig. I.80. $\Delta NIIRS$ vs. CR para $JP2_{41}$.

Resultados de evaluación de CCSDS-IDC

A continuación se presentan todos los resultados de los casos de prueba de CCSDS-IDC mostrados en la Tabla 5.12. Aunque la información de cada caso está en la misma aquí se indican algunos parámetros de configuración y el conjunto de datos de prueba de cada caso.

Caso de prueba IDC_01: DWT entera, $M_{si}=M/4$, imágenes PAN urbanas.

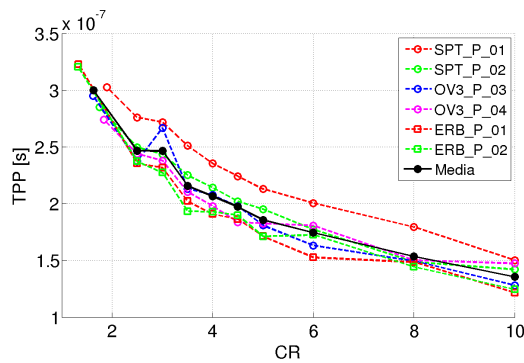


Fig. J.1. TPP vs. CR para IDC_{01} .

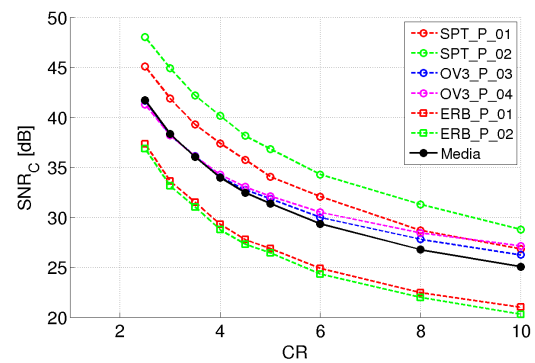


Fig. J.2. SNR_C vs. CR para IDC_{01} .

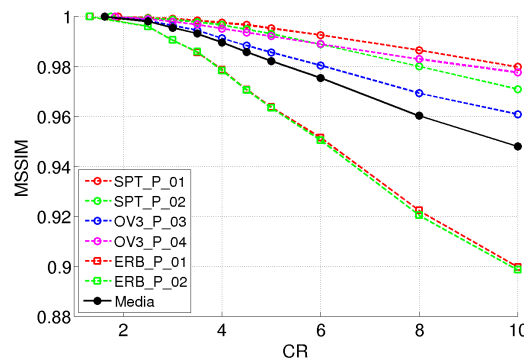


Fig. J.3. $MSSIM$ vs. CR para IDC_{01} .

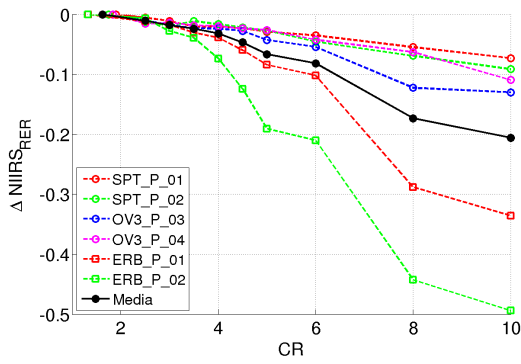


Fig. J.4. $\Delta NIIRS_{RER}$ vs. CR para IDC_01.

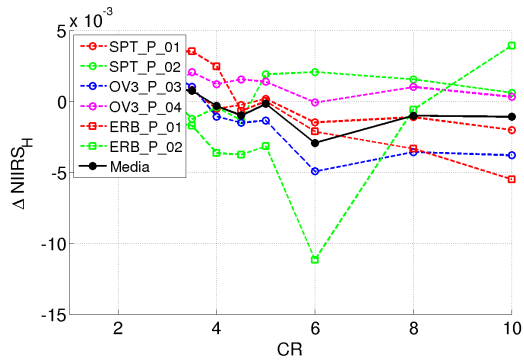


Fig. J.5. $\Delta NIIRS_H$ vs. CR para IDC_01.

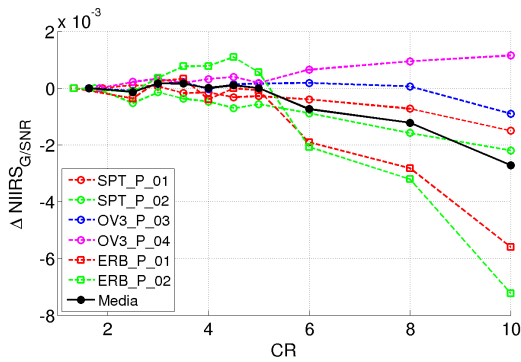


Fig. J.6. $\Delta NIIRS_{SNR}$ vs. CR para IDC_01.

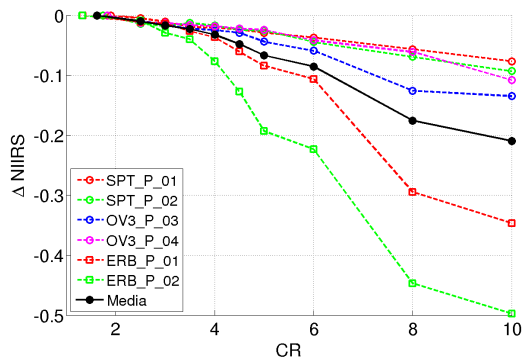


Fig. J.7. $\Delta NIIRS$ vs. CR para IDC_01.

Caso de prueba IDC_02: DWT flotante, $M_{si}=M/4$, imágenes PAN urbanas.

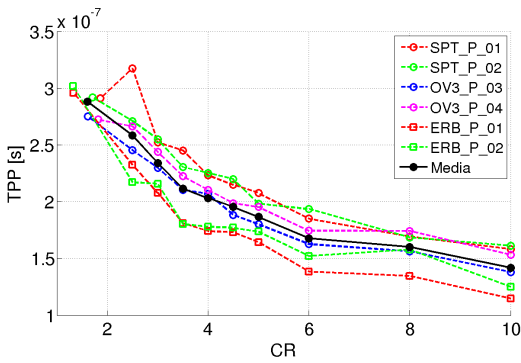


Fig. J.8. TPP vs. CR para IDC_02.

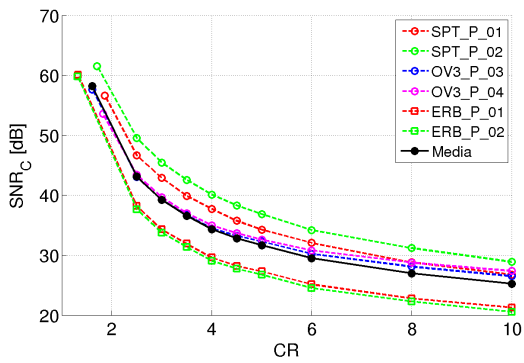


Fig. J.9. SNR_C vs. CR para IDC_02.

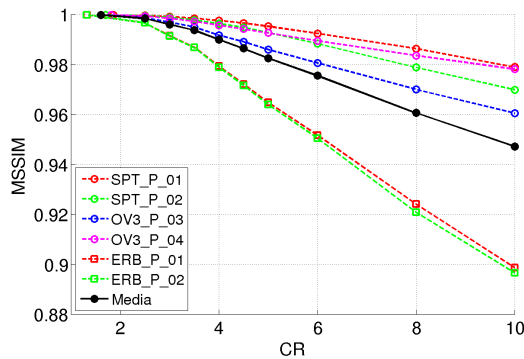


Fig. J.10. MSSIM vs. CR para IDC_02.

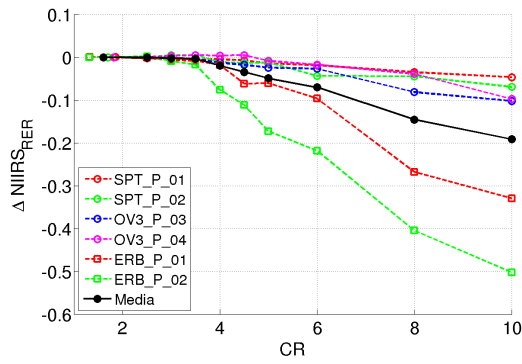


Fig. J.11. $\Delta NIIRS_{RER}$ vs. CR para IDC_02.

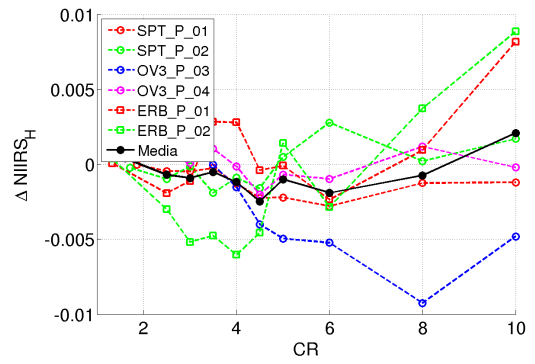


Fig. J.12. $\Delta NIIRS_H$ vs. CR para IDC_02.

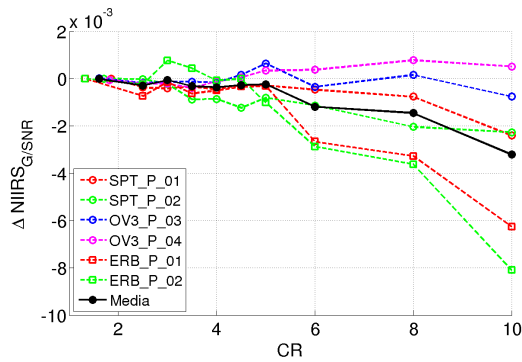


Fig. J.13. $\Delta NIIRS_{SNR}$ vs. CR para IDC_02.

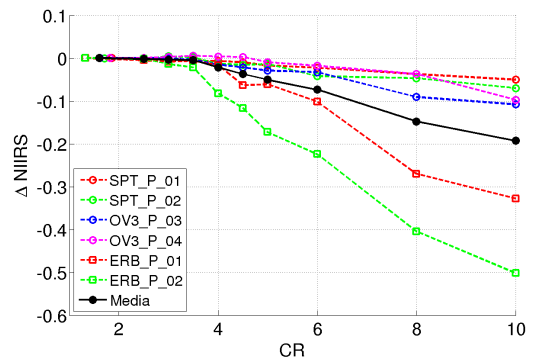


Fig. J.14. $\Delta NIIRS$ vs. CR para IDC_02.

Caso de prueba IDC_03: Barrido para la cantidad de filas por subimagen (M_{si}), DWT entera, $CR=0$, imágenes PAN urbanas.

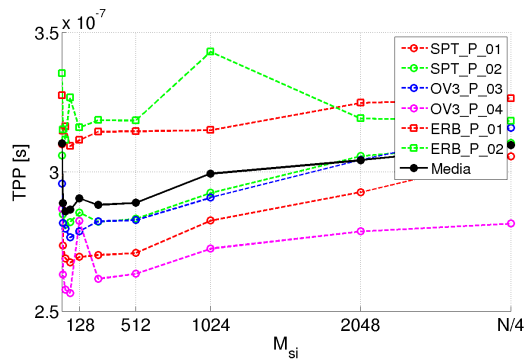


Fig. J.15. TPP vs. M_{si} para IDC_03.

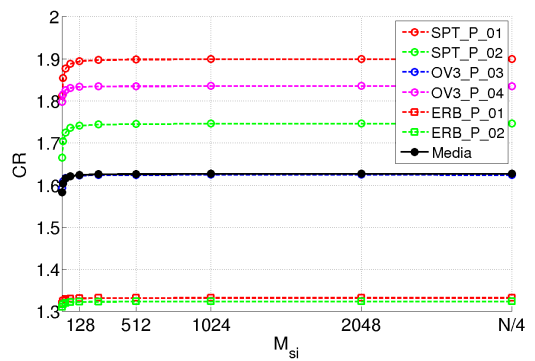


Fig. J.16. CR vs. M_{si} para IDC_03.

Caso de prueba IDC_04: Barrido para la cantidad de filas por subimagen (M_{si}), DWT entera, $CR=10$, imágenes PAN urbanas.

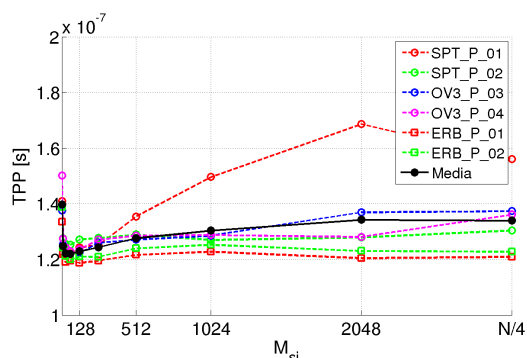


Fig. J.17. TPP vs. M_{si} para IDC_04.

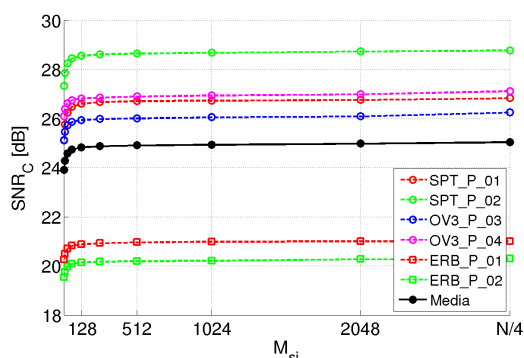


Fig. J.18. SNR_C vs. M_{si} para IDC_04.

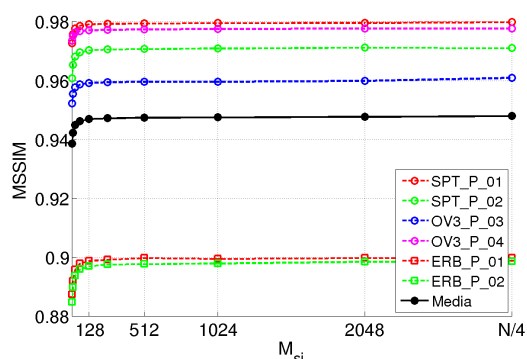


Fig. J.19. $MSSIM$ vs. M_{si} para IDC_04.

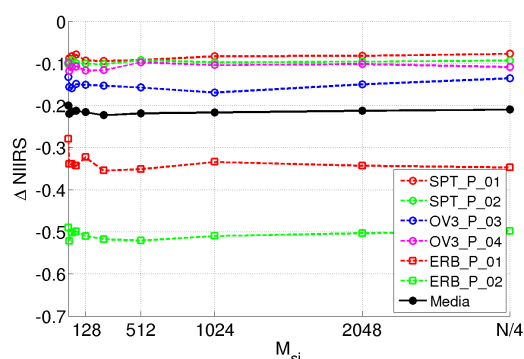


Fig. J.20. $\Delta NIIRS$ vs. M_{si} para IDC_04.

Caso de prueba IDC_05: DWT entera, $M_{si}=32$, imágenes PAN urbanas.

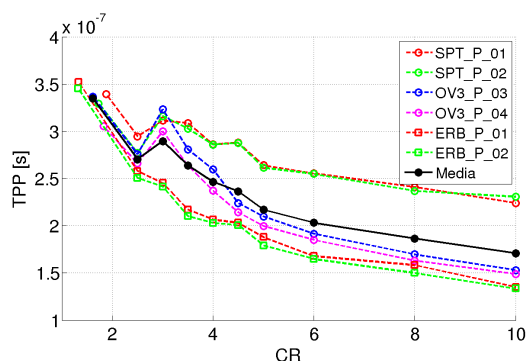


Fig. J.21. TPP vs. CR para IDC_05.

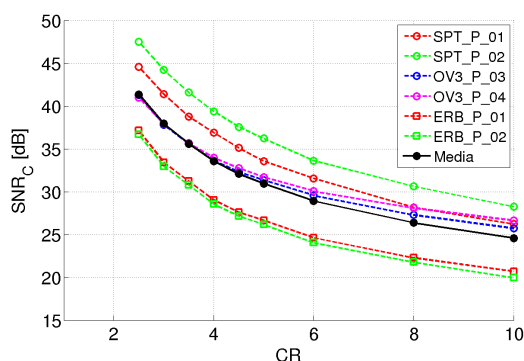


Fig. J.22. SNR_C vs. CR para IDC_05.

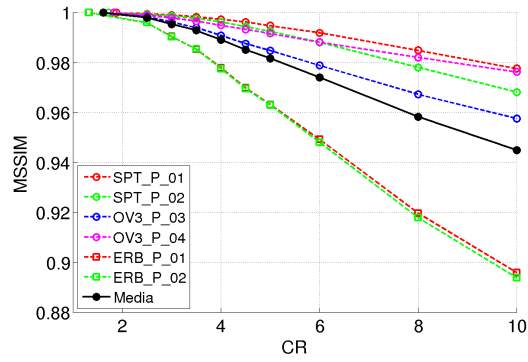


Fig. J.23. $MSSIM$ vs. CR para IDC_{05} .

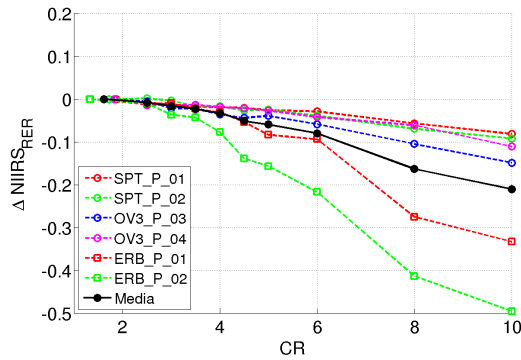


Fig. J.24. $\Delta NIIRS_{RER}$ vs. CR para IDC_{05} .

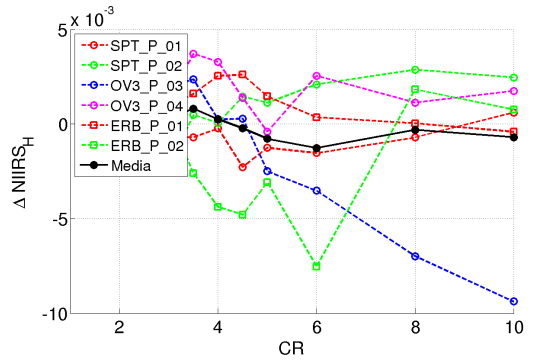


Fig. J.25. $\Delta NIIRS_H$ vs. CR para IDC_{05} .

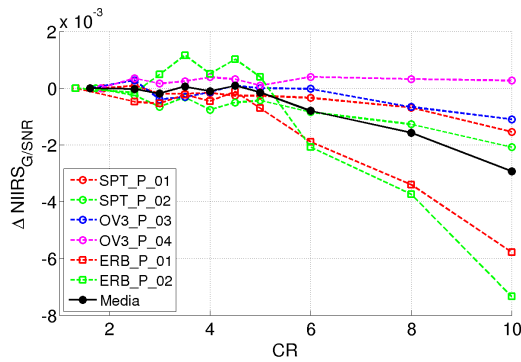


Fig. J.26. $\Delta NIIRS_{SNR}$ vs. CR para IDC_{05} .

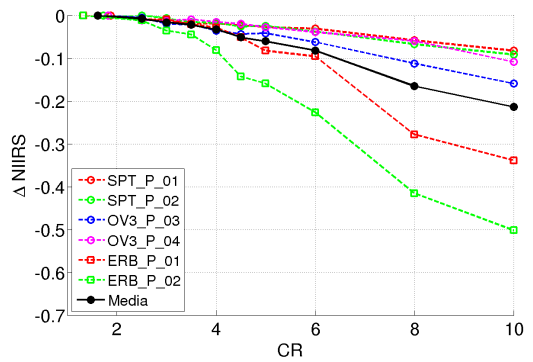


Fig. J.27. $\Delta NIIRS$ vs. CR para IDC_{05} .

Caso de prueba IDC_{06} : DWT entera, $M_{si}=512$, imágenes PAN urbanas.

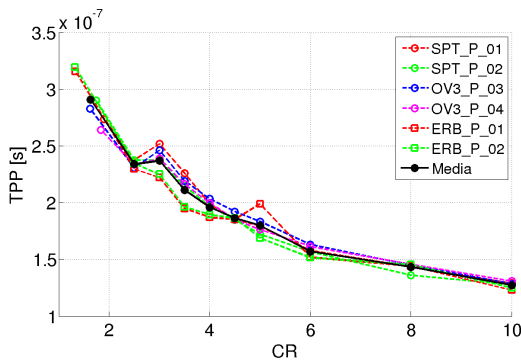


Fig. J.28. TPP vs. CR para IDC_{06} .

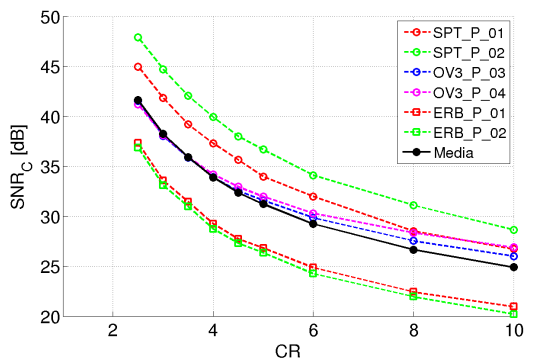


Fig. J.29. SNR_C vs. CR para IDC_{06} .

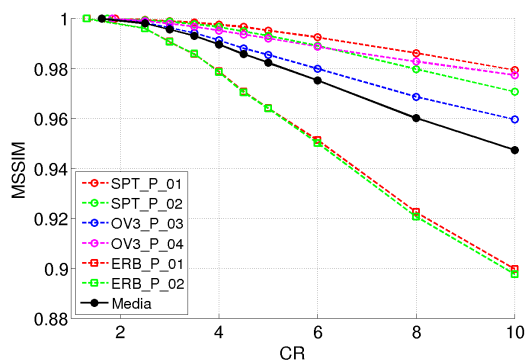


Fig. J.30. $MSSIM$ vs. CR para IDC_{06} .

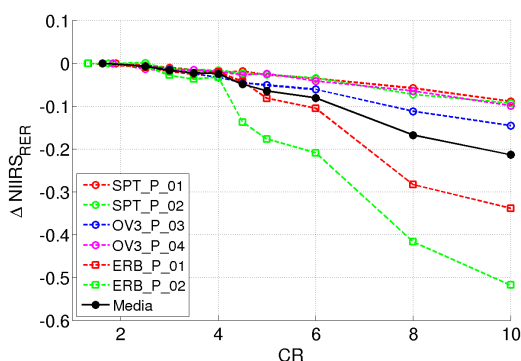


Fig. J.31. $\Delta NIIRS_{RER}$ vs. CR para IDC_{06} .

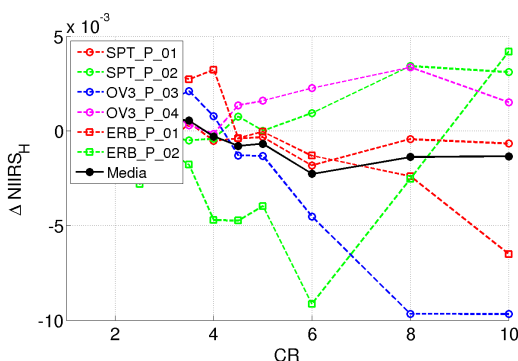


Fig. J.32. $\Delta NIIRS_H$ vs. CR para IDC_{06} .

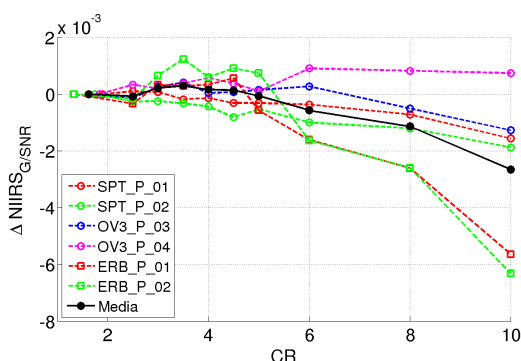


Fig. J.33. $\Delta NIIRS_{SNR}$ vs. CR para IDC_{06} .

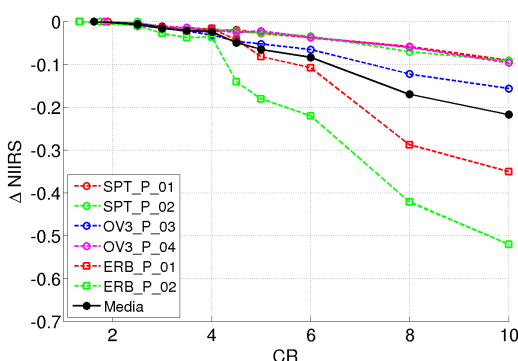


Fig. J.34. $\Delta NIIRS$ vs. CR para IDC_{06} .

Caso de prueba IDC_{21} : DWT entera, $M_{si}=512$, imágenes PAN rurales.

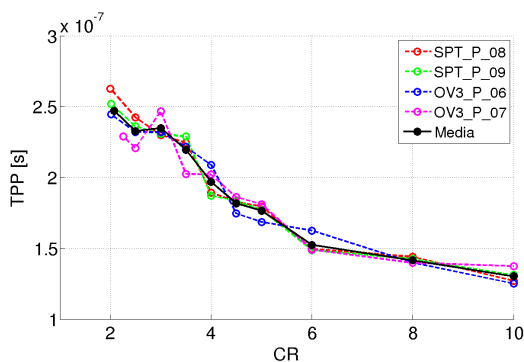


Fig. J.35. TPP vs. CR para IDC_{21} .

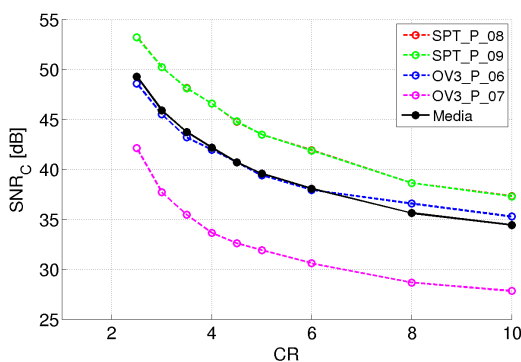


Fig. J.36. SNR_C vs. CR para IDC_{21} .

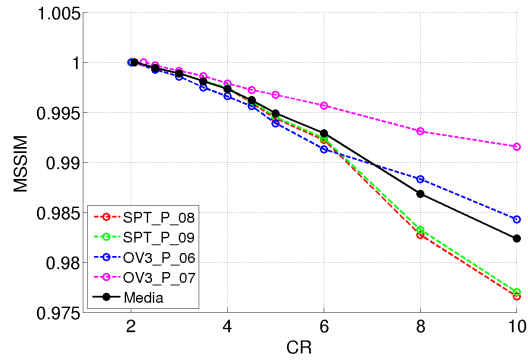


Fig. J.37. MSSIM vs. CR para IDC_21.

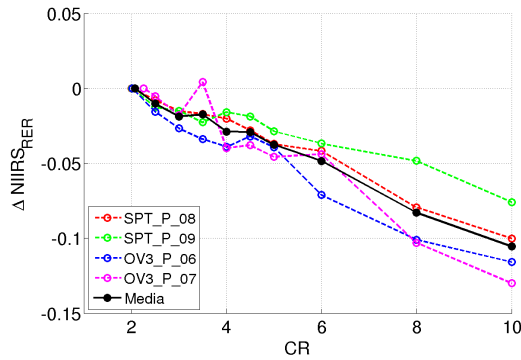


Fig. J.38. $\Delta NIIRS_{RER}$ vs. CR para IDC_21.

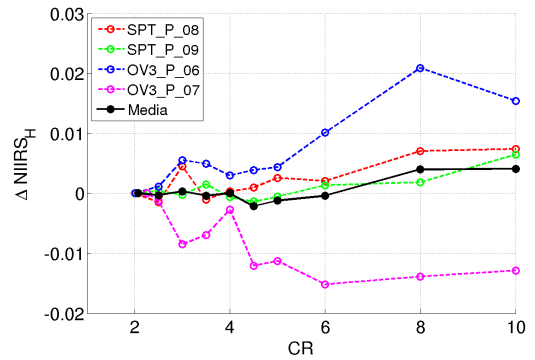


Fig. J.39. $\Delta NIIRS_H$ vs. CR para IDC_21.

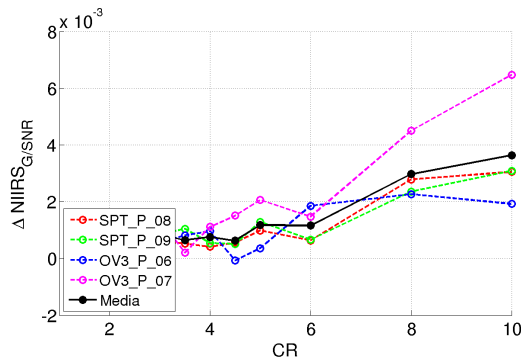


Fig. J.40. $\Delta NIIRS_{SNR}$ vs. CR para IDC_21.

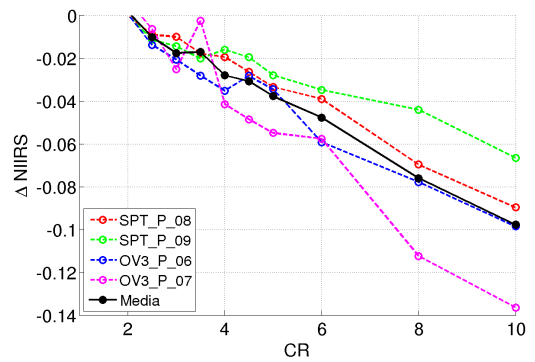


Fig. J.41. $\Delta NIIRS$ vs. CR para IDC_21.

Caso de prueba IDC_23: Barrido para la cantidad de filas por subimagen (M_{si}), DWT entera, $CR=0$, imágenes PAN rurales.

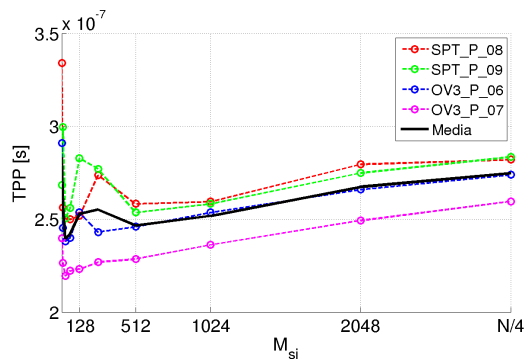


Fig. J.42. TPP vs. M_{si} para IDC_23.

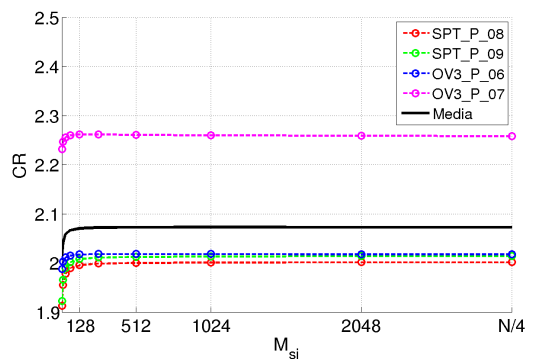


Fig. J.43. CR vs. M_{si} para IDC_23.

Caso de prueba IDC_24: Barrido para la cantidad de filas por subimagen (M_{si}), DWT entera, $CR=10$, imágenes PAN rurales.

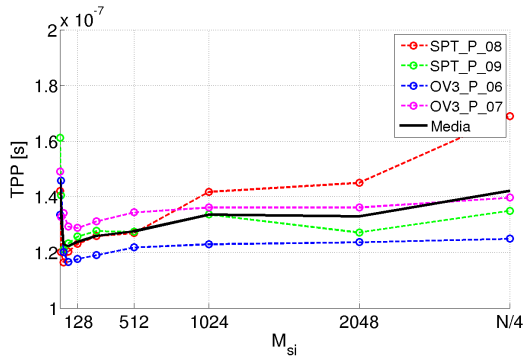


Fig. J.44. TPP vs. M_{si} para IDC_24.

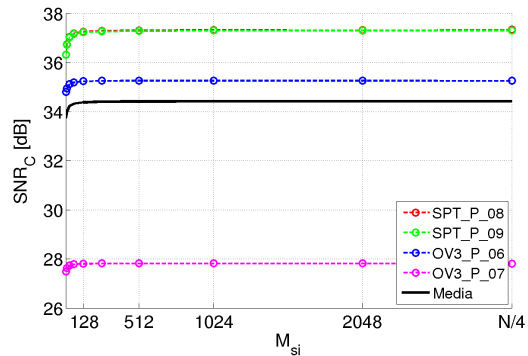


Fig. J.45. SNR_C vs. M_{si} para IDC_24.

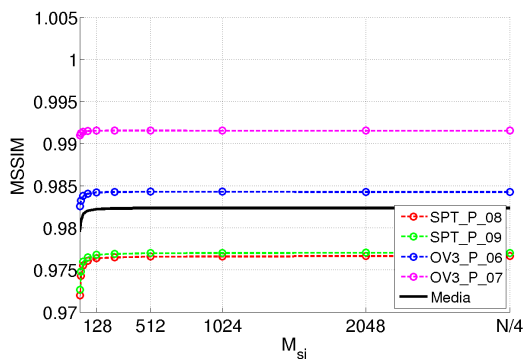


Fig. J.46. $MSSIM$ vs. M_{si} para IDC_24.

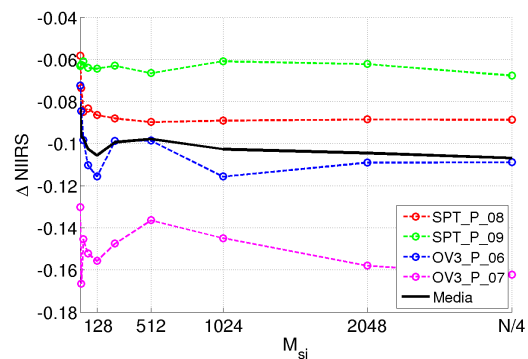


Fig. J.47. $\Delta NIIRS$ vs. M_{si} para IDC_24.

Caso de prueba IDC_31: DWT entera, $M_{si}=512$, imágenes MS urbanas.

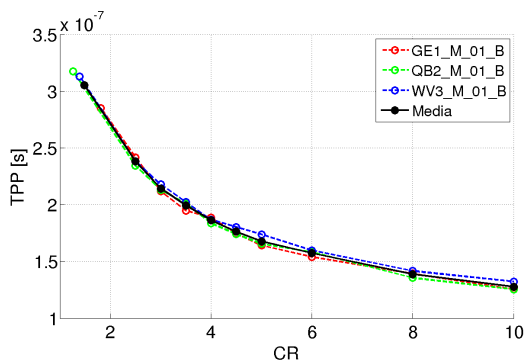


Fig. J.48. TPP vs. CR para IDC_31.

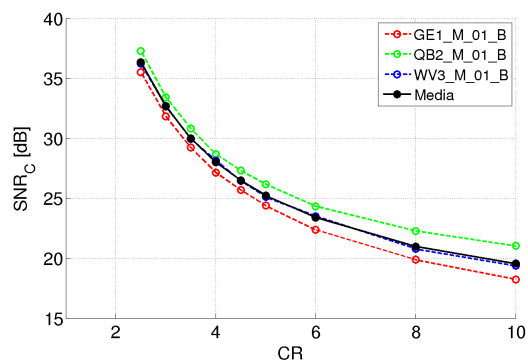


Fig. J.49. SNR_C vs. CR para IDC_31.

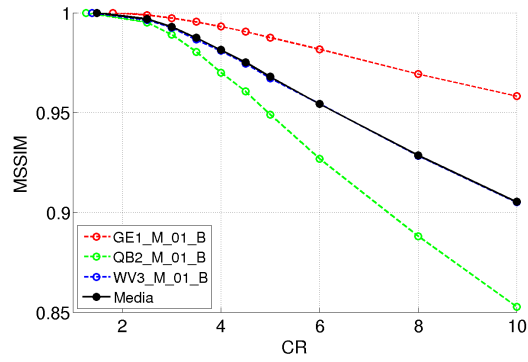


Fig. J.50. $MSSIM$ vs. CR para IDC_{31} .

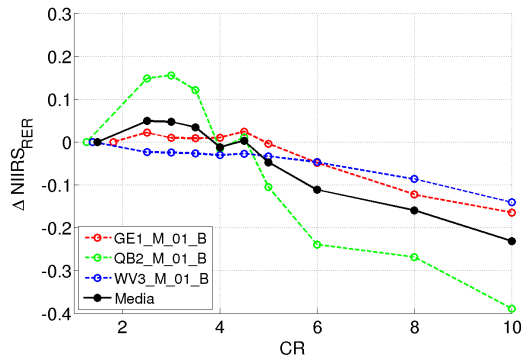


Fig. J.51. $\Delta NIIRS_{RER}$ vs. CR para IDC_{31} .

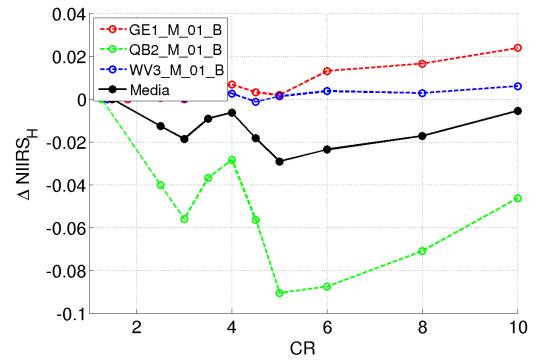


Fig. J.52. $\Delta NIIRS_H$ vs. CR para IDC_{31} .

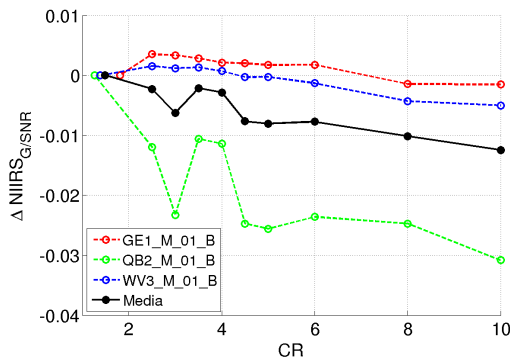


Fig. J.53. $\Delta NIIRS_{SNR}$ vs. CR para IDC_{31} .

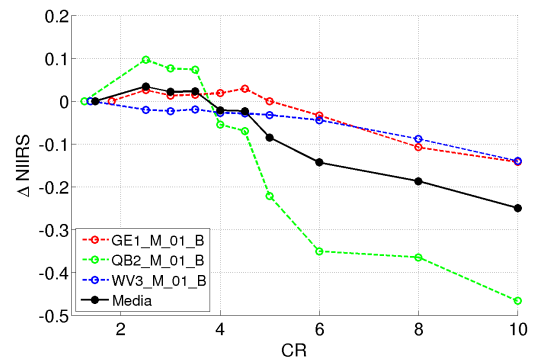


Fig. J.54. $\Delta NIIRS$ vs. CR para IDC_{31} .

Caso de prueba IDC_{41} : DWT entera, $M_{si}=512$, imágenes MS rurales.

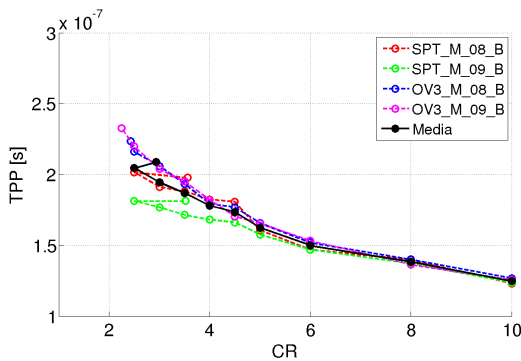


Fig. J.55. TPP vs. CR para IDC_{41} .

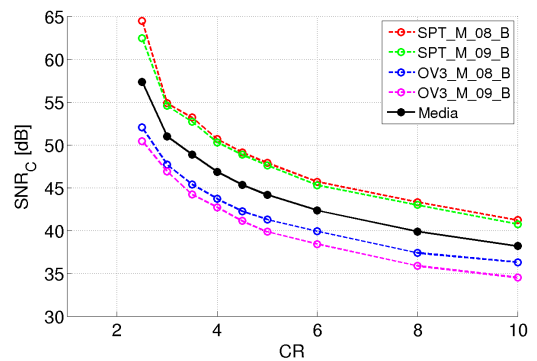


Fig. J.56. SNR_C vs. CR para IDC_{41} .

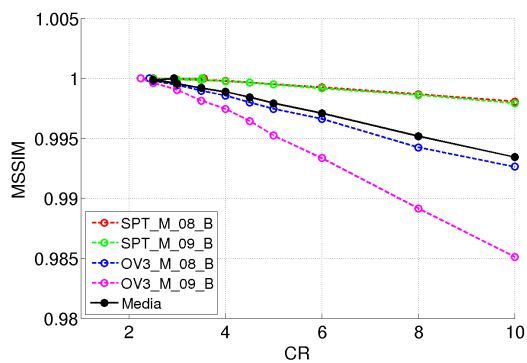


Fig. J.57. $MSSIM$ vs. CR para IDC_{41} .

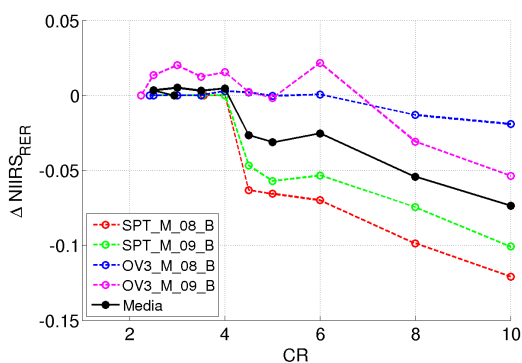


Fig. J.58. $\Delta NIIRS_{RE}$ vs. CR para IDC_{41} .

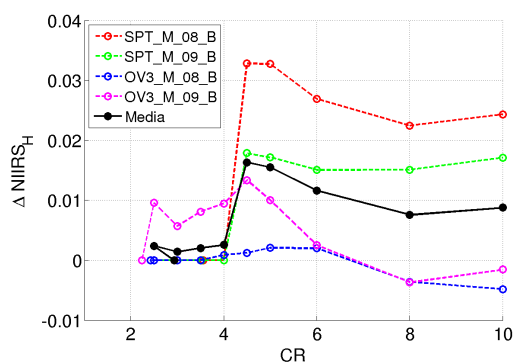


Fig. J.59. $\Delta NIIRS_H$ vs. CR para IDC_{41} .

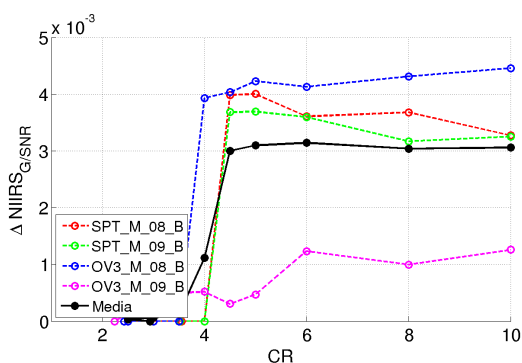


Fig. J.60. $\Delta NIIRS_{SNR}$ vs. CR para IDC_{41} .

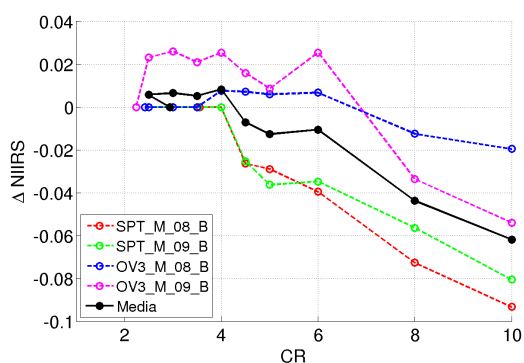


Fig. J.61. $\Delta NIIRS$ vs. CR para IDC_{41} .

Bibliografía

- [1] C. Lambert-Nebout and G. Moury, “A survey of on-board image compression for CNES space missions,” in *IEEE 1999 International Geoscience and Remote Sensing Symposium (IGARSS)*, Institute of Electrical & Electronics Engineers (IEEE), 1999.
- [2] C. Thiebaut and R. Camarero, “CNES studies for on-board compression of high-resolution satellite images,” in *Satellite Data Compression*, pp. 29–46, Springer Science + Business Media, sep 2011.
- [3] G. Yu, T. Vladimirova, and M. N. Sweeting, “Image compression systems on board satellites,” *Acta Astronautica*, vol. 64, pp. 988–1005, may 2009.
- [4] P. Lier, G. A. Moury, C. Latry, and F. Cabot, “Selection of the SPOT5 image compression algorithm,” in *Earth Observing Systems III*, SPIE, oct 1998.
- [5] I. Blanes, E. Magli, and J. Serra-Sagrasta, “A tutorial on image compression for optical space imaging systems,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 2, pp. 8–26, sep 2014.
- [6] L. N. Faria, L. M. G. Fonseca, and M. H. M. Costa, “Performance evaluation of data compression systems applied to satellite imagery,” *Journal of Electrical and Computer Engineering*, vol. 2012, pp. 1–15, 2012.
- [7] CONAE, “Plan Espacial Nacional, Argentina en el Espacio 2004 – 2015, actualización 2010 – 2015,” 2010.
- [8] CONAE, “Serie SARE.” <http://www.conae.gov.ar/index.php/espanol/misiones-satelitales/sare/objetivos>. Last access 07 - 06 - 2017.
- [9] G. Joseph, *Building Earth Observation Cameras*. CRC Press, 2015.

- [10] S.-E. Qian, *Optical Satellite Data Compression and Implementation*. SPIE Press, 2013.
- [11] B. V. Brower, M. A. Cosgrove, D. H. Lewis, G. R. VanLare, T. J. Fisher, and D. M. Charneski, “Advanced space-qualified downlink image compression ASIC for commercial remote sensing applications,” in *Applications of Digital Image Processing XXIII* (A. G. Tescher, ed.), SPIE, dec 2000.
- [12] G. Tyc *et al.*, “RAPIDEYE - an earth observation smallsat constellation for daily agricultural monitoring,” *AIAA/USU Conference on Small Satellites*, 2003.
- [13] Astrium, “SPOT 6 & SPOT 7 imagery - user guide, version 1.0,” 2013.
- [14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. PEARSON, 3rd ed., 2008.
- [15] D. Salomon, *Data Compression: The Complete Reference*. Springer, 4th ed., 2007.
- [16] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann, 4 ed., 2012.
- [17] D. S. Taubman and M. W. Marcellin, “JPEG2000: standard for interactive imaging,” *Proceedings of the IEEE*, vol. 90, pp. 1336–1357, aug 2002.
- [18] P. Schelkens, A. Skodras, and T. Ebrahimi, eds., *The JPEG 2000 Suite*. Wiley, 2009.
- [19] E. Delp and O. Mitchell, “Image compression using block truncation coding,” *IEEE Transactions on Communications*, vol. 27, pp. 1335–1342, sep 1979.
- [20] A. Kumar, , R. Kumaran, S. Paul, and S. Mehta, “ADPCM image compression techniques for remote sensing applications,” *International Journal of Information Engineering and Electronic Business*, vol. 7, pp. 26–31, may 2015.
- [21] “Lossless data compression,” Recommended Standard CCSDS 121.0-B-2 Blue Book, CCSDS, may 2012.
- [22] U.S. Geological Survey, “Landsat 8 (L8) data user’s handbook, version 2.0,” 2012.

-
- [23] T. Smith, “Wideband Advanced Recorder and Processor (WARP) validation report,” tech. rep., NASA Goddard Space Flight Center, mar 2002.
- [24] ICs LLC, *Universal Source Encoder for Space - USES*, nov 2005.
- [25] L. Miles, P. Yeh, J. Feeley, D. Fisher, S. Whitaker, and G. Maki, “Over 3Gb/s universal lossless compressor for space use.” NASA-NEPP, aug 2011.
- [26] G. Wallace, “The JPEG still picture compression standard,” *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [27] M. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.
- [28] M. J. Weinberger, G. Seroussi, and G. Sapiro, “LOCO-I: A low complexity, context-based, lossless image compression algorithm,” in *Data Compression Conference, 1996. DCC’96. Proceedings*, pp. 140–149, IEEE, 1996.
- [29] A. Kiely and M. Klimesh, “The ICER progressive wavelet image compressor,” *IPN Progress Report*, vol. 42, no. 155, pp. 1–46, 2003.
- [30] N. Navarathinam, A. Cawthorne, L. Gomes, M. Sweeting, and J. Paffet, “DMC3 and Carbonite-1: Two sides of small satellites,” 2016.
- [31] S. J. Visser, A. S. Dawood, and J. A. Williams, “FPGA based satellite adaptive image compression system,” *Journal of Aerospace Engineering*, 2003.
- [32] A. Baskurt, C. Odet, and R. Goutte, “Image coding on the Phobos space probe,” *Signal Processing: Image Communication*, 1994.
- [33] P.-S. Yeh, P. Armbruster, A. Kiely, B. Masschelein, G. Moury, C. Schaefer, and C. Thiebaut, “The new CCSDS image compression recommendation,” in *Proc. IEEE Aerospace Conf*, pp. 4138–4145, mar 2005.
- [34] “Image data compression,” Recommended Standard 122.0-B-1 Blue Book, CCSDS, nov 2005.
- [35] L. Li, G. Zhou, B. Fiethe, H. Michalik, and B. Osterloh, “Efficient implementation of the CCSDS 122.0-B-1 compression standard on a space-qualified field programmable gate array,” *Journal of Applied Remote Sensing*, vol. 7, p. 074595, aug 2013.

- [36] A. Gimona, L. Poggio, I. Aalders, and M. Aitkenhead, “The effect of image compression on synthetic PROBA-V images,” *International Journal of Remote Sensing*, vol. 35, pp. 2639–2653, mar 2014.
- [37] A. Lin, C. F. Chang, M. C. Lin, L. J. Jan, and H. Ren, “Implementation of CCSDS data compression for remote sensing image,” in *Satellite Data Compression, Communications, and Processing VI*, SPIE, aug 2010.
- [38] L. Guanter *et al.*, “The EnMAP spaceborne imaging spectroscopy mission for earth observation,” *MDPI, Remote Sensing*, vol. 7, pp. 8830–8857, jul 2015.
- [39] C. Latry, S. Fourest, and C. Thiebaut, “Restoration techniques for PLEIADES-HR panchromatic images,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012.
- [40] Airbus Defense & Space, “CORECI, an integrated COmpression REcording and CIphering solution for earth observation stellites,” 2014.
- [41] “Image data compression,” Informational Report CCSDS 120.1-G-2 Green Book, CCSDS, feb 2015.
- [42] S. Purcell, “The C-cube CL550-JPEG image compression processor,” in *COMPCON Spring 91 Digest of Papers*, Institute of Electrical & Electronics Engineers (IEEE), 1991.
- [43] C. Alonso and C. Hofmann, “SAC-C mission.” CONAE, 2002.
- [44] B. Huang, ed., *Satellite Data Compression*. Springer, 2011.
- [45] C. Thiebaut, E. Christophe, D. Lebedeff, and C. Latry, “CNES studies of on-board compression for multispectral and hyperspectral images,” in *Satellite Data Compression, Communications, and Archiving III*, SPIE, sep 2007.
- [46] “Lossless multispectral & hyperspectral image compression,” Recommended Standard CCSDS 123.0-B-1 Blue Book, CCSDS, may 2012.
- [47] “Lossless multispectral & hyperspectral image compression,” Informational Report CCSDS 120.2-G-1 Green Book, CCSDS, dec 2015.
- [48] D. Shah, K. Bera, and S. Joshi, “Software implementation of CCSDS recommended hyperspectral lossless image compression,” *International*

-
- Journal of Image, Graphics and Signal Processing*, vol. 7, pp. 35–41, mar 2015.
- [49] K. Manthey, D. Krutz, and B. Juurlink, “A new real-time system for image compression on-board satellites,” *DLR*, 2015.
- [50] R. Camarero, C. Thiebaut, P. Dejean, and A. Speciel, “CNES studies for on-board implementation via HLS tools of a cloud-detection module for selective compression,” in *Satellite Data Compression, Communications, and Processing VI*, SPIE, aug 2010.
- [51] “Information technology – lossless and near-lossless compression of continuous-tone still images: Baseline,” Standard ISO/IEC: 14495-1:1999, ISO, 1999. Also ITU Recommendation T.87.
- [52] E. Feig and S. Winograd, “Fast algorithms for the discrete cosine transform,” *IEEE Transactions on Signal processing*, vol. 40, no. 9, pp. 2174–2193, 1992.
- [53] “Information technology – digital compression and coding of continuous-tone still images: Requirements and guidelines,” Standard ISO/IEC 10918-1:1994, ISO, 1994. Also ITU Recommendation T.81.
- [54] “Information technology – JPEG 2000 image coding system: Core coding system,” Standard ISO/IEC 15444-1:2004, ISO, 2002. Also ITU Recommendation T.800.
- [55] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer, 2013.
- [56] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Transactions on Image Processing*, vol. 9, pp. 1158–1170, jul 2000.
- [57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “The SSIM index for image quality assessment.” <https://ece.uwaterloo.ca/~z70wang/research/ssim/>. Last access 09 - 01 - 2017.
- [58] Z. Wang *et al.*, “Image quality assessment: From error measurement to structural similarity,” *IEEE Transactions on image processing*, 2004.
- [59] Z. Wang and A. Bovik, “Mean squared error: Love it or leave it? a new look at signal fidelity measures,” *IEEE Signal Processing Magazine*, vol. 26, pp. 98–117, jan 2009.

- [60] R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing*. Academic Press, 2006.
- [61] G. D. Boreman, *Modulation Transfer Function in Optical and ElectroOptical Systems*. SPIE Publications, 2001.
- [62] R. R. Auelmann, “Image quality metrics,” 2012.
- [63] J. M. Irvine, “National Imagery Interpretability Rating Scales (NIIRS): Overview and methodology,” in *Airborne Reconnaissance XXI*, SPIE, nov 1997.
- [64] R. D. Fiete and T. Tantaló, “Comparison of SNR image quality metrics for remote sensing systems,” *Optical Engineering*, vol. 40, p. 574, apr 2001.
- [65] “Civil NIIRS Reference Guide - Appendix II: Additional NIIRS Criteria.” https://fas.org/irp/imint/niirs_c/app2.htm. Last access 22 - 08 - 2017.
- [66] J. C. Leachtenauer, W. Malila, J. Irvine, and L. Colburn, “General Image-Quality Equation: GIQE,” *OSA applied optics*, 1997.
- [67] E. Chuvieco, *Teledetección ambiental: La observación de la tierra desde el espacio*. Ariel Ciencias, Editorial Ariel, 2010.
- [68] D. Léger, F. Viallefont, and P. Déliot, “On-orbit MTF assessment of satellite cameras,” in *Post-Launch Calibration of Satellite Sensors* (S. A. Morain and A. M. Budge, eds.), pp. 67–76, International Society for Photogrammetry and Remote Sensing (ISPRS) Book Series, 2004.
- [69] D. Helder, T. Choi, and M. Rangaswamy, “In-flight characterization of spatial quality using point spread functions,” in *Post-Launch Calibration of Satellite Sensors* (S. A. Morain and A. M. Budge, eds.), pp. 151–170, International Society for Photogrammetry and Remote Sensing (ISPRS) Book Series, 2004.
- [70] A. P. Tzannes and J. M. Mooney, “Measurement of the modulation transfer function of infrared cameras,” *Optical Engineering*, vol. 34, no. 6, pp. 1808–1817, 1995.
- [71] T. Choi and D. L. Helder, “Generic sensor modeling for modulation transfer function (MTF) estimation,” *Pecora 16 Global Priorities in Land Remote Sensing*, 2005.

-
- [72] R. Ryan, B. Baldrige, R. A. Schowengerdt, T. Choi, D. L. Helder, and S. Blonski, “IKONOS spatial resolution and image interpretability characterization,” *Remote Sensing of Environment*, vol. 88, pp. 37–52, nov 2003.
- [73] S. Blonski, K. Ross, and M. Pagnutti, “Spatial resolution characterization for aerial digital imagery.” SSC-NASA, mar 2006.
- [74] S. Slawomir Blonski, “Spatial resolution characterization for QuickBird image products 2003-2004 season.” SSC-NASA, nov 2004.
- [75] M. Crespi and L. D. Vendictis, “A procedure for high resolution satellite imagery quality assessment,” *Sensors*, vol. 9, pp. 3289–3313, may 2009.
- [76] T. Kim, H. Kim, and H. Kim, “Image-based estimation and validation of NIIRS for high-resolution satellite images,” *American Society of Photogrammetry and Remote Sensing (ASPRS)*, 2008.
- [77] T. Kim *et al.*, “Automated image interpretability assessment by edge profile analysis of natural targets,” *American Society of Photogrammetry and Remote Sensing (ASPRS)*, 2010.
- [78] J.-I. Kim and T.-J. Kim, “Development of a natural target-based edge analysis method for NIIRS estimation,” *Korean Journal of Remote Sensing*, vol. 27, pp. 587–599, oct 2011.
- [79] M. A. Ali, F. Eltohamy, and G. I. Salama, “Estimation of NIIRS, for high resolution satellite images, using the simplified GIQE,” *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, 2016.
- [80] H.-M. Chen, E. Blasch, K. Pham, Z. Wang, and G. Chen, “An investigation of image compression on NIIRS rating degradation through automated image analysis,” in *Sensors and Systems for Space Applications IX*, SPIE, may 2016.
- [81] M. Estriebeau and P. Magnan, “Fast MTF measurement of CMOS imagers using ISO 12333 slanted-edge methodology,” in *Detectors and Associated Signal Processing* (J.-P. Chatard and P. N. J. Dennis, eds.), SPIE-Intl Soc Optical Eng, feb 2004.
- [82] J. Roland, “A study of slanted-edge MTF stability and repeatability,” in *SPIE/IS&T Electronic Imaging*, pp. 93960L–93960L, International Society

- for Optics and Photonics, 2015.
- [83] K. J. Hermiston and D. M. Booth, “NIIRS and objective image quality measures,” in *International Conference on Computer Analysis of Images and Patterns*, pp. 385–394, Springer, 1999.
- [84] S. T. Thurman and J. R. Fienup, “Analysis of the general image quality equation,” in *Visual Information Processing XVII*, SPIE, apr 2008.
- [85] R. Hellerman, “Growing demand for commercial very high resolution imaging satellite.” International Commercial Remote Sensing Symposium, mar 2014.
- [86] B. Baugh, S. Malitz, N. Longbotham, C. Adams, S. Starr, and M. Clark, “WorldView-3: Seeing the world in new ways,” nov 2014.
- [87] D. Globe, “WorldView-3 data sheet,” may 2014.
- [88] G. A. Boggione, L. M. G. Fonseca, S. Bosch-Puig, and F. J. Ponzoni, “Simulation of Mux camera images of the Brazil-China satellite earth resources (CBERS-3),” *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 607–612, 2012.
- [89] I. International, “EROS imagery products guide,” 2010.
- [90] GeoEye, “OrbView-3 commercial satellite imagery product catalog, rev. 1/23/06,” 2006.
- [91] “HP labs LOCO-I/JPEG-LS home page.” http://www.labs.hp.com/research/info_theory/loco/. Last access 10 - 07 - 2017.
- [92] “Independent JPEG Group.” <http://www.ijg.org/>. Last access 10 - 07 - 2017.
- [93] “OpenJPEG.” <http://www.openjpeg.org/>. Last access 10 - 07 - 2017.
- [94] M. D. Adams and F. Kossentini, “JasPer: a software-based JPEG-2000 codec implementation,” in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 2, pp. 53–56 vol.2, sep 2000.
- [95] “CCSDS Image Data Compression implementation - University of Nebraska-Lincoln.” <http://hyperspectral.unl.edu/index.html>. Last access 10 - 07 - 2017.

-
- [96] L. Zhai and X. Tang, “Fuzzy comprehensive evaluation method and its application in subjective quality assessment for compressed remote sensing images,” in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, vol. 1, pp. 145–148, aug 2007.
- [97] H. Hihara, M. Sato, K. Fukasawa, H. Sase, Y. Osawa, and N. Ito, “High speed image data compression processor for Advanced Land Observing Satellite (ALOS),” in *2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering.*, IEEE, 2010.
- [98] N. Ismailoglu *et al.*, “A real time image processing subsystem: GEZGIN,” *16th Conference on Small Satellites*, 2006.
- [99] N. Ismailoglu, O. Benderli, S. Yesil, R. Sever, B. Okcan, O. Sengul, and R. Oktem, “GEZGIN & GEZGIN-2: Adaptive real-time image processing subsystems for earth observing small satellites,” in *First NASA/ESA Conference on Adaptive Hardware and Systems*, Institute of Electrical & Electronics Engineers (IEEE), 2006.
- [100] G. Yüksel, Ö. Belce, H. Urhan, L. Gomes, A. Bradford, N. Bean, and A. da Silva, “Bilsat-1: First year in orbit-operations and lessons learned,” 2004.
- [101] P. Yeh, “Implementation of CCSDS lossless data compression for space and data archive applications,” in *SpaceOps 2002 Conference*, AIAA, oct 2002.
- [102] J. Blamont and C. Lambert-Nebout, “Clementine: On-board image compression,” in *Lunar and Planetary Science Conference*, vol. 26, 1995.
- [103] T. Smith and J. Kessler, “Earth Orbiter 1: Wideband Advanced Recorder and Processor (WARP),” 1999.
- [104] C.-F. Chang, “Field-programmable gate array implementation of Consultative Committee for Space Data Systems image data compression,” *Journal of Applied Remote Sensing*, vol. 6, p. 063551, jul 2012.
- [105] A. Lin, “Hardware implementation of a real-time image data compression for satellite remote sensing,” in *Remote Sensing-Advanced Techniques and Platforms*, InTech, 2012.
- [106] “Ikonos-2 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/i/ikonos-2>. Last access 17 - 08 - 2017.
- [107] “IMS-1 - eoPortal Directory.” <https://directory.eoportal.org/web/>

- eoportal/satellite-missions/i/ims-1. Last access 17 - 08 - 2017.
- [108] “Kazeosat-1 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/k/kazeosat-1>. Last access 17 - 08 - 2017.
- [109] “Landsat-8 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-8-ldcm>. Last access 17 - 08 - 2017.
- [110] “Meteosat-8 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/m/meteosat-second-generation>. Last access 17 - 08 - 2017.
- [111] T. Yamamoto, H. Hashimoto, S. Nishida, S. Kimura, and S. Nakasuka, “MICRO LABSAT-technology demonstration microsatellite for future missions,” 2004.
- [112] “MTI - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/m/mti>. Last access 17 - 08 - 2017.
- [113] “OrbView-3 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/o/orbview-3>. Last access 17 - 08 - 2017.
- [114] “PROBA-2 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/p/proba-2>. Last access 17 - 08 - 2017.
- [115] “PROBA-V - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/p/proba-v>. Last access 17 - 08 - 2017.
- [116] “Copernicus: Sentinel-2 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/copernicus-sentinel-2>. Last access 17 - 08 - 2017.
- [117] “SPOT-6 and SPOT-7 - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/s/spot-6-7>. Last access 17 - 08 - 2017.
- [118] “TRACE - eoPortal Directory.” <https://directory.eoportal.org/web/eoportal/satellite-missions/t/trace>. Last access 17 - 08 - 2017.

- [119] S. Zhou, C. Deng, B. Zhao, Y. Xia, Q. Li, and Z. Chen, “Remote sensing image compression: A review,” in *2015 IEEE International Conference on Multimedia Big Data*, Institute of Electrical & Electronics Engineers (IEEE), apr 2015.
- [120] M. Konnik and J. Welsh, “High-level numerical simulations of noise in CCD and CMOS photosensors: review and tutorial,” *arXiv preprint arXiv:1412.4031*, 2014.
- [121] D. L. Snyder, C. W. Helstrom, A. D. Lanterman, R. L. White, and M. Faisal, “Compensation for readout noise in CCD images,” *J. Opt. Soc. Am. A*, vol. 12, pp. 272–283, feb 1995.