

# CLASIFICACIÓN DE IMÁGENES DE SATÉLITE MEDIANTE AUTÓMATAS CELULARES



DEPARTAMENTO DE INFORMÁTICA  
UNIVERSIDAD DE ALMERÍA

## TESIS DOCTORAL

Presentada por

Antonio Moisés Espínola Pérez

para optar al grado de  
Doctor Ingeniero en Informática

Dirigida por

Dr. D. Luis F. Iribarne Martínez  
Profesor Titular de Universidad  
Departamento de Informática  
Universidad de Almería

Dra. Dña. Rosa M. Ayala Palenzuela  
Profesora Titular de Universidad  
Departamento de Informática  
Universidad de Almería

Dr. D. José Antonio Piedra Fernández  
Profesor Colaborador de Universidad  
Departamento de Informática  
Universidad de Almería

ALMERÍA, 21 DE NOVIEMBRE DE 2014

Escrito y editado en L<sup>A</sup>T<sub>E</sub>Xpor: Antonio Moisés Espínola Pérez  
Impresión: Murex Factoría de Color (Almería, España)

Noviembre de 2014

TESIS DOCTORAL

CLASIFICACIÓN DE IMÁGENES DE SATÉLITE  
MEDIANTE AUTÓMATAS CELULARES

ANTONIO MOISÉS ESPÍNOLA PÉREZ

Este documento ha sido generado con L<sup>A</sup>T<sub>E</sub>X.

Tesis doctoral: "Clasificación de Imágenes de Satélite mediante Autómatas Celulares"

Antonio Moisés Espinola Pérez  
Grupo de investigación de Informática Aplicada  
Departamento de Informática  
Universidad de Almería  
Almería, 21 de Noviembre de 2014  
<http://www.ual.es/personal/moises.espinola/>



**A mis padres,**  
*porque me habéis inculcado mi espíritu de sacrificio,  
a través de vuestro ejemplo.*

**A Sandra,**  
*porque me has iluminado con tu luz,  
en mis momentos de oscuridad.*



# AGRADECIMIENTOS

Cuando era adolescente, soñaba con ser astronauta, y de este modo poder observar la superficie de la Tierra desde el espacio. Aquel sueño lejano lo he podido cumplir, en parte, gracias al desarrollo de esta tesis doctoral, en la que he estado trabajando con imágenes de satélite de la *NASA*, observando por lo tanto la Tierra desde el espacio. Los últimos años de mi vida han sido fascinantes desde un punto de vista profesional, ya que he podido participar activamente en diversos proyectos internacionales de investigación, y he podido aplicar los conocimientos teóricos y prácticos, adquiridos durante mis años de estudios universitarios, a la resolución de problemas de distinta complejidad. Desde un punto de vista personal, esta experiencia ha supuesto una intensa y emocionante aventura en la que he conocido a personas, cuya profesionalidad y carisma, dejarán para siempre una profunda huella en mi memoria. A continuación quiero hacer una mención especial a la personas que más han inspirado mi labor de investigación.

En primer lugar, quiero mencionar a Luis Iribarne, mi verdadero maestro y mentor. Deseo agradecerle, por un lado, la gran oportunidad que me ofreció hace años, al invitarme a participar en esta apasionante aventura, y por otro lado, la enorme confianza que siempre ha depositado en mí. También quiero agradecerle toda la ayuda que me ha ofrecido durante estos años, el exquisito respeto con el que siempre me ha tratado y la cercanía en los momentos más difíciles. Si mi tesis ha dado frutos, ha sido porque Luis sembró en mí, hace años, la semilla del interés por el mundo de la investigación. Luis, es un gran honor tenerte como director de mi tesis doctoral.

En segundo lugar, deseo mencionar a Rosa Ayala. Se suele decir que detrás de un gran hombre, siempre hay una gran mujer. En este caso, es totalmente cierto, no sólo por haberme instruido en los aspectos fundamentales de la teledetección, sino también por haberme impregnado positivamente con su personalidad fuerte y optimista. Rosa, gracias de corazón por tus palabras de aliento en los momentos más difíciles.

En tercer lugar, quiero mencionar a José Antonio Piedra. Deseo agradecerle todas las ideas que, de manera continua, me ha sugerido para desarrollar y mejorar mi trabajo de investigación. En numerosas ocasiones, en las que no sabía cómo seguir avanzando en mi trabajo, he recibido su apoyo incondicional. José, muchas gracias por haberme ofrecido tu ayuda cada vez que la he necesitado.

En cuarto lugar, quiero mencionar a Saturnino Leguizamón, persona de naturaleza extremadamente sabia e inquieta, por la formación que me ofreció en las técnicas de los autómatas celulares. Para mí, mi cuarto director de tesis. Saturnino, muchas gracias por tu formación y sabios consejos.

También quiero destacar el gran apoyo que siempre he recibido por parte de los Hermanos de las Escuelas Cristianas de La Salle-Virgen del Mar, mi Colegio, y especialmente de los directores: Teodosio Oliver, Narciso Arozamena, Inocencio Lorite, Antonio Trujillo, Celerino Sanz y Miguel Tena, que me han permitido, durante años, asistir en período docente a los congresos en los que he tenido que impartir alguna conferencia sobre mis trabajos de investigación. Sin su apoyo a mi labor científica, este trabajo jamás habría salido a la luz. A vosotros, deseo expresaros mi más sincero agradecimiento.

Sería injusto no mencionar a mis amigos de La Salle, que me han escuchado hablar una y otra vez, durante años, sobre mi tesis doctoral. De manera muy especial quiero mencionar a José Carlos, y me vienen a la memoria muchas noches en la Jaima, y a Juan Pablo, por los incontables cafés durante los recreos de La Salle. También quiero mencionar a Juanje, Sergio, Juanjo, Nico, Salva, Antonio, Cristian, Guille, José, Inma, Ana, Maripi, Laura, Eva, María José, Veli, y muchos más. También sería injusto no mencionar a mi hermano David, y a mis amigos Fernando, Rubén, Acedo, Javi, Paco y José Manuel, que también han supuesto siempre un gran apoyo para mí. Muchas gracias, de todo corazón, por haber tenido tanta paciencia conmigo en los momentos más difíciles.

Quisiera dar las gracias también a mis amigos de la Universidad de Almería, Antonio López Márquez, José Antonio Álvarez Bermejo y Antonio Fernández Álvarez, que siempre me han apoyado y aconsejado durante mi labor de investigación. Para mí, siempre habéis sido un ejemplo a seguir (aunque nunca os lo haya dicho antes).

También, y no menos importante, quiero dar las gracias a mis padres, por haberme ofrecido la educación y formación necesaria para ser lo que soy en la actualidad. Todo, absolutamente todo, se lo debo a ellos. Por eso, a vosotros, os dedico esta tesis doctoral.

He dejado para el final a la persona que más me ha ayudado durante el desarrollo de mi tesis doctoral: mi compañera de viaje. Sandra, te debo tanto, que no puedo expresarlo con palabras. Por eso, esta tesis también te la dedico a ti.

Este trabajo de investigación ha sido desarrollado en el marco de los proyectos I+D SOLERES: “un sistema de información espacio-temporal para la gestión medioambiental basado en redes neuronales, agentes y componentes software” (TIN2007-61497) e iSOLERES: “una metodología para la recuperación y explotación de información medioambiental mediante interfaces de usuario evolutivas y cooperativas” (TIN2010-15588), subvencionados por la UE (FEDER) y el Ministerio de Ciencia e Innovación de España, en la actualidad Ministerio de Economía y Competitividad (MINECO), y desarrollados por el grupo de investigación de Informática Aplicada (TIC-211) del Departamento de Informática de la Universidad de Almería.

A. Moisés Espínola Pérez  
Grupo de investigación de Informática Aplicada  
Departamento de Informática  
Universidad de Almería  
ALMERÍA, 28 DE SEPTIEMBRE DE 2014

# Contenido

PRÓLOGO . . . . .	xix
<b>1. TELEDETECCIÓN . . . . .</b>	<b>29</b>
1.1. INTRODUCCIÓN A LA TELEDETECCIÓN . . . . .	31
1.1.1. Definición de teledetección . . . . .	32
1.1.2. Historia de la teledetección . . . . .	32
1.1.3. Elementos básicos de la teledetección . . . . .	34
1.1.4. Fundamentos físicos de la teledetección . . . . .	35
1.1.5. Aplicaciones de la teledetección . . . . .	37
1.2. CARACTERÍSTICAS DE LAS IMÁGENES DE SATÉLITE . . . . .	39
1.2.1. Imágenes de satélite . . . . .	39
1.2.2. Resolución de las imágenes de satélite . . . . .	42
1.2.3. Errores en la captación de imágenes . . . . .	43
1.3. SATÉLITES LANDSAT . . . . .	44
1.3.1. Características técnicas de los satélites <i>Landsat</i> . . . . .	44
1.3.2. Combinación 3,2,1 (color verdadero) . . . . .	46
1.3.3. Combinaciones 4,3,2 / 5,4,3 / 4,5,3 . . . . .	47
1.3.4. Otras combinaciones . . . . .	49
1.4. CLASIFICACIÓN DE IMÁGENES DE SATÉLITE . . . . .	49
1.4.1. Clasificación no supervisada y supervisada . . . . .	50
1.4.2. Clasificador <i>k-medias</i> . . . . .	53
1.4.3. Clasificador <i>isodata</i> . . . . .	53
1.4.4. Clasificador <i>paralelepípedos</i> . . . . .	55
1.4.5. Clasificador <i>mínima distancia</i> . . . . .	56
1.4.6. Clasificador <i>máxima verosimilitud</i> . . . . .	57
1.5. RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .	59
<b>2. AUTÓMATAS CELULARES . . . . .</b>	<b>63</b>
2.1. INTRODUCCIÓN A LOS AUTÓMATAS CELULARES . . . . .	65
2.1.1. Definición conceptual de autómatas celulares . . . . .	66
2.1.2. Definición formal de autómatas celulares . . . . .	67
2.1.3. Clasificación de los autómatas celulares . . . . .	68
2.2. APLICACIONES DE LOS AUTÓMATAS CELULARES . . . . .	68
2.2.1. Autómatas celulares unidimensionales. <i>Stephen Wolfram</i> . . . . .	69

2.2.2.	Autómatas celulares bidimensionales. <i>Juego de la vida</i> . . . . .	72
2.2.3.	Autómatas celulares aplicados a la teledetección . . . . .	75
2.3.	SIMULACIÓN DE PRECIPITACIONES MEDIANTE RACA . . . . .	76
2.3.1.	Modelado y simulación de precipitaciones . . . . .	76
2.3.2.	Objetivos propuestos . . . . .	78
2.3.3.	Algoritmo RACA . . . . .	78
2.3.4.	Características experimentales . . . . .	82
2.3.5.	Casos de estudio del algoritmo RACA . . . . .	83
2.4.	CARACTERIZACIÓN DE TEXTURAS MEDIANTE TACA . . . . .	87
2.4.1.	Números decimales de las reglas de <i>Stephen Wolfram</i> . . . . .	87
2.4.2.	Caracterización de texturas de imágenes-AC . . . . .	88
2.4.3.	Caracterización de texturas naturales. Algoritmo TACA . . . . .	90
2.4.4.	Casos de estudio del algoritmo TACA . . . . .	92
2.5.	RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .	94
<b>3.</b>	<b>CLASIFICACIÓN MEDIANTE AUTÓMATAS CELULARES</b> . . . . .	<b>99</b>
3.1.	PROBLEMAS DE LOS ALGORITMOS DE CLASIFICACIÓN . . . . .	101
3.1.1.	Limitaciones de <i>paralelepípedos</i> y <i>mínima distancia</i> . . . . .	102
3.1.2.	Análisis de la tasa de acierto: matriz de confusión . . . . .	104
3.1.3.	Análisis de errores durante el proceso de clasificación . . . . .	105
3.2.	CLASIFICACIÓN DE IMÁGENES DE SATÉLITE CON ACA . . . . .	109
3.2.1.	Trabajo previo existente . . . . .	109
3.2.2.	Objetivos principales del algoritmo ACA . . . . .	110
3.2.3.	Información complementaria de los objetivos principales . . . . .	111
3.2.4.	Consideraciones previas del algoritmo ACA . . . . .	112
3.3.	DESCRIPCIÓN DEL ALGORITMO ACA . . . . .	113
3.3.1.	Estados y reglas del autómata celular . . . . .	113
3.3.2.	Descripción matemática del autómata celular . . . . .	115
3.3.3.	Pseudocódigo del algoritmo ACA . . . . .	116
3.3.4.	Fórmulas complementarias del algoritmo ACA . . . . .	119
3.3.5.	Clasificación jerárquica del algoritmo ACA . . . . .	121
3.3.6.	Arquitectura del algoritmo ACA . . . . .	122
3.4.	HERRAMIENTAS UTILIZADAS PARA LA INVESTIGACIÓN . . . . .	124
3.4.1.	Erdas Imagine . . . . .	124
3.4.2.	Developers Toolkit . . . . .	126
3.4.3.	Weka . . . . .	128
3.4.4.	MATLAB . . . . .	129
3.5.	RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .	130
<b>4.</b>	<b>RESULTADOS OBTENIDOS</b> . . . . .	<b>133</b>
4.1.	RESULTADOS PRINCIPALES DE ACA . . . . .	135
4.1.1.	Características experimentales . . . . .	136
4.1.2.	Objetivo #1: mejorar la tasa de acierto de la clasificación . . . . .	142
4.1.3.	Objetivo #2: construir una clasificación jerárquica . . . . .	145
4.1.4.	Objetivo #3: detectar bordes y píxeles inciertos/ruidosos . . . . .	147

4.1.5. Conclusiones . . . . .	151
4.2. RESULTADOS COMPLEMENTARIOS DE ACA . . . . .	152
4.2.1. Características experimentales . . . . .	153
4.2.2. Información #1: evolución de las matrices de confusión . . . . .	154
4.2.3. Información #2: evolución de la precisión y el <i>recall</i> . . . . .	158
4.2.4. Información #3: evolución de las tablas de píxeles . . . . .	166
4.2.5. Conclusiones . . . . .	167
4.3. FORMATO DE SALIDA DEL ALGORITMO ACA . . . . .	168
4.3.1. Índice completo de los archivos de salida . . . . .	168
4.3.2. Clasificación jerárquica: número de iteraciones y color . . . . .	169
4.3.3. Matrices de confusión intermedias por iteraciones . . . . .	173
4.3.4. Datos de las gráficas evolutivas: acierto, precisión y <i>recall</i> . . . . .	174
4.4. RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .	175
<b>5. APORTACIONES Y LÍNEAS DE INVESTIGACIÓN ABIERTAS . . . . .</b>	<b>179</b>
5.1. APORTACIONES A LA COMUNIDAD CIENTÍFICA . . . . .	181
5.1.1. Clasificación de imágenes de satélite con ACA: objetivo#1 . . . . .	182
5.1.2. Clasificación de imágenes de satélite con ACA: objetivo#2 . . . . .	183
5.1.3. Clasificación de imágenes de satélite con ACA: objetivo#3 . . . . .	183
5.1.4. Simulación de precipitaciones con el algoritmo RACA . . . . .	184
5.1.5. Caracterización de texturas con el algoritmo TACA . . . . .	185
5.1.6. Correlación de variables ecológicas con <i>RBFs</i> . . . . .	185
5.2. PUBLICACIONES DERIVADAS . . . . .	186
5.2.1. Clasificación de imágenes de satélite: algoritmo ACA . . . . .	186
5.2.2. Autómatas celulares: algoritmos RACA y TACA . . . . .	187
5.2.3. Teledetección: correlación de variables ecológicas con <i>RBFs</i> . . . . .	187
5.3. LÍNEAS DE INVESTIGACIÓN ABIERTAS . . . . .	188
5.3.1. Clasificación de imágenes de satélite con ACA . . . . .	188
5.3.2. Simulación de precipitaciones con el algoritmo RACA . . . . .	189
5.3.3. Caracterización de texturas con el algoritmo TACA . . . . .	189
<b>A. TRABAJO DE CAMPO . . . . .</b>	<b>195</b>
A.1. IMÁGENES <i>LANDSAT</i> . . . . .	195
A.2. IMAGEN <i>DEM</i> . . . . .	200
<b>B. PUBLICACIONES ADICIONALES . . . . .</b>	<b>205</b>
B.1. PROYECTOS I+D . . . . .	205
B.2. PUBLICACIONES . . . . .	206
<b>GLOSARIO DE TÉRMINOS . . . . .</b>	<b>211</b>
<b>ACRÓNIMOS . . . . .</b>	<b>225</b>
<b>BIBLIOGRAFÍA . . . . .</b>	<b>229</b>





# Índice de Figuras

1.1. Elementos básicos de un sistema de teledetección (Fuente: <i>E. Chuvieco</i> ). . .	35
1.2. Espectro electromagnético. . . . .	36
1.3. Evolución del mar de Aral en Asia central (Fuente: <i>Nature</i> ). . . . .	37
1.4. (a) Olas de gravedad atmosférica en Nueva Zelanda. (b) Nieve en Gran Breña e Irlanda. (c) Tormentas de polvo en el suroeste de Asia. (d) Floración de fitoplancton en Argentina (Fuente: <i>NASA</i> ). . . . .	38
1.5. (a) Imagen multiespectral. (b) Imagen hiperespectral. . . . .	39
1.6. Representación 3D de una imagen <i>DEM</i> . . . . .	40
1.7. Imagen de satélite en el espacio de características (con 3 clases). . . . .	40
1.8. (a) Imagen <i>Landsat</i> de las provincias Almería-Granada (bandas 2, 3 y 4). (b) Imagen <i>Landsat</i> de Almería-Granada (bandas 1, 2 y 3: color verdadero). . . . .	41
1.9. Esquema general del satélite <i>Landsat 7</i> (Fuente: <i>NASA</i> ). . . . .	44
1.10. Combinación 3,2,1 de <i>El Ejido</i> , Almería. . . . .	46
1.11. Combinación 4,3,2 de <i>El Ejido</i> , Almería. . . . .	47
1.12. Combinación 5,4,3 de <i>El Ejido</i> , Almería. . . . .	48
1.13. Combinación 4,5,3 de <i>El Ejido</i> , Almería. . . . .	49
1.14. Esquema general del proceso de clasificación de imágenes de satélite. . . . .	50
1.15. (Izq.) Clasificación no supervisada. (Dcha.) Clasificación supervisada. . . . .	51
1.16. (a) Descripción del proceso general de clasificación. (b) Resultados de una clasificación no supervisada. (c) Resultados de una clasificación supervisada. . . . .	52
1.17. Clasificación mediante el algoritmo <i>paralelepípedos</i> . . . . .	56
1.18. Clasificación mediante el algoritmo <i>mínima distancia</i> . . . . .	57
1.19. Clasificación mediante el algoritmo <i>máxima verosimilitud</i> . . . . .	58
2.1. Tipos de vecindad: (a) <i>von Neumann</i> . (b) <i>Moore</i> . (c) <i>Moore extendida</i> . . . . .	66
2.2. Imagen-AC generada por la regla <i>R94</i> . . . . .	71
2.3. Imagen-AC generada por la regla <i>R54</i> . . . . .	71
2.4. Reglas que generan las imágenes-AC más conocidas. . . . .	72
2.5. Imágenes-AC más conocidas de <i>Stephen Wolfram</i> . . . . .	73

2.6. (a) <i>Soledad</i> : una celda viva con menos de 2 celdas vivas adyacentes muere. (b) <i>Sobrepoblación</i> : una celda viva con más de 3 celdas vivas adyacentes también muere. (c) <i>Reproducción</i> : una celda muerta con 3 celdas vivas adyacentes nace. (d) <i>Estabilidad</i> : una celda viva con dos celdas vivas adyacentes permanece viva. . . . .	74
2.7. (a) Condición inicial. (b) Iteración 10. (c) Iteración 25. (d) Iteración 50. (e) Iteración 75. . . . .	75
2.8. Arquitectura del algoritmo RACA. . . . .	81
2.9. Imagen <i>DEM S32W070</i> (misión <i>SRTM</i> de la <i>NASA</i> ): (a) Vista en planta. (b) Vista en perspectiva. . . . .	82
2.10. Gráficas de precipitaciones anuales de los 4 tipos de climas: (a) Clima árido de sierras y bolsones. (b) Clima frío húmedo. (c) Clima templado pampeano húmedo. (d) Clima subtropical sin estación seca. . . . .	83
2.11. Resultados gráficos 3D anuales obtenidos en los 4 tipos de climas: (a) Clima árido de sierras y bolsones. (b) Clima frío húmedo. (c) Clima templado pampeano húmedo. (d) Clima subtropical sin estación seca. . . . .	85
2.12. Resultados del clima subtropical sin estación seca. . . . .	86
2.13. (a) Imagen-AC obtenida mediante la regla <i>R225</i> de <i>Stephen Wolfram</i> . (b) Histograma de números decimales obtenidos aplicando el método propuesto. . . . .	89
2.14. Primitivas de textura correspondientes a la imagen-AC generada por la regla <i>R102</i> de <i>Stephen Wolfram</i> . . . . .	89
2.15. (Izquierda) Imagen (256×256) compuesta por 4 texturas distintas: sectores superiores ( <i>R121</i> y <i>R153</i> ), sectores inferiores ( <i>R60</i> y <i>R90</i> ). (Derecha) Imagen en escala de grises resultante de la aplicación del método propuesto a la imagen original. . . . .	89
2.16. (Arriba) Tres texturas naturales del <i>Álbum Brodatz</i> : <i>D3</i> , <i>D49</i> y <i>D85</i> . (Abajo) Imágenes binarias correspondientes. . . . .	91
2.17. Histogramas de primitivas normalizados para la imagen <i>Brodatz D85</i> : (a) <i>HPH</i> (dirección horizontal). (b) <i>HPV</i> (dirección vertical). . . . .	92
2.18. Imágenes-AC obtenidas como resultado de elegir las primitivas más importantes de los histogramas anteriores. Del histograma vertical se han tomado 6 primitivas y del horizontal 3. Aplicando el método propuesto se han obtenido las siguientes reglas: (a) <i>R207</i> (horizontal). (b) <i>R193</i> (vertical). . . . .	92
2.19. (Izquierda) Imagen <i>bin3</i> (256×256). (Derecha) Aproximación de la imagen <i>bin3</i> obtenida por la composición de las imágenes-AC <i>R95</i> (H) y <i>R95</i> (V), obtenidas usando 6 componentes de cada histograma primitivo (H y V). . . . .	93
2.20. (Izquierda) Imagen <i>bin49</i> (256×256). (Derecha) Aproximación de la imagen <i>bin49</i> obtenida por la composición de las imágenes-AC <i>R217</i> (H) y <i>R76</i> (V), obtenidas usando 5 componentes del <i>HPH</i> y 3 componentes del <i>HPV</i> . . . . .	93

2.21. (Izquierda) Imagen *bin85* (256×256). (Derecha) Aproximación de la imagen *bin85* obtenida por la composición de las imágenes-AC *R207* (H) y *R193* (V), obtenidas usando 6 componentes del *HPH* y 3 componentes del *HPV*. . . . . 94

3.1. Limitaciones del algoritmo *paralelepípedos*. . . . . 103

3.2. Limitaciones del algoritmo *mínima distancia*. . . . . 103

3.3. Zona de estudio. . . . . 104

3.4. (a) Clasificación realizada por *trabajo de campo* centrada en los dos píxeles. (b) Clasificación obtenida por *mínima distancia* centrada en los dos píxeles de estudio. En ambas imágenes, los píxeles de estudio ocupan la posición [003, 161] y [003, 162]. . . . . 105

3.5. Valor espectral de un píxel ruidoso. . . . . 108

3.6. Evolución del algoritmo ACA mediante 3 iteraciones del autómata celular: (Línea 1) *ACA paralelepípedos*. El *threshold* aumenta en proporción a la covarianza. (Línea 2) *ACA mínima distancia*. El *threshold* aumenta en igual proporción. . . . . 121

3.7. Arquitectura del algoritmo ACA. . . . . 122

3.8. Interfaz gráfica de *Erdas Imagine 9.2*. Imagen de satélite: *El Ejido* (1994). 125

3.9. Código fuente del algoritmo ACA en *Visual Studio*. . . . . 127

3.10. Panel de clasificación de *Weka*. Algoritmo *J48 (C4.5)*. . . . . 128

3.11. Interfaz gráfica de *MATLAB*. Imagen de satélite *DEM*: vista en planta. . 129

4.1. (a) Regiones de estudio de la provincia de Almería, España (abril de 2003). (b) *Níjar*. (c) *El Ejido Oeste*. (d) *El Ejido Este*. . . . . 136

4.2. Sectores de vegetación del sureste de España. *Níjar* (cuadrado rojo) y *El Ejido* (cuadrados azules). . . . . 137

4.3. Gráficas espectrales de la imagen de satélite de *El Ejido Oeste*: (a) Bandas 2-4. (b) Bandas 3-4. (c) Bandas 2-7. (d) Bandas 3-7. (e) Bandas 4-7. (f) Bandas 6-7. . . . . 140

4.4. Evolución de la tasa de acierto de los algoritmos de clasificación: (a) *Níjar*. (b) *El Ejido Oeste*. (c) *El Ejido Este*. . . . . 144

4.5. Iteraciones 10, 20, 30, 40 y 100 de la clasificación jerárquica del algoritmo *ACA mínima distancia*: (a) *Níjar*. (b) *El Ejido Oeste*. (c) *El Ejido Este*. . 146

4.6. (a) Área de estudio de Almería, España (abril de 2003). (b) *El Ejido Oeste*. 153

4.7. Evolución de la tasa de acierto de los algoritmos *ACA paralelepípedos* y *ACA mínima distancia* a lo largo de las 100 iteraciones de la clasificación. 158

4.8. Iteraciones 1-12 de la clasificación *ACA mínima distancia*, *El Ejido Oeste*. 159

4.9. Iteraciones 13-24 de la clasificación *ACA mínima distancia*, *El Ejido Oeste*. 160

4.10. Iteraciones 25-100 de la clasificación *ACA mínima distancia*, *El Ejido Oeste*. 161

4.11. Precisión de los algoritmos de la comparativa, *El Ejido Oeste*. . . . . 162

4.12. *Recall* de los algoritmos de la comparativa, *El Ejido Oeste*. . . . . 163

4.13. Evolución de la precisión de las clases con *ACA paralelepípedos*. . . . . 164

4.14. Evolución del *recall* de las clases con *ACA paralelepípedos*. . . . . 164

4.15. Evolución de la precisión de las clases con *ACA mínima distancia*. . . . . 165

4.16. Evolución del <i>recall</i> de las clases con <i>ACA mínima distancia</i> . . . . .	165
4.17. Zona de estudio tomada de la imagen de satélite original de las provincias de Almería y Granada (España). . . . .	169
4.18. Clasificación jerárquica dividida en 6 capas de fiabilidad. . . . .	170
4.19. Iteraciones 1-12 de la clasificación <i>ACA paralelepípedos</i> . . . . .	171
4.20. Iteraciones 13-30 de la clasificación <i>ACA paralelepípedos</i> . . . . .	172
4.21. Archivo de datos de los resultados de las matrices de confusión intermedias, proporcionadas por el algoritmo <i>ACA paralelepípedos</i> para la imagen de <i>El Ejido Oeste</i> . . . . .	173
4.22. Archivo de datos de los resultados numéricos de las gráficas de evolución, proporcionadas por el algoritmo <i>ACA paralelepípedos</i> para la imagen de <i>El Ejido Oeste</i> . . . . .	174
A.1. Sectores climáticos. <i>Níjar</i> (cuadrado rojo) y <i>El Ejido</i> (cuadrados azules). . . . .	195
A.2. Sectores edáficos. <i>Níjar</i> (cuadrado rojo) y <i>El Ejido</i> (cuadrados azules). . . . .	196
A.3. Sectores litológicos. <i>Níjar</i> (cuadrado rojo) y <i>El Ejido</i> (cuadrados azules). . . . .	197
A.4. Sectores de vegetación. <i>Níjar</i> (cuadrado rojo) y <i>El Ejido</i> (cuadrados azules). . . . .	198
A.5. Sectores geomorfológicos. <i>Níjar</i> (cuadrado rojo) y <i>El Ejido</i> (azules). . . . .	199
A.6. Zona de estudio de la imagen <i>S32W070</i> (Fuente: <i>Google Maps</i> ). . . . .	200
A.7. Perfil topográfico con la zona de estudio de la imagen <i>S32W070</i> en azul (Fuente: <i>Atlas Geográfico de la República de Argentina - Instituto Geográfico Militar</i> ). . . . .	201

# Índice de Tablas

1.1. Período de actividad de los satélites <i>Landsat</i> (Fuente: <i>NASA</i> ). . . . .	44
1.2. Características técnicas del sistema <i>ETM+</i> (Fuente: <i>NASA</i> ). . . . .	45
1.3. Combinación de bandas de <i>Landsat 7</i> . . . . .	49
1.4. Algoritmo de clasificación <i>k-medias</i> . . . . .	54
1.5. Algoritmo de clasificación <i>isodata</i> . . . . .	55
2.1. Evolución del autómata celular unidimensional del ejemplo. . . . .	69
2.2. Descripción del algoritmo RACA. . . . .	81
2.3. Resultados numéricos anuales obtenidos en los 4 tipos de climas. . . . .	85
2.4. Adquisición de los números decimales de la regla de 3 celdas. . . . .	88
2.5. Descripción del algoritmo TACA. . . . .	91
3.1. Ventajas y limitaciones de los algoritmos <i>paralelepípedos</i> y <i>mínima distancia</i> . . . . .	102
3.2. Lista de 200 píxeles mal clasificados por parte del algoritmo <i>mínima distancia</i> clásico, tomados al azar. En negrita, los píxeles que poseen la relación de clases “5, 7”. . . . .	106
3.3. Valores espectrales de los dos píxeles de estudio. . . . .	107
3.4. Valores espectrales del conjunto de entrenamiento. . . . .	107
3.5. Cálculo de la distancia del píxel de estudio a la clase 5. . . . .	107
3.6. Cálculo de la distancia del píxel de estudio a la clase 7. . . . .	108
3.7. Objetivos principales e información complementaria del algoritmo ACA. . . . .	111
3.8. Pseudocódigo del algoritmo <i>ACA principal</i> . . . . .	116
3.9. Pseudocódigo del algoritmo <i>ACA espectral</i> . . . . .	117
3.10. Pseudocódigo del algoritmo <i>ACA contextual</i> . . . . .	118
4.1. Variables ecológicas de vegetación de los 6 sectores principales. . . . .	138
4.2. Características principales de las 6 clases de <i>Níjar</i> y <i>El Ejido</i> . . . . .	139
4.3. Características de <i>Níjar</i> , <i>El Ejido Oeste</i> y <i>El Ejido Este</i> . . . . .	141
4.4. Tasa de acierto de los algoritmos de clasificación. . . . .	143
4.5. Coste computacional de los algoritmos de clasificación (donde: $n$ es el número de instancias de entrenamiento, $m$ es el número de atributos e $i$ es el número de iteraciones del autómata celular). . . . .	144
4.6. Píxeles ruidosos bien clasificados de <i>El Ejido Oeste</i> y <i>Este</i> . . . . .	147

4.7. Píxeles inciertos bien clasificados de <i>Níjar</i> . . . . .	148
4.8. Píxeles inciertos bien clasificados de <i>El Ejido Oeste</i> . . . . .	148
4.9. Píxeles inciertos bien clasificados de <i>El Ejido Este</i> . . . . .	148
4.10. Píxeles borde y foco de <i>Níjar</i> . . . . .	150
4.11. Píxeles borde y foco de <i>El Ejido Oeste</i> . . . . .	150
4.12. Píxeles borde y foco de <i>El Ejido Este</i> . . . . .	150
4.13. Características principales de las 6 clases de <i>El Ejido Oeste</i> . . . . .	153
4.14. Matriz de confusión del algoritmo <i>paralelepípedos</i> clásico. . . . .	154
4.15. Matriz de confusión del algoritmo <i>ACA paralelepípedos</i> (iteración 2). . . . .	155
4.16. Matriz de confusión del algoritmo <i>ACA paralelepípedos</i> (iteración 40). . . . .	155
4.17. Matriz de confusión del algoritmo <i>ACA paralelepípedos</i> (iteración 100). . . . .	156
4.18. Matriz de confusión del algoritmo mínima distancia clásico. . . . .	156
4.19. Matriz de confusión del algoritmo <i>ACA mínima distancia</i> (iteración 3). . . . .	157
4.20. Matriz de confusión del algoritmo <i>ACA mínima distancia</i> (iteración 44). . . . .	157
4.21. Matriz de confusión del algoritmo <i>ACA mínima distancia</i> (iteración 100). . . . .	157
4.22. Píxeles ruidosos, inciertos, bordes, focos y erróneos. <i>ACA paralelepípedos</i> . . . . .	166
4.23. Píxeles ruidosos, inciertos, bordes, focos y erróneos. <i>ACA mínima distancia</i> . . . . .	166
5.1. Correspondencia entre las publicaciones y las aportaciones. . . . .	187

# PRÓLOGO

**L**a **teledetección** es una ciencia aplicada que permite la adquisición de información mediante técnicas que no requieren contacto físico con el objeto o área que está siendo observada. La teledetección ha sido utilizada en incontables aplicaciones medioambientales con el objetivo de resolver y optimizar problemas de índole muy variada, como por ejemplo los estudios de calidad del suelo, la búsqueda de recursos hídricos, las simulaciones meteorológicas o la protección ambiental, entre otros.

Para solventar todos estos problemas, uno de los desafíos más complicados a los que se enfrenta la teledetección es el procesamiento de las enormes cantidades de datos recolectados por los satélites. Existen muchas técnicas que se utilizan en el ámbito de la teledetección con el objetivo de ayudar a los analistas expertos a interpretar los datos generados y ofrecidos por los satélites. Algunas de las técnicas más importantes son los *algoritmos de clasificación* de imágenes de satélite, cuya finalidad es agrupar todos los píxeles de una imagen digital en un número finito de clases, facilitando de este modo la interpretación de la enorme cantidad de datos contenidos en las bandas espectrales captadas por los sensores de los satélites. Cuando se aplica un algoritmo de clasificación a una imagen de satélite, los datos obtenidos por los sensores, almacenados como niveles digitales en forma de bandas espectrales, se transforman en una escala categórica fácil de interpretar por parte de los analistas expertos. La imagen clasificada que se obtiene como resultado es un mapa temático de la imagen de satélite original, donde los píxeles pertenecientes a la misma clase tienen unas características espectrales similares. Los algoritmos de clasificación de imágenes de satélite se pueden dividir en dos categorías principales: no supervisados y supervisados, y el uso de unos u otros depende del conocimiento previo que tengan los analistas expertos de la zona de estudio.

Los algoritmos de clasificación de imágenes de satélite *no supervisados* trabajan sin el conocimiento previo del área de estudio. En este caso, el analista experto sólo tiene que especificar el número total de clases deseadas, y el algoritmo no supervisado agrupa los píxeles de la imagen en dicho número de clases, basándose únicamente en los datos espectrales de las bandas. La tasa de acierto de la clasificación obtenida por estos métodos es bastante menos fiable que los resultados ofrecidos por los algoritmos de clasificación supervisados, ya que en los algoritmos no supervisados la computadora agrupa los píxeles de la imagen en clases sin tener en cuenta el conocimiento que tiene el analista experto del área de estudio. Existen muchos algoritmos de clasificación no supervisados, como *k-medias*, *isodata*, *maximin* y el *modelo neural*.

Los algoritmos de clasificación de imágenes de satélite *supervisados* trabajan con el conocimiento previo del área de estudio, por lo que pueden hacer una clasificación mucho más precisa que los algoritmos de clasificación no supervisados, basándose en el conocimiento experto humano. En una fase anterior al proceso de clasificación, el analista experto selecciona, de la propia imagen de satélite, varias muestras de las diferentes clases. Dichas muestras están formadas por píxeles representativos que componen el denominado conjunto de entrenamiento (o firmas), sobre el que posteriormente se basa el proceso de clasificación supervisado. Por lo tanto, en este método, el conocimiento del analista experto sobre el área de estudio determina la calidad, tanto del conjunto de entrenamiento como de la tasa de acierto obtenida por parte del proceso de clasificación. Los algoritmos de clasificación supervisados comparan cada píxel de la imagen con estas firmas elegidas por el analista experto y, a continuación, cada píxel es etiquetado en la clase a la que más se asemeja espectralmente. Los algoritmos de clasificación supervisados, en general, ofrecen mejores resultados que los algoritmos de clasificación no supervisados. Existen muchos algoritmos de clasificación de imágenes de satélite supervisados, como *paralelepípedos*, *mínima distancia*, *máxima verosimilitud*, *distancia de Mahalanobis* y *algoritmos difusos supervisados*.

Sin embargo, a pesar del gran número de algoritmos de clasificación de imágenes de satélite que existe en la actualidad, todos presentan limitaciones que provocan que ninguno de ellos sea completamente fiable en términos de acierto. Estas limitaciones se incrementan cuando existen en la imagen de satélite determinadas clases con un alto *grado de heterogeneidad*, es decir, clases cuyos píxeles presentan un abanico espectral bastante amplio y por lo tanto se puede dudar de la clase a la que pertenecen (denominados a lo largo de este documento como *píxeles inciertos*). Además, las imágenes de satélite pueden estar alteradas con píxeles de ruido de tipo *impulsivo-gaussiano* (denominados a lo largo de este documento como *píxeles ruidosos*), provocando que la imagen tenga muchas zonas diminutas (a menudo de un solo píxel) que han sido mal etiquetadas durante el proceso de clasificación.

Por otro lado, un **autómata celular** es un modelo matemático que consiste en un conjunto de celdas, normalmente distribuidas en forma de matriz bidimensional, donde cada celda posee un estado, que puede cambiar a lo largo de un número determinado de iteraciones, como consecuencia de la aplicación de un conjunto de reglas a través de una función de transición  $f$ , teniendo en cuenta no sólo el estado actual de la celda en cada iteración concreta, sino también el estado de sus celdas vecinas. En los últimos años, los autómatas celulares se han convertido en una potente herramienta que se ha aplicado a numerosos ámbitos científicos. Dentro del ámbito de la teledetección, se han aplicado sobre todo para implementar procesos de simulación en imágenes de satélite, ya que sus propiedades facilitan trabajar con este tipo de problemas.

No obstante, en el ámbito específico de la *clasificación de imágenes de satélite*, son escasos los trabajos relacionados con los autómatas celulares, a pesar de las ventajas que pueden llegar a ofrecer, como por ejemplo la mejora de la tasa de acierto del proceso de clasificación mediante el uso de técnicas contextuales a través de la vecindad de las celdas, la obtención de una clasificación jerárquica basada en el grado de pertenencia a las clases a través del uso de iteraciones, o la personalización del proceso de clasificación utilizando diferentes reglas dependiendo de los resultados que se deseen obtener.



Esta tesis doctoral se centra en mejorar, mediante el uso de los autómatas celulares, las limitaciones que presentan la inmensa mayoría de los algoritmos de clasificación clásicos. Para conseguir este objetivo, el trabajo desarrollado se ha centrado en la implementación de una nueva aplicación de los autómatas celulares: un algoritmo de clasificación supervisado basado en autómatas celulares, que no sólo mejora la tasa de acierto de los resultados obtenidos a través de la clasificación clásica, sino que también ofrece información sobre el grado de pertenencia de cada píxel a su clase correspondiente a través de una clasificación jerárquica dividida en niveles de fiabilidad, y además permite personalizar el proceso de clasificación de la zona de estudio concreta.

## JUSTIFICACIÓN

Los resultados proporcionados por los algoritmos de clasificación de imágenes de satélite tienen muchas aplicaciones medioambientales, militares, sociales y políticas. Estos resultados son los principales elementos que utilizan muchos *sistemas de información geográficos (SIG)* para ofrecer al usuario la información requerida, como por ejemplo el cálculo del crecimiento del suelo urbano en las ciudades en un intervalo de tiempo determinado, la monitorización de la calidad ambiental después de desastres naturales, la creación de mapas *GPS* de manera automatizada, la actualización de la superficie cartográfica de una zona concreta, la prevención de desastres naturales (como la propagación de incendios o las avalanchas de nieve), la evaluación de los riesgos de los recursos naturales o el estudio de la evolución del cambio climático en las distintas zonas del planeta. Por lo tanto, ya que los algoritmos de clasificación de imágenes de satélite son la base de muchos *SIG*, es importante proporcionar unos resultados tan optimizados como sea posible. En la actualidad existe una gran cantidad de algoritmos de clasificación, y el uso de uno u otro en particular depende del conocimiento que posea el analista experto sobre la zona de estudio. Sin embargo, los algoritmos clásicos presentan tres *limitaciones* importantes: la baja tasa de acierto obtenida en imágenes de satélite complejas, la rigidez de los resultados y la falta de personalización del proceso de clasificación.

En primer lugar, a pesar del gran número de algoritmos de clasificación de imágenes de satélite que existe en la actualidad, ninguno es completamente fiable en términos de **tasa de acierto**. En general, los algoritmos de clasificación funcionan de un modo correcto si las propiedades espectrales de los píxeles determinan correctamente las clases, o si las imágenes no presentan ruido adicional. Sin embargo, si existen clases en la imagen de satélite con un alto *grado de heterogeneidad*, es decir, que agrupan píxeles con diferentes características, y que presentan un amplio abanico espectral (*píxeles inciertos*) o las imágenes son alteradas con un ruido de tipo *impulsivo-gaussiano* (*píxeles ruidosos*), la imagen resultante puede presentar muchas áreas diminutas mal clasificadas. Todos estos problemas provocan una pérdida de la calidad de la tasa de acierto durante el proceso de clasificación. Para solucionar estos problemas, podemos aplicar posteriormente algoritmos contextuales de post-clasificación, que utilizan datos contextuales además de los datos espectrales. Existen muchos algoritmos contextuales de post-clasificación, que utilizan valores promedio o la descripción de texturas, para mejorar los resultados obtenidos por los algoritmos de clasificación espectral.

Sin embargo, el inconveniente de su uso consiste en que es necesario aplicar tres algoritmos a lo largo de todo el proceso: un algoritmo de *pre-clasificación* para eliminar los píxeles ruidosos, el algoritmo de clasificación propiamente dicho y un algoritmo de *post-clasificación* para mejorar la clasificación de los píxeles inciertos. A veces, como consecuencia de la aplicación de todos estos algoritmos, puede incluso empeorar la calidad del acierto. Por lo tanto, sería interesante disponer de un solo algoritmo que contara con las bondades de los tres procesos anteriormente mencionados, para incrementar la calidad de la tasa de acierto mediante una clasificación espectral-contextual.

En segundo lugar, los algoritmos de clasificación de imágenes de satélite clásicos ofrecen **resultados demasiado rígidos**, ya que cada píxel es etiquetado en su clase correspondiente independientemente del nivel real de cercanía respecto al centro espectral de dicha clase, es decir, no se ofrece un grado de pertenencia. Sólo los algoritmos de *clasificación difusa* proporcionan dicha información, pero no los clásicos. Sería interesante que un algoritmo ofreciera una *clasificación jerárquica* dividida en niveles de pertenencia a las clases, para que los analistas expertos pudieran comprobar qué píxeles están más cerca de sus clases y cuáles están más distantes en el espacio de características, con el objetivo de detectar los píxeles más problemáticos.

En tercer lugar, los algoritmos de clasificación clásicos **no permiten personalizar** el funcionamiento para ajustarse a una *zona de estudio concreta*, o para obtener unos resultados específicos. Sería interesante poder personalizar el proceso de clasificación, para obtener unos u otros resultados dependiendo del objetivo que se quisiera conseguir en cada zona de estudio determinada.

Estas tres limitaciones se pueden resolver mediante las propiedades de los autómatas celulares. La primera limitación se puede superar mediante el uso de la *vecindad* de cada celda del autómata celular durante el proceso de clasificación, obteniendo una pre-clasificación (para píxeles ruidosos), clasificación y post-clasificación (para píxeles inciertos) agrupados en un solo algoritmo espectral-contextual, pudiendo ajustar además el nivel contextual, usando más o menos vecinos para las celdas. La segunda limitación se puede superar mediante el uso de las *iteraciones* del autómata celular, como método de división del proceso de clasificación en varios niveles jerárquicos con distinto grado de calidad. La tercera limitación se puede superar mediante el empleo de distintas *reglas* aplicadas, a través de la función de transición  $f$ , para poder personalizar lo máximo posible el proceso de clasificación en cada caso concreto. De este modo, se pueden solventar todas las limitaciones propuestas, mediante los autómatas celulares.

Esta tesis doctoral se centra en el desarrollo de un nuevo algoritmo de clasificación de imágenes de satélite basado en autómatas celulares, denominado en este documento **algoritmo ACA** (*classification Algorithm based on Cellular Automata*), que resuelve las tres limitaciones descritas, proporcionando una clasificación con una tasa de acierto mejorada mediante resultados espectrales-contextuales, divididos en niveles de fiabilidad relacionados con el grado de pertenencia de los píxeles a sus clases correspondientes, y con la posibilidad de personalizar el proceso de clasificación. Todas estas mejoras permiten a los analistas expertos disponer de una mayor cantidad de información para mejorar la interpretación de los resultados obtenidos. De este modo, el algoritmo ACA optimiza la funcionalidad de cualquier *SIG* que lo utilice como base, ya que mejora los resultados de la clasificación en tres niveles: acierto, flexibilidad y personalización.

## OBJETIVOS

El objetivo general de esta tesis doctoral consiste en la implementación de un nuevo algoritmo de clasificación de imágenes de satélite supervisado, basado en autómatas celulares (algoritmo ACA), que optimice la tasa de acierto en la clasificación de los píxeles de una imagen basándose tanto en los datos espectrales como en los contextuales de cada píxel, proporcione una clasificación jerárquica de los resultados divididos en niveles de grado de pertenencia a las clases y permita personalizar el proceso de clasificación a la zona de estudio concreta. Los principales objetivos propuestos durante el desarrollo de esta tesis doctoral son los siguientes:

- **Objetivo #1.** Mejorar la *tasa de acierto* obtenida por los algoritmos de clasificación clásicos, utilizando para ello no sólo información espectral sino también información contextual, con el objetivo de no clasificar erróneamente los píxeles inciertos y ruidosos. El algoritmo ACA debe clasificar los píxeles problemáticos teniendo en cuenta no sólo sus datos espectrales (ambiguos para los píxeles inciertos, erróneos para los píxeles ruidosos), sino también sus datos contextuales vecinos, para que mejore la calidad de la tasa de acierto final. El algoritmo ACA debe combinar las tres técnicas en un solo algoritmo: el proceso de pre-clasificación (eliminación de píxeles ruidosos), el proceso de clasificación supervisado y el proceso de post-clasificación (refinamiento de píxeles inciertos).
- **Objetivo #2.** Obtener una *clasificación jerárquica* dividida en niveles o capas que indiquen el grado de pertenencia a las clases. El algoritmo ACA debe clasificar, en cada iteración, sólo aquellos píxeles que estén dentro de una distancia espectral máxima en el espacio de características con respecto al centro de sus clases correspondientes, y tal distancia debe aumentar en la siguiente iteración. Así, el algoritmo ACA ofrecerá una clasificación jerárquica dividida en niveles o capas de fiabilidad, donde las primeras capas ofrecerán más fiabilidad que las últimas. Estos resultados podrán ser muy útiles para la posterior interpretación de los resultados por parte de los analistas expertos. Por otra parte, este objetivo ayudará al primero, ya que los píxeles inciertos y ruidosos, por lo general más lejos del centro de sus clases, deberán ser clasificados en las últimas iteraciones del autómata celular y usar, como vecinos, otros píxeles que serán muy propensos a ser clasificados en iteraciones anteriores, por lo que optimizarán la tasa de acierto final.
- **Objetivo #3.** *Personalizar* el proceso de clasificación de la imagen de satélite mediante la inclusión de reglas, para poder obtener, por ejemplo, una lista detallada de los píxeles inciertos y ruidosos, lo cual podrá ser muy útil si el proceso de clasificación falla incluso cuando se utilizan técnicas contextuales, y obtener una lista de los píxeles que determinen los bordes espaciales de las clases de la imagen. Esta segunda opción será muy interesante, ya que la detección de bordes espaciales (no espectrales) también se trata de un problema bastante común en el ámbito de la teledetección. Gracias a esta información, podremos ser capaces, por ejemplo, de establecer cuántos grupos de píxeles de la misma clase se encuentran en una imagen de satélite concreta.

## CONTEXTO CIENTÍFICO

El algoritmo ACA ha sido experimentado y validado en el marco de los proyectos I+D *SOLERES* [83]: *un sistema de información espacio-temporal para la gestión medioambiental basado en redes neuronales, agentes y componentes software* (TIN2007-61497) e *iSOLERES* [85]: *una metodología para la recuperación y explotación de información medioambiental mediante interfaces de usuario evolutivas y cooperativas* (TIN2010-15588), subvencionados por la UE (FEDER) y por el Ministerio de Ciencia e Innovación de España, en la actualidad Ministerio de Economía y Competitividad (MINECO).

El principal objetivo del proyecto *SOLERES* consistía en desarrollar investigación aplicada al diseño e implementación básica de un sistema de información espacio-temporal para la gestión medioambiental, soportado por la integración de innovación tecnológica en imágenes de satélite, redes neuronales, sistemas cooperativos basados en arquitecturas multiagente y agentes inteligentes, y sistemas software basados en componentes comerciales. En este proyecto se pretendía, por un lado, estudiar la *generación automática de mapas ecológicos* a partir de información de satélite. Para ello, se realizó un trabajo de investigación para correlacionar las variables ecológicas de vegetación con imágenes de satélite *Landsat TM* a través de redes de base radial (*Radial Basis Function nets, RBFs*), con el objetivo de optimizar el coste en tiempo requerido para realizar cualquier trabajo de campo. La obtención automática de estos mapas es de gran utilidad en actuaciones críticas de gestión medioambiental, como en tareas de predicción/prevención y de toma de decisiones, así como en fenómenos naturales de riesgo (p.e., inundaciones, incendios forestales, terremotos). También se crearon propuestas científico-técnicas reales y prácticas para la construcción de eficientes *sistemas de explotación de información* (en este caso medioambiental), facilitando la interacción hombre-máquina con interfaces de usuario dinámicas que se adaptaran a los hábitos de los perfiles de los usuarios. Además, se realizaron varios trabajos de investigación para optimizar los procesos de *clasificación y simulación* de imágenes de satélite mediante autómatas celulares, con el objetivo de mejorar el funcionamiento de los *SIG*. Esta tesis doctoral se centra en estos últimos trabajos de investigación, en los que se combina la teledetección con los autómatas celulares.

## ESTRUCTURA DE LA TESIS DOCTORAL

El presente documento está organizado en cinco capítulos principales. El *Capítulo 1* ofrece una introducción teórica al estado del arte de la teledetección, centrándose en las características principales de las imágenes de los satélites *Landsat*, así como en los algoritmos de clasificación no supervisados y supervisados. El *Capítulo 2* realiza una introducción teórica al modelo matemático del autómata celular, y describe el funcionamiento de dos algoritmos secundarios que se han implementado durante el desarrollo de esta tesis doctoral: el algoritmo RACA (*Rainfall Algorithm with Cellular Automata*), para la simulación de precipitaciones sobre imágenes de satélite *DEM*, y el algoritmo TACA (*Texture Algorithm with Cellular Automata*), para la caracterización de texturas naturales, ambos basados en los autómatas celulares.

El *Capítulo 3* describe detalladamente el funcionamiento del algoritmo ACA, para la clasificación de imágenes de satélite mediante autómatas celulares, que es el objetivo principal de esta tesis doctoral. El *Capítulo 4* muestra los resultados principales y complementarios obtenidos por parte del algoritmo ACA, así como el formato de salida de dichos resultados. El *Capítulo 5* ofrece una lista de las aportaciones, publicaciones derivadas y líneas de investigación abiertas de esta tesis doctoral.

Además, el documento contiene dos apéndices. El *Apéndice A* realiza una descripción del trabajo de campo e información recopilada de las distintas imágenes de satélite *Landsat* y *DEM*, utilizadas durante la fase de experimentación de los algoritmos ACA y RACA. El *Apéndice B* ofrece información sobre las publicaciones adicionales del autor del presente documento, a través de una lista de proyectos I+D y publicaciones complementarias a las desarrolladas para esta tesis doctoral.

Para finalizar, el documento también ofrece un *Glosario de términos y Acrónimos*, tanto universales como propios de esta tesis doctoral, así como la *Bibliografía* consultada para desarrollar el presente trabajo de investigación.



---

# CAPÍTULO 1

## TELEDETECCIÓN

---





# Capítulo 1

## TELEDETECCIÓN

### Contenidos

---

<b>1.1.</b>	<b>INTRODUCCIÓN A LA TELEDETECCIÓN . . . . .</b>	<b>31</b>
1.1.1.	Definición de teledetección . . . . .	32
1.1.2.	Historia de la teledetección . . . . .	32
1.1.3.	Elementos básicos de la teledetección . . . . .	34
1.1.4.	Fundamentos físicos de la teledetección . . . . .	35
1.1.5.	Aplicaciones de la teledetección . . . . .	37
<b>1.2.</b>	<b>CARACTERÍSTICAS DE LAS IMÁGENES DE SATÉLITE . . . . .</b>	<b>39</b>
1.2.1.	Imágenes de satélite . . . . .	39
1.2.2.	Resolución de las imágenes de satélite . . . . .	42
1.2.3.	Errores en la captación de imágenes . . . . .	43
<b>1.3.</b>	<b>SATÉLITES LANDSAT . . . . .</b>	<b>44</b>
1.3.1.	Características técnicas de los satélites <i>Landsat</i> . . . . .	44
1.3.2.	Combinación 3,2,1 (color verdadero) . . . . .	46
1.3.3.	Combinaciones 4,3,2 / 5,4,3 / 4,5,3 . . . . .	47
1.3.4.	Otras combinaciones . . . . .	49
<b>1.4.</b>	<b>CLASIFICACIÓN DE IMÁGENES DE SATÉLITE . . . . .</b>	<b>49</b>
1.4.1.	Clasificación no supervisada y supervisada . . . . .	50
1.4.2.	Clasificador <i>k-medias</i> . . . . .	53
1.4.3.	Clasificador <i>isodata</i> . . . . .	53
1.4.4.	Clasificador <i>paralelepípedos</i> . . . . .	55
1.4.5.	Clasificador <i>mínima distancia</i> . . . . .	56
1.4.6.	Clasificador <i>máxima verosimilitud</i> . . . . .	57
<b>1.5.</b>	<b>RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .</b>	<b>59</b>

---



**A**ctualmente existen en órbita un elevado número de satélites artificiales para la observación de la Tierra, que generan miles de imágenes a diario para todo tipo de aplicaciones de uso medioambiental, militar, social y político. Debido a la enorme cantidad de información que se genera relacionada con el ámbito de la *teledetección*, es necesaria la ayuda de algoritmos computacionales que automaticen los procesos de análisis e interpretación de las imágenes de satélite, y por este motivo una gran cantidad de grupos de investigación de todo el mundo están trabajando en la actualidad en el desarrollo de nuevas técnicas informáticas que optimicen los resultados obtenidos por los algoritmos computacionales ya existentes en la literatura. Este capítulo ofrece una introducción teórica al estado del arte de la teledetección, y está dividido en los siguientes apartados principales:

- **Introducción a la teledetección:** en este apartado se define el término teledetección, se ofrece un resumen de su historia, se enumeran los elementos básicos que intervienen en un sistema genérico de teledetección, se explican los fundamentos físicos en los que se basa su funcionamiento, y se exponen las aplicaciones más importantes de esta ciencia aplicada.
- **Características de las imágenes de satélite:** en este apartado se enumeran las propiedades más comunes que pueden presentar las imágenes de satélite, se exponen los diferentes tipos de resolución que existen, y se explican los errores más comunes que pueden ocurrir durante el proceso de captación de las imágenes de satélite.
- **Satélites *Landsat*:** en este apartado se explican las características técnicas de los satélites *Landsat*, y a continuación se exponen las aplicaciones más utilizadas de las combinaciones de bandas para las imágenes de satélite capturadas por el satélite *Landsat 7*.
- **Clasificación de imágenes de satélite:** en este apartado se exponen las diferencias entre los algoritmos de clasificación de imágenes de satélite no supervisados y supervisados, y posteriormente se realiza una descripción de varios clasificadores no supervisados (*k-medias* e *isodata*) y supervisados (*paralelepípedos*, *mínima distancia* y *máxima verosimilitud*).
- **Resumen y conclusiones del capítulo:** en este apartado se destacan los aspectos de la teledetección en los que se centra esta tesis doctoral.

## 1.1. INTRODUCCIÓN A LA TELEDETECCIÓN

En este apartado se define el término teledetección, se ofrece un resumen de la historia de la teledetección, se enumeran los elementos básicos que intervienen en un sistema genérico de teledetección, se explican los fundamentos físicos para su funcionamiento, y se exponen las aplicaciones más importantes de esta ciencia aplicada.

### 1.1.1. Definición de teledetección

Las *Naciones Unidas* definen el término **teledetección** como “la observación de la superficie terrestre desde el espacio, utilizando las propiedades de las ondas electromagnéticas emitidas, reflejadas o difractadas por los objetos observados, para fines de mejoramiento de la ordenación de los recursos naturales, de utilización de tierras y de protección del medio ambiente” [4].

Desde un punto de vista estrictamente técnico, podemos definir la teledetección como “el conjunto de técnicas, aparatos y procedimientos que permiten obtener y analizar imágenes de la superficie de la Tierra desde sensores ubicados remotamente” [27] [28]. La palabra *teledetección* realmente corresponde a la traducción de la expresión inglesa *remote sensing*, ciencia aplicada que surgió a principio de los años 60 para definir los métodos de observación remota de la superficie de la Tierra. Aunque en sus orígenes este término se aplicó principalmente a la fotografía aérea, posteriormente también se incluyó la observación a través de plataformas de observación satelitales, surgiendo así la teledetección satelital.

La teledetección es una ciencia aplicada que nos permite la adquisición de información de la superficie terrestre y de valores ambientales sin necesidad de tener contacto real con la zona que está siendo observada [89]. Esta ciencia aplicada se puede utilizar en muchas aplicaciones medioambientales, ayudando a resolver y mejorar los problemas derivados de ellos. Entre el amplio repertorio de aplicaciones de la teledetección, podemos encontrar ejemplos que incluyen estudios de calidad del suelo, búsqueda de recursos hídricos, protección del medio ambiente o simulaciones meteorológicas, entre otros. La teledetección se ha convertido en la actualidad en una herramienta indispensable para la humanidad, que puede ser usada en numerosos problemas de enorme relevancia medioambiental, militar, social y política, como analizar y predecir fenómenos meteorológicos, realizar espionaje militar para obtener ventaja estratégica en conflictos bélicos, prevenir desastres naturales para evitar pérdidas humanas, y registrar el uso del suelo de la superficie terrestre, respectivamente.

### 1.1.2. Historia de la teledetección

El origen de la teledetección se remonta a mediados del siglo XIX, y abarca desde las primeras fotografías aéreas tomadas con globo, pasando por las fotografías aéreas tomadas desde avión, hasta las avanzadas imágenes digitales captadas por los sensores ubicados en los sofisticados satélites espaciales. *Barret y Curtis* [15] dividen la historia de la teledetección en siete períodos distintos:

- **Primer período: inicio-1925.** Este período, que se trata del más largo de todos, abarca desde la captación de la primeras fotografías aéreas a mediados del siglo XIX, hasta la finalización de la Primera Guerra Mundial. La historia de la teledetección tiene sus orígenes en 1839, año en el que se captaron las primeras fotografías aéreas en Francia, para posteriormente realizar las primeras aplicaciones al mapeo topográfico, concebidas por el director del Observatorio de París. Unos años más tarde, en 1858, *Gaspard Felix Tournachon* mapeó la ciudad de París ascendiendo sobre un globo.

Gracias a los avances tecnológicos de la fotografía que se produjeron hacia 1871, se construyeron cámaras mucho más livianas, lo que impulsó el uso del globo para la captación de fotografías aéreas. En 1906, *G. Lawrence* captó los daños ocasionados por el terremoto de San Francisco desde una altitud de 600 metros. Las primeras fotografías aéreas tomadas desde un avión las captó *Wilbur Wright* en 1909 sobre Centocelli, Italia.

Durante la *Primera Guerra Mundial* se utilizó la fotografía aérea captada desde avión como herramienta de inteligencia, con el objetivo de conseguir ventaja estratégica en el campo de batalla. Este acontecimiento supuso un punto de inflexión en la historia de la teledetección, ya que a partir de este momento, demostrada su eficacia en aplicaciones militares, se comenzaron a invertir muchos más recursos económicos. Durante el primer período de la historia de la teledetección quedó demostrada la utilidad de las fotografías aéreas, obtenidas mediante el uso de globos aerostáticos o aviones, con el objetivo de conseguir ventaja estratégica militar sobre el ejército enemigo.

- **Segundo período: 1925-45.** Esta fase abarca desde la finalización de la Primera Guerra Mundial hasta la finalización de la Segunda Guerra Mundial. La *Segunda Guerra Mundial* supuso un gran impulso a la tecnología de las fotografías aéreas gracias al desarrollo de nuevas cámaras y técnicas de reconocimiento, lo que permitió mejorar la interpretación de la superficie terrestre y aumentar aún más la ventaja estratégica.
- **Tercer período: 1945-55.** Esta etapa se caracterizó por una gran difusión y diversificación de las técnicas de fotografía aérea, que comenzaron a aplicarse a diversas disciplinas *no militares*: geología, arqueología, agricultura, etc.
- **Cuarto período: 1955-60.** Durante esta fase se produjeron los primeros lanzamientos de *satélites artificiales*. El 4 de octubre de 1957, la *URSS* lanzó con éxito el *Sputnik I*, primer satélite artificial de la historia. Posteriormente se lanzó el *Sputnik II*, que puso en órbita al primer ser vivo en el espacio: la perrita *Laika*.
- **Quinto período: 1960-80.** Durante este período surgió la *teledetección satelital*, que posibilitó la adquisición sistemática de todo tipo de información de la superficie de la Tierra, sin necesidad de tener contacto físico con la zona de estudio concreta. A lo largo de este documento, cada vez que se utilice el término teledetección, se hará referencia a la teledetección satelital. Los programas espaciales permitieron impulsar los avances tecnológicos relacionados con la teledetección. En 1960 se lanzó el satélite *TIROS-I*, primer satélite meteorológico de una familia de satélites posteriormente rebautizada como *NOAA*, que permitía distinguir entre nubes, agua, hielo y nieve. En 1969, el vuelo tripulado del *Apolo 9* recogió las primeras fotografías orbitales multiespectrales para el estudio de los recursos terrestres. En 1972, se puso en órbita el primer satélite dedicado específicamente a la teledetección: el *Landsat 1*, primero de la familia de satélites *Landsat* puestos en órbita por la *NASA*.

- **Sexto período: 1980-95.** Esta fase se caracteriza por el gran desarrollo de software para la *interpretación computerizada* de las imágenes de satélite digitales. Este proceso de automatización comenzó a ser necesario como consecuencia del aumento de la cantidad de datos obtenidos, a través de las cada vez más numerosas plataformas satelitales. La puesta en marcha, durante esta fase, del transbordador espacial y de la estación espacial *MIR*, influyeron positivamente en el avance de las técnicas de teledetección espacial. Durante esta etapa, además, se lanzaron varios satélites muy importantes en el ámbito de la teledetección, como el satélite francés *SPOT* en el año 1985, el *IRS* de la India en 1988 y el *ERS* de la *Agencia Espacial Europea* en 1991.
- **Séptimo período: 1995-actualidad.** Los últimos años de la historia de la teledetección se han caracterizado sobre todo por la *comercialización* cada vez más creciente de la información obtenida mediante técnicas de teledetección. Esto ha provocado que, información que en otros períodos de la historia de la teledetección hubiera sido clasificada de espionaje militar debido a su alta relevancia estratégica, esté disponible de manera gratuita para cualquier ser humano del planeta con acceso a Internet (p.e., *Google Maps*).

Para evitar futuras apropiaciones políticas del espacio exterior existe un foro llamado *COPUOS (Committee On the Peaceful Uses of Outer Space)*, que se encarga del desarrollo de una normativa internacional del espacio, con el objetivo de que siempre sea patrimonio común de la Humanidad desde un punto de vista formalmente jurídico. Para ello se han establecido varios tratados y acuerdos internacionales que reglamentan todas las actividades relacionadas con el espacio exterior.

### 1.1.3. Elementos básicos de la teledetección

Un sistema genérico de teledetección [27] suele incluir todos los elementos que se muestran en la *Figura 1.1*:

- **Fuente de energía:** produce la radiación electromagnética captada por los sensores del satélite, que puede ser pasiva (como la luz solar) o activa (emitida por el propio sensor para posteriormente captar el reflejo).
- **Superficie terrestre:** natural o artificial, y refleja la radiación electromagnética.
- **Sistema sensor:** formado por los sensores (cámaras, radar, etc) y la plataforma de observación (satélite, avión, etc), que recoge la radiación electromagnética emitida por la superficie terrestre y la envía al sistema de recepción.
- **Sistema de recepción:** recibe y almacena la información del sistema sensor.
- **Intérprete:** transforma los datos recibidos en información útil para el usuario.
- **Analista experto:** utiliza la información interpretada con algún objetivo.

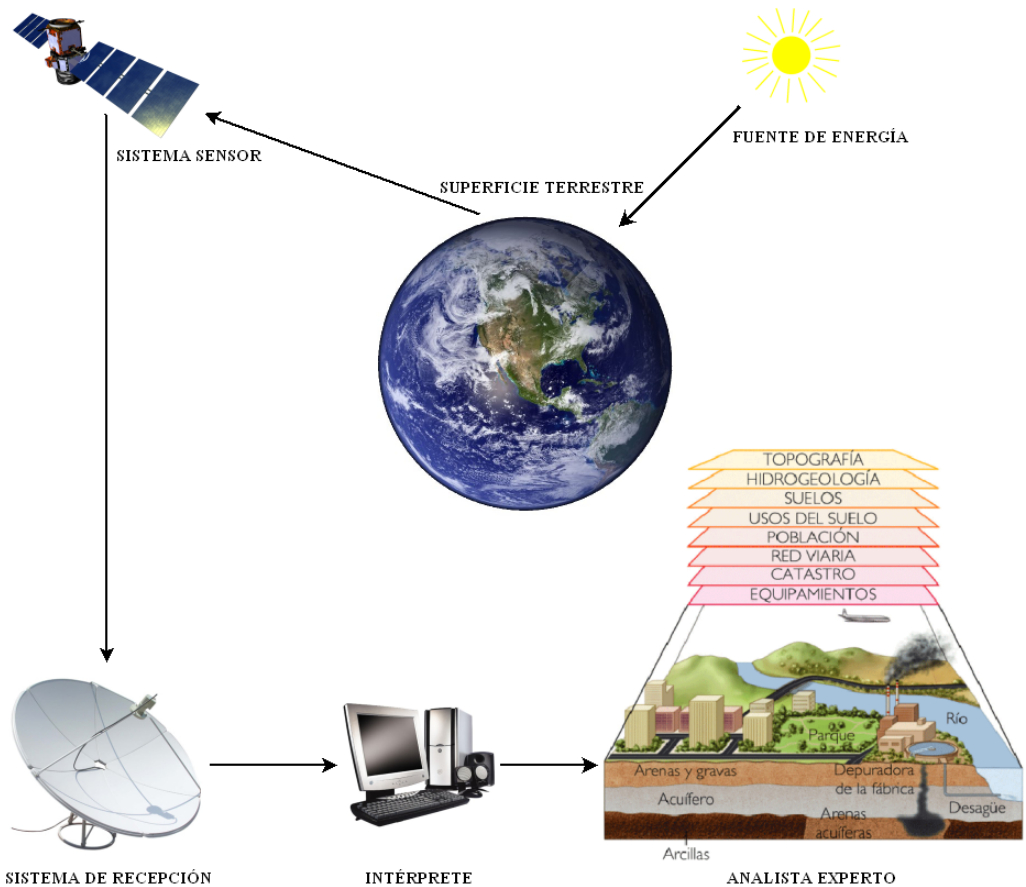


Figura 1.1: Elementos básicos de un sistema de teledetección (Fuente: E. Chuvieco).

#### 1.1.4. Fundamentos físicos de la teledetección

Según la **termodinámica**, importante rama de la física, existen tres mecanismos fundamentales para la transmisión de la energía térmica: la conducción, la convección y la radiación. De estos tres mecanismos, la *radiación* es el único en el que se transmite energía sin que exista un contacto físico entre el emisor y el receptor. Por lo tanto, es evidente que el mecanismo físico de la transmisión de energía térmica en el ámbito de la teledetección es la radiación. Existen muchos tipos de radiación: la radiación electromagnética, la radiación térmica, la radiación nuclear, la radiación ionizante, etc. De todos los tipos de radiación existentes, en esta sección nos vamos a centrar en la *radiación electromagnética*, ya que se trata de uno de los fundamentos físicos más importantes del funcionamiento de las distintas técnicas que se utilizan en el ámbito de la teledetección [51]. La naturaleza de la radiación electromagnética se puede explicar, desde un punto de vista físico, a partir de los modelos de onda y de partícula.

El **modelo de onda**, desarrollado por científicos como *Huygens y Maxwell*, establece que la radiación se produce como resultado de oscilaciones en los campos eléctrico y magnético de los objetos, lo que genera ondas con dos campos de fuerzas ortogonales entre sí (el eléctrico y el magnético), que transmiten energía. Los dos parámetros básicos que definen las ondas electromagnéticas son la longitud y la frecuencia. Dichos parámetros son inversamente proporcionales, y se relacionan entre sí a través de la siguiente expresión matemática:

$$\lambda = \frac{v}{f} \quad (1.1)$$

donde  $\lambda$  es la longitud de onda,  $f$  es la frecuencia y  $v$  la velocidad.

El **modelo de partícula**, desarrollado por científicos tan importantes como *Planck y Einstein*, establece que la energía se transmite como un flujo de partículas llamadas fotones. Los modelos de onda y partícula se relacionan entre sí a través de la siguiente expresión matemática:

$$E = \frac{hc}{\lambda} \quad (1.2)$$

donde  $E$  es la energía de los fotones,  $h$  es la constante de *Planck* ( $6,626 \cdot 10^{-34}$  Js),  $c$  es la velocidad de la luz ( $3 \cdot 10^8$  m/s) y  $\lambda$  es la longitud de onda.

El *espectro electromagnético* es la distribución energética de las longitudes de onda que puede adoptar la radiación electromagnética. Se extiende desde las ondas electromagnéticas de menor longitud de onda, como los rayos gamma y los rayos X, pasando por la luz ultravioleta, luz visible y rayos infrarrojos, hasta las radiaciones de mayor longitud de onda, como las ondas de radio. La *Figura 1.2* muestra las franjas más importantes del espectro electromagnético.

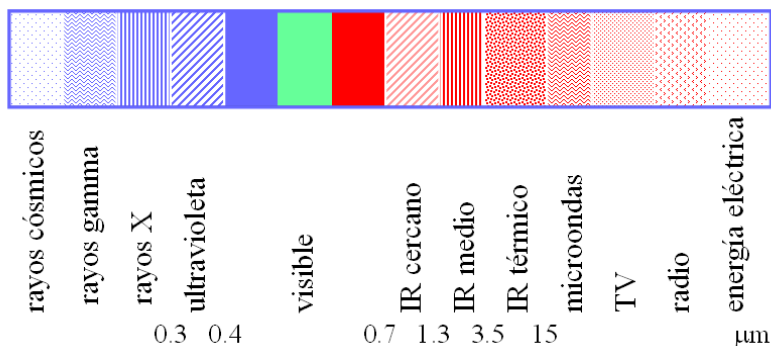


Figura 1.2: Espectro electromagnético.



### 1.1.5. Aplicaciones de la teledetección

La teledetección presenta una enorme variedad de aplicaciones de tipo medioambiental, militar, social y político:

- **Medioambiental:** la teledetección se ha aplicado a numerosos estudios relacionados con la cartografía de la cobertura vegetal del suelo, el control del movimiento de icebergs en zonas polares, la evaluación de desastres humanos o naturales, el estudio de la erosión de playas, el análisis de masas nubosas, o la verificación del índice de salinidad en corrientes de agua, entre otros.
- **Militar:** la teledetección se ha convertido en una de las mejores armas defensivas/ofensivas durante el desarrollo de un conflicto bélico. En la actualidad es impensable, para un país desarrollado, lanzar una estrategia de ataque sin haber examinado previamente el terreno mediante técnicas relacionadas con la teledetección, para establecer los objetivos estratégicos con mayor claridad.
- **Social:** la avanzada tecnología que ofrecen los satélites ha pasado de ser casi inexistente hace unos años a estar continuamente presente en nuestras vidas cotidianas: navegación a través de un *GPS*, búsqueda de lugares en *Google Maps*, televisión por satélite, o videoconferencias a través de dispositivos móviles, entre otros.
- **Político:** la teledetección se ha aplicado sobre todo para realizar el registro cartográfico de zonas rurales y urbanas, o seleccionar rutas óptimas para nuevas vías de comunicación terrestre o marítima, entre otros.

La *Figura 1.3* muestra, a través de tres imágenes *Landsat*, la evolución sufrida por el mar de Aral, situado en Asia central, durante un período de 33 años. La *Figura 1.4* muestra diversos fenómenos naturales atmosféricos captados mediante satélite.

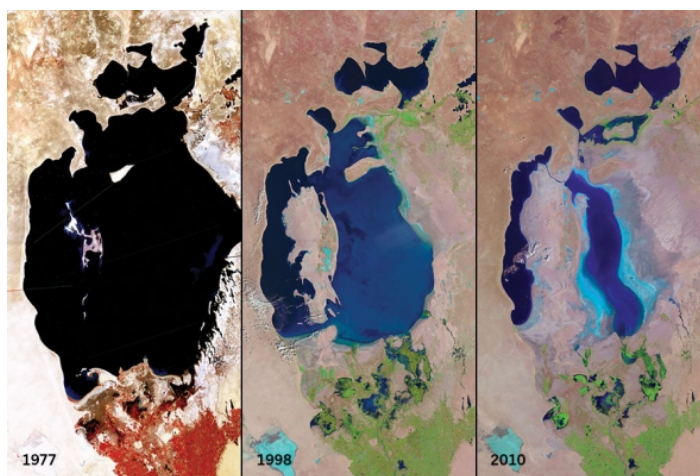


Figura 1.3: Evolución del mar de Aral en Asia central (Fuente: *Nature*).

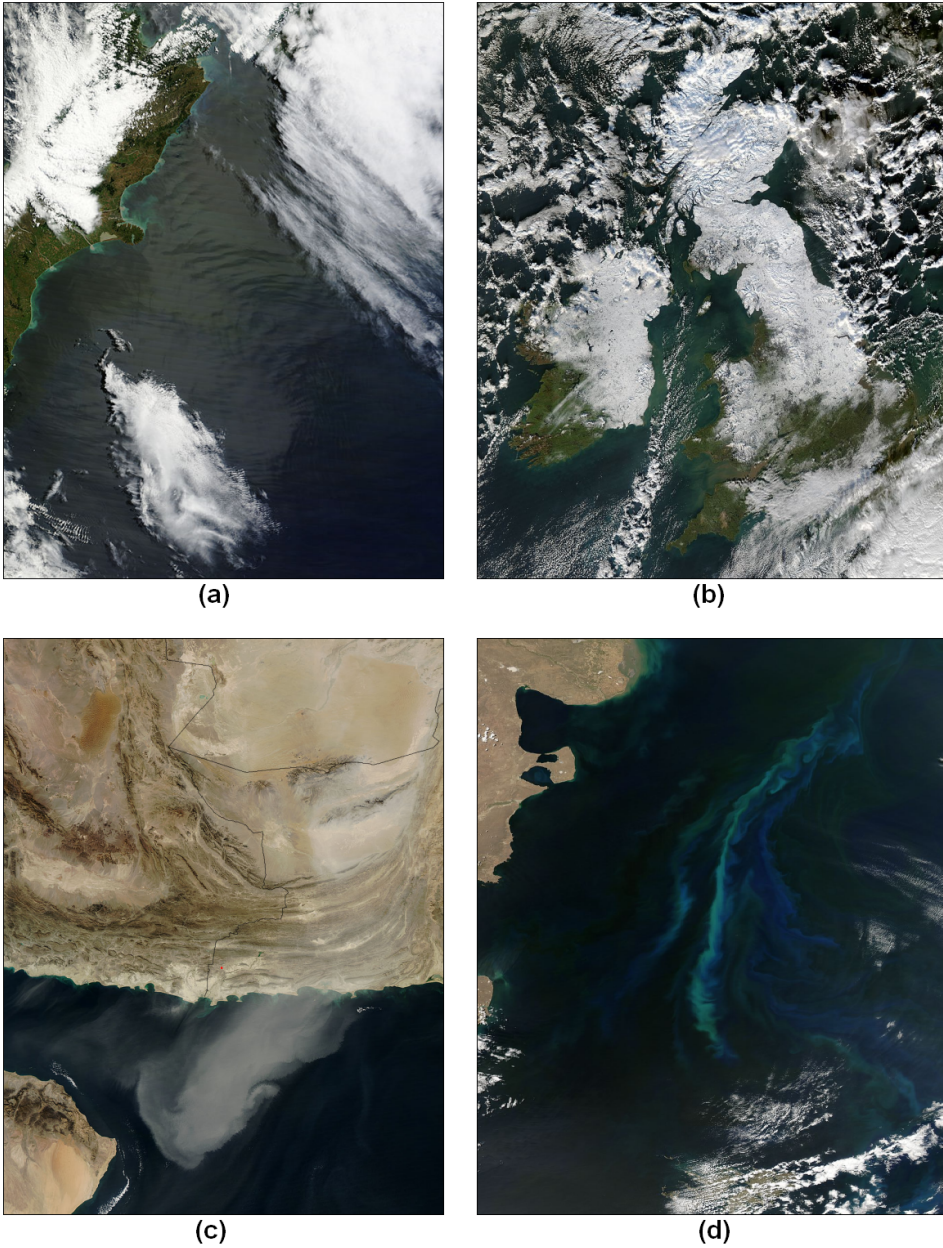


Figura 1.4: (a) Olas de gravedad atmosférica en Nueva Zelanda. (b) Nieve en Gran Bretaña e Irlanda. (c) Tormentas de polvo en el suroeste de Asia. (d) Floración de fitoplancton en Argentina (Fuente: NASA).

## 1.2. CARACTERÍSTICAS DE LAS IMÁGENES DE SATÉLITE

En este apartado se enumeran las propiedades más comunes que pueden presentar las imágenes de satélite, se exponen los diferentes tipos de resolución que existen, y se explican los errores más comunes que pueden ocurrir durante el proceso de captación de las imágenes de satélite.

### 1.2.1. Imágenes de satélite

Una **imagen de satélite** es un archivo tipo *raster* formado por una matriz de celdas, donde cada celda se denomina *píxel*. A cada píxel se le asignan varios valores digitales, que corresponden a la reflectividad recogida por un sensor específico. Las imágenes de satélite son almacenadas en diversas *bandas espectrales*, donde cada banda almacena el valor que corresponde a cada píxel de la imagen en un intervalo concreto del espectro electromagnético. Por lo tanto, una imagen de satélite es en realidad un conjunto de imágenes, con las mismas propiedades geométricas, donde cada imagen almacena el valor de reflectancia de los píxeles en un intervalo de longitud de onda concreto del espectro electromagnético. Las imágenes de satélite pueden tener un número reducido de bandas, como es el caso de las imágenes *multiespectrales* (menos de 10 bandas) o un número muy elevado, como ocurre con las imágenes *hiperespectrales* (con cientos de bandas). La *Figura 1.5* muestra una imagen de satélite multiespectral y otra hiperespectral.

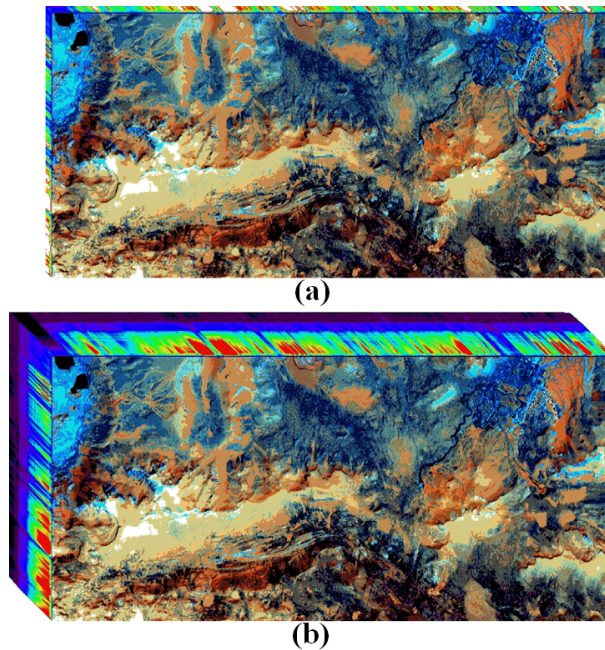


Figura 1.5: (a) Imagen multiespectral. (b) Imagen hiperespectral.

Algunas imágenes de satélite también se pueden representar como modelos de elevación digital o **DEM** (*Digital Elevation Model*), donde cada píxel almacena la altura a la que se encuentra la superficie terrestre en ese punto determinado de la imagen. De este modo, se pueden realizar representaciones tridimensionales, donde se muestra la elevación del terreno de la imagen. La *Figura 1.6* muestra un ejemplo de imagen *DEM*.

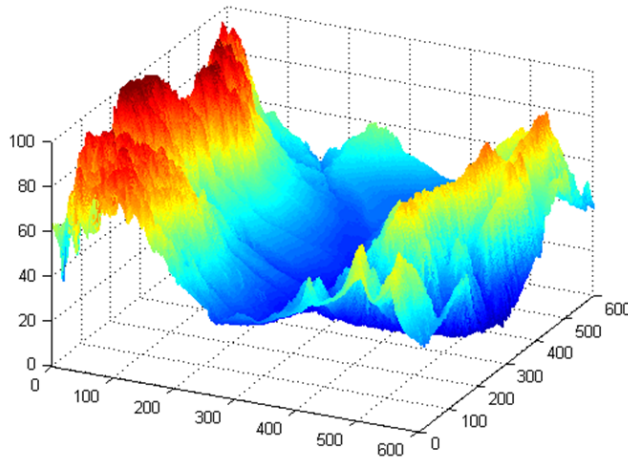


Figura 1.6: Representación 3D de una imagen *DEM*.

Las imágenes de satélite también se pueden representar, desde un punto de vista espectral, en un **espacio de características**, que es una gráfica no espacial donde se representan las clases con sus características principales (centro, variabilidad, etc), y los píxeles como los puntos correspondientes a sus valores espectrales. La *Figura 1.7* muestra un ejemplo de representación de una imagen de satélite en el espacio de características.

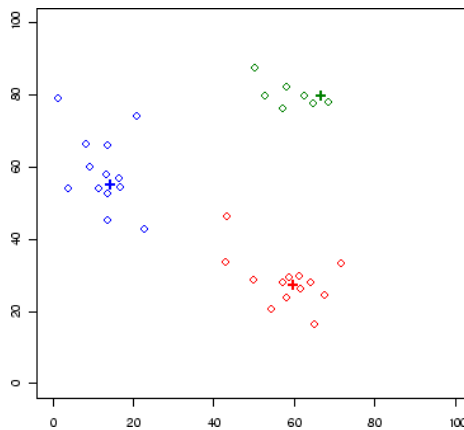
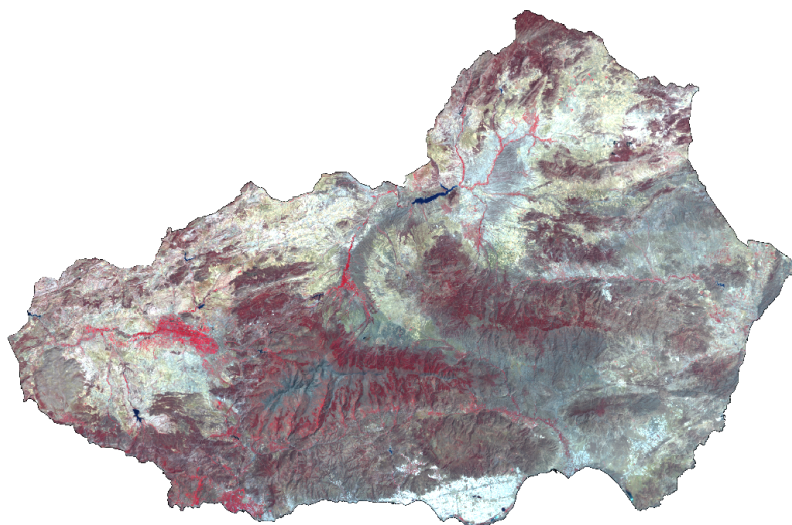


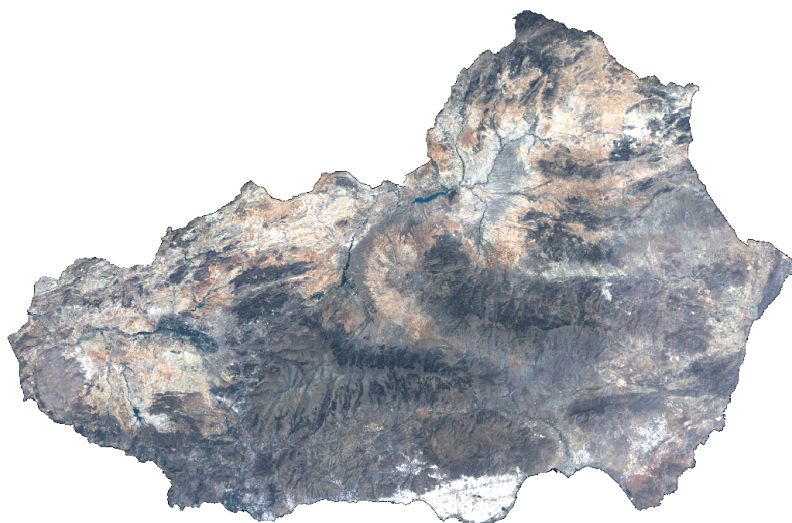
Figura 1.7: Imagen de satélite en el espacio de características (con 3 clases).



En una imagen de satélite también se pueden *combinar* distintas bandas entre sí para conseguir diferentes resultados, dependiendo del estudio que se quiera realizar. La *Figura 1.8* muestra la misma imagen de satélite, combinando distintas bandas.



(a)



(b)

Figura 1.8: (a) Imagen *Landsat* de las provincias Almería-Granada (bandas 2, 3 y 4).  
(b) Imagen *Landsat* de Almería-Granada (bandas 1, 2 y 3: color verdadero).

### 1.2.2. Resolución de las imágenes de satélite

Los satélites captan la radiación emitida o reflejada por la superficie terrestre en cuatro dimensiones distintas: espacio, tiempo, longitud de onda y radiancia. Toda esta información continua debe ser muestreada por los sensores para convertirla en datos digitales discretos. Este proceso de discretización tiene por lo tanto cuatro resoluciones posibles:

- **Resolución espacial:** es la superficie terrestre que almacena cada píxel de la imagen. También se suele usar el concepto de *IFOV*, o campo instantáneo de visión, que se define como la sección angular en radianes observada en un momento determinado. La siguiente expresión muestra la relación entre *IFOV* y el tamaño de píxel:

$$d = 2H \tan\left(\frac{IFOV}{2}\right) \quad (1.3)$$

donde  $d$  es el tamaño de píxel y  $H$  la distancia del sensor a la superficie terrestre.

Este valor puede variar bastante dependiendo del tipo de satélite que tome la imagen: desde el satélite *Meteosat*, destinado a captar información meteorológica, que ofrece una resolución espacial de 5000 metros, pasando por los satélites *Landsat*, que poseen una resolución espacial de 30 metros, hasta el sensor *Ikonos-1*, que permite una resolución de 1 metro. Dependiendo del tipo de fenómeno que queramos estudiar, convendrá elegir un tipo de resolución u otra. Si el estudio está enfocado a grandes fenómenos meteorológicos, la resolución que ofrece el satélite *Meteosat* será la adecuada para poder disponer de una cobertura global de la superficie terrestre. Si deseamos estudiar la cobertura ecológica de una determinada zona, la resolución espacial ofrecida por el satélite *Landsat* puede ser adecuada, debido a que no es ni muy elevada ni demasiado reducida. En cambio, si queremos realizar un estudio de objetos de manera individual, necesitamos utilizar la resolución ofrecida por el sensor *Ikonos-1*. Por lo tanto, la resolución espacial que elijamos para nuestro estudio debe ser lo suficientemente grande como para poder disponer de la información que necesitamos y, a su vez, lo suficientemente pequeña como para descartar información redundante. Mantener dicho equilibrio puede ahorrarnos no sólo costes computacionales innecesarios, sino también resultados erróneos, por no disponer de la suficiente información.

- **Resolución temporal:** es el intervalo de tiempo que transcurre entre cada imagen obtenida por el sensor de la misma zona de la superficie terrestre. Esta resolución puede ser la que queramos en el caso de los *aviones*, cada media hora en el caso de los satélites *geosíncronos* y variable en el caso de los satélites *heliosíncronos*.
- **Resolución espectral:** es el *número* y *anchura* de las bandas electromagnéticas captadas por el sensor. A mayor número de bandas, dispondremos de mayor número de variables que puedan describir cada píxel de la imagen. Por otro lado, las bandas estrechas aumentan el poder discriminante de los valores captados por el sensor sobre las bandas anchas. El número de bandas y la anchura depende del objetivo que se pretende cubrir con la información captada por el sensor.

- **Resolución radiométrica:** es la sensibilidad para discriminar entre pequeñas variaciones en la radiación captada por el sensor. Esta resolución se suele expresar como el número de bits necesarios para almacenar cada píxel, lo que nos ofrece el número de *niveles digitales (ND)* del sensor. A mayor resolución radiométrica, mayor información será captada por el sensor.

### 1.2.3. Errores en la captación de imágenes

Existen diversos factores que pueden influir, negativamente, durante el proceso de captación de la radiación electromagnética de la superficie terrestre por parte de los sensores de las plataformas satelitales, introduciendo errores no deseados en los datos recogidos. Estos errores se pueden clasificar en tres grandes categorías: geométricos, radiométricos y atmosféricos. A continuación se muestran los errores más frecuentes que podemos encontrar dentro de cada categoría:

- **Errores geométricos:** provocados por la propia plataforma satelital, por los sensores instalados en ella o por el fenómeno de rotación terrestre. En primer lugar, las distorsiones provocadas por la *plataforma satelital* tienen su origen en oscilaciones aleatorias de su altitud, orientación y velocidad, que alteran de manera impredecible la relación que se establece entre la superficie terrestre y las posiciones de la imagen adquirida. En segundo lugar, los errores introducidos por los *sensores* se deben a la elevada complejidad del proceso de captación de la imagen, y puede producir que no todos los píxeles de la imagen tengan la misma resolución. En tercer lugar, el fenómeno natural de *rotación de la Tierra* produce que la superficie terrestre se desplace espacialmente desde el momento de inicio del proceso de la captura de la imagen hasta el final del mismo, debido a que dicho proceso puede llegar a requerir bastante tiempo. Todos estos errores se pueden solucionar gracias a la *corrección geométrica*, mediante un proceso llamado *georreferenciación* de la imagen de satélite.
- **Errores radiométricos:** provocados por la *descalibración* de algún detector que los sensores poseen por cada una de las bandas electromagnéticas que captan, lo que puede producir efectos de bandeo en la imagen obtenida. En algunos casos pueden perderse algunos píxeles o líneas enteras. Estos errores pueden solucionarse mediante la *corrección radiométrica*.
- **Errores atmosféricos:** provocados por la interacción que se produce entre la radiación electromagnética y la *atmósfera*. Estos errores pueden solucionarse mediante la *corrección atmosférica*.

Todos los errores mencionados, aunque no suelen ser demasiado frecuentes, pueden dificultar y entorpecer el análisis de las imágenes de satélite. Por lo tanto, es importante aplicar las correcciones descritas a las imágenes de satélite que presenten cualquiera de estas características.

### 1.3. SATÉLITES LANDSAT

En este apartado se explican las características técnicas de los satélites *Landsat*, y a continuación se exponen las distintas aplicaciones más utilizadas de las combinaciones de bandas para las imágenes de satélite capturadas por *Landsat 7*.

#### 1.3.1. Características técnicas de los satélites *Landsat*

El programa *Landsat* está formado por un grupo de satélites creados y puestos en órbita por *EE.UU.* con el objetivo de observar la superficie terrestre a alta resolución. Los satélites *Landsat* están controlados por la **NASA**, aunque las imágenes recibidas son procesadas y comercializadas por la *Servicio Geológico de los Estados Unidos (USGS)* en inglés). La *Tabla 1.1* muestra el período de actividad de los 8 satélites *Landsat* lanzados hasta la fecha.

Satélite	Fecha de lanzamiento	Fin de operación
<i>Landsat 1</i>	23-07-1972	05-01-1978
<i>Landsat 2</i>	22-01-1975	27-07-1983
<i>Landsat 3</i>	05-03-1978	07-09-1993
<i>Landsat 4</i>	16-06-1982	14-12-1993
<i>Landsat 5</i>	01-03-1984	30-11-2011
<i>Landsat 6</i>	03-10-1993	Lanzamiento fallido
<i>Landsat 7</i>	15-04-1999	En activo
<i>Landsat 8</i>	11-02-2013	En activo

Tabla 1.1: Período de actividad de los satélites *Landsat* (Fuente: *NASA*).

Los experimentos llevados a cabo para desarrollar esta tesis doctoral se han realizado sobre imágenes de satélite capturadas en 2003 por parte del *Landsat 7*, cuyo esquema se muestra en la *Figura 1.9*.

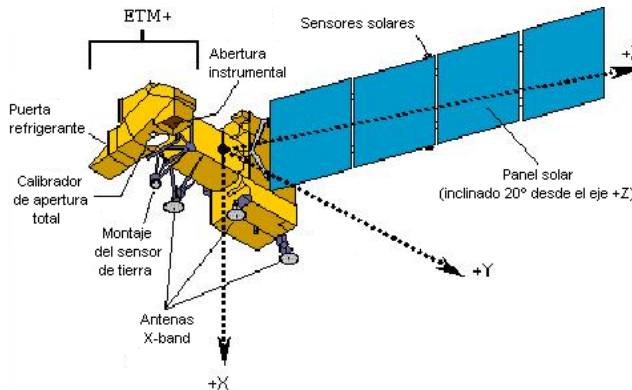


Figura 1.9: Esquema general del satélite *Landsat 7* (Fuente: *NASA*).



Los dos primeros satélites *Landsat* incorporaban un sistema de sensores formado por tres cámaras *RBV*, que registraban información en una banda espectral comprendida entre el verde y el infrarrojo cercano. Este sistema fue sustituido en el tercer satélite *Landsat* por el sistema *Vidicon*, que mejoraba la resolución espacial de las imágenes de satélite capturadas. Más adelante, el *Landsat 5* utilizó un nuevo tipo de sensor denominado *Thematic Mapper (TM)*, que mejoraba la resolución espacial, espectral y radiométrica respecto al sistema *Vidicon*. Posteriormente, el satélite *Landsat 7* fue equipado con una versión mejorada del sensor *TM*, denominado *ETM+* (*Enhanced Thematic Mapper*), que incorporaba una banda pancromática con una resolución espacial de 15 metros, y un total de 8 bandas espectrales. Para finalizar, el *Landsat 8* cuenta con dos sensores: el *OLI (Operational Land Imager)* y el *TIRS (Thermal Infrared Sensor)*, y con un total de 11 bandas espectrales. En el caso del satélite ***Landsat 7***, cuyas imágenes han sido utilizadas durante la experimentación realizada en esta tesis doctoral, la resolución espacial de sus bandas es de 30 metros, lo que permite disponer de una gran cantidad de información del área de estudio, sin que a su vez sea redundante por tener exceso de resolución. La *Tabla 1.2* muestra las características técnicas de las bandas proporcionadas por el sistema *ETM+* del *Landsat 7*.

Banda	Longitud de onda	Aplicaciones
1	0,45-0,52 (azul)	- Mapeo de aguas costeras. - Diferenciación entre suelo y vegetación.
2	0,52-0,60 (verde)	- Calidad de agua. - Mapeo de vegetación.
3	0,63-0,90 (rojo)	- Agricultura. - Calidad de agua. - Absorción de la clorofila. - Áreas urbanas y uso del suelo. - Diferenciación de especies vegetales.
4	0,76-0,90 (infrarrojo cercano)	- Agricultura y vegetación. - Delimitación de cuerpos de agua. - Mapeo geomorfológico y geológico.
5	1,55-1,75 (infrarrojo medio)	- Vegetación. - Agricultura. - Uso del suelo. - Diferenciación entre nubes y nieve. - Medición de humedad en la vegetación.
6	10,40-12,50 (infrarrojo térmico lejano)	- Corrientes marinas. - Propiedades termales del suelo. - Mapeo de stress térmico en plantas.
7	2,08-2,35 (infrarrojo térmico cercano)	- Mapeo hidrotermal. - Identificación de minerales.
8	0,52-0,90 (pancromático)	- Catastro rural, infraestructuras.

Tabla 1.2: Características técnicas del sistema *ETM+* (Fuente: *NASA*).

Estas bandas espectrales se pueden combinar entre sí para optimizar los resultados de un estudio concreto, como se muestra en las siguientes secciones.

### 1.3.2. Combinación 3,2,1 (color verdadero)

Esta combinación utiliza las tres bandas de la región visible del espectro electromagnético (*azul, verde, rojo*), por lo que es la que más se aproxima a los colores reales de la superficie terrestre vista desde el espacio. Por lo tanto, a esta combinación también se le conoce con el nombre de **color verdadero**. Es una combinación ideal para realizar estudios del agua, ya que las bandas visibles nos ofrecen una mayor respuesta a la luz que ha penetrado con mayor profundidad, y por lo tanto podemos distinguir tanto la profundidad del agua como las corrientes marinas, zonas turbias y zonas con sedimentos en suspensión. Además, nos permite observar la imagen de satélite con los colores reales que se ven con el ojo humano. La *Figura 1.10* muestra la combinación 3,2,1 de una imagen de satélite de *El Ejido*, Almería. Con esta combinación de bandas, las distintas zonas de una imagen de satélite son coloreadas de la siguiente forma:

- Azul claro: zonas costeras y con sedimentos en suspensión.
- Azul oscuro: aguas profundas y sin sedimentos en suspensión.
- Verde: vegetación variada.
- Marrón y amarillo: áreas urbanas y suelo expuesto.
- Gris azulado: invernaderos.



Figura 1.10: Combinación 3,2,1 de *El Ejido*, Almería.

### 1.3.3. Combinaciones 4,3,2 / 5,4,3 / 4,5,3

La **combinación 4,3,2** utiliza dos bandas de la región visible del espectro electromagnético (*verde, rojo*) y una banda de la región infrarroja (*infrarrojo cercano*). La banda 4 es sensible a la clorofila, por lo que nos permite observar variaciones en la vegetación. La *Figura 1.11* muestra la combinación 4,3,2 de la misma imagen de satélite. Con esta combinación de bandas, las distintas zonas de una imagen de satélite son coloreadas de la siguiente forma:

- Azul claro: zonas costeras y con sedimentos en suspensión.
- Azul oscuro: aguas profundas y sin sedimentos en suspensión.
- Rojo: vegetación variada.
- Azul claro: áreas urbanas y suelo expuesto.
- Rosa claro: invernaderos.



Figura 1.11: Combinación 4,3,2 de *El Ejido*, Almería.

La **combinación 5,4,3**, que utiliza dos bandas de la región infrarroja del espectro electromagnético (*infrarrojo cercano, infrarrojo medio*) y una banda de la región visible (*rojo*), muestra una mayor diferenciación entre el suelo y el agua. La *Figura 1.12* muestra la combinación 5,4,3 de la misma imagen de satélite. Con esta combinación de bandas, las distintas zonas de una imagen de satélite son coloreadas de la siguiente forma:



- Negro: zonas costeras y con sedimentos en suspensión.
- Negro: aguas profundas y sin sedimentos en suspensión.
- Verde: vegetación variada.
- Rosa: áreas urbanas y suelo expuesto.
- Azul: invernaderos.

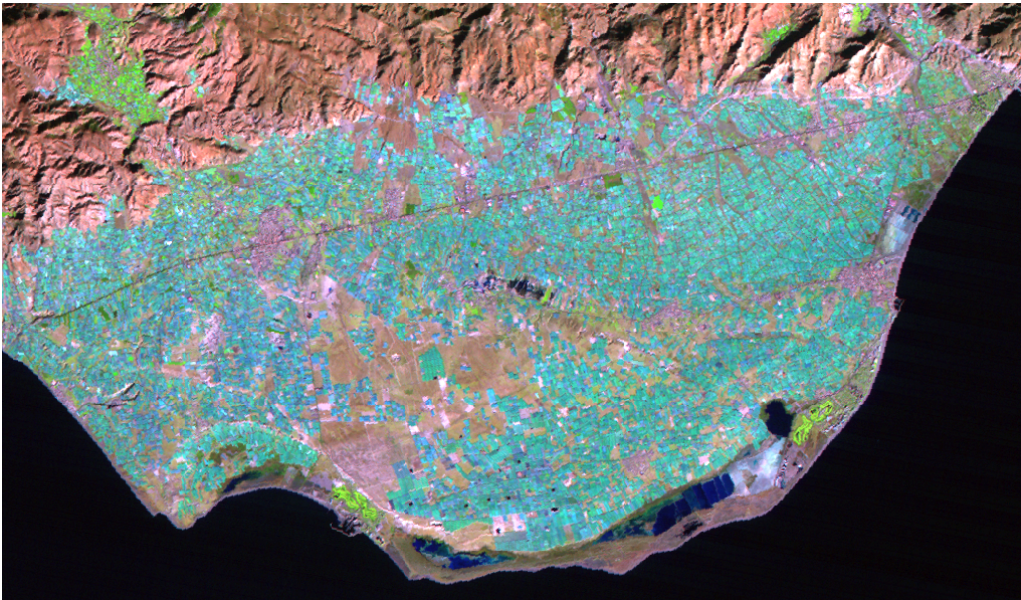


Figura 1.12: Combinación 5,4,3 de *El Ejido*, Almería.

La **combinación 4,5,3** utiliza las mismas bandas que la combinación anterior, pero asociadas a colores diferentes. La *Figura 1.13* muestra la combinación 4,5,3 de la misma imagen de satélite. Con esta combinación de bandas, las distintas zonas de una imagen de satélite son coloreadas de la siguiente forma:

- Azul oscuro: zonas costeras y con sedimentos en suspensión.
- Azul oscuro: aguas profundas y sin sedimentos en suspensión.
- Marrón, verde y amarillo: vegetación variada.
- Azul claro: áreas urbanas y suelo expuesto.
- Morado: invernaderos.

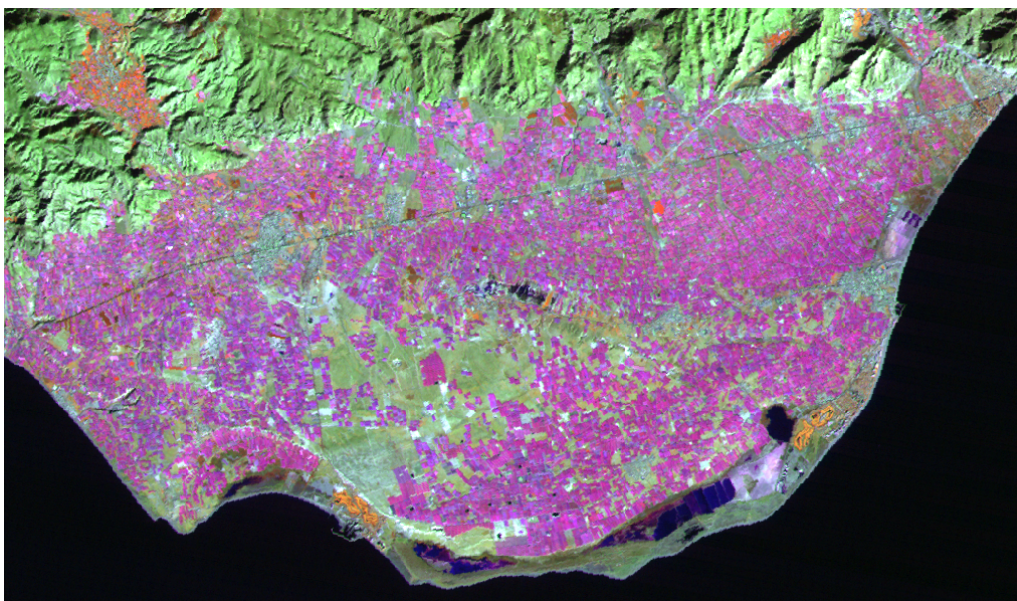


Figura 1.13: Combinación 4,5,3 de *El Ejido*, Almería.

#### 1.3.4. Otras combinaciones

Se pueden realizar otras combinaciones con las distintas bandas de las imágenes *Landsat 7*, y cada combinación tiene una aplicación concreta, dependiendo del tipo de estudio que se esté desarrollando. La *Tabla 1.3* muestra las combinaciones de bandas más frecuentes de las imágenes *Landsat 7*.

Combinación	Tipo de estudio
354	Clasificación de suelos
354, 357	Erosión de suelos
234, 345, 347	Sectores ambientales
135, 752, 753, 754	Geología
145, 157, 245, 247, 257	Clasificación de cubiertas

Tabla 1.3: Combinación de bandas de *Landsat 7*.

## 1.4. CLASIFICACIÓN DE IMÁGENES DE SATÉLITE

En este apartado se exponen las diferencias entre los algoritmos de clasificación de imágenes de satélite no supervisados y supervisados, y posteriormente se realiza una descripción de varios algoritmos no supervisados (*k-medias* e *isodata*) y supervisados (*paralelepípedos*, *mínima distancia* y *máxima verosimilitud*).

### 1.4.1. Clasificación no supervisada y supervisada

La **clasificación** de imágenes de satélite es un proceso que consiste en agrupar los píxeles de una imagen en un número finito de clases, basándose en los valores espectrales de las distintas bandas, convirtiendo de este modo la información captada por los sensores del satélite como niveles digitales a una escala categórica fácil de interpretar por los analistas expertos [73]. Los píxeles que pertenezcan a la misma clase deberán tener unas características espectrales similares [96].

Los **algoritmos de clasificación** de imágenes de satélite son una de las técnicas más importantes utilizadas en el ámbito de la teledetección, ya que facilitan a los investigadores la interpretación de la gran cantidad de información contenida en sus bandas. El objetivo de los algoritmos de clasificación de imágenes de satélite consiste en dividir los píxeles de la imagen en distintas clases, llamadas *clases espectrales*, teniendo en cuenta la similitud existente entre dichos píxeles. La imagen que se obtiene como resultado consiste básicamente en un mapa temático de la imagen original [96]. Los analistas expertos utilizan posteriormente la imagen clasificada para interpretar de manera más sencilla la información contenida en la imagen de satélite. Los algoritmos de clasificación de imágenes de satélite han experimentado un gran avance en los últimos años, debido a que existen numerosos grupos de investigación intentando mejorar la tasa de acierto de los algoritmos ya existentes. Como consecuencia, en la literatura existe una gran cantidad de procedimientos para clasificar imágenes de satélite.

La *Figura 1.14* muestra el esquema general del proceso de clasificación de una imagen de satélite. Como se puede observar en la parte izquierda, la imagen tiene 4 bandas, y se han tomado los valores espectrales de un píxel concreto como ejemplo, con el objetivo de clasificarlo en una de las tres clases existentes: tierra, agua o plantas. En la parte derecha de la imagen se muestra un ejemplo de cómo quedaría la imagen original después del proceso de clasificación, en la que cada píxel de la imagen ha sido etiquetado en su clase correspondiente, teniendo en cuenta sus cuatro valores espectrales.



Figura 1.14: Esquema general del proceso de clasificación de imágenes de satélite.

En cualquier algoritmo de clasificación de imágenes de satélite podemos encontrar las siguientes fases genéricas:

- **Fase de entrenamiento:** definición de las *clases* en las que se van a agrupar los píxeles de la imagen. En esta fase se define el número de clases que participará en el proceso de clasificación, así como los rangos que abarcarán las propiedades espectrales de sus píxeles.
- **Fase de asignación:** agrupación de los píxeles en las distintas clases previamente definidas. Durante esta fase, cada píxel de la imagen de satélite es *etiquetado* en su clase correspondiente, teniendo en cuenta la información obtenida por la fase de entrenamiento.
- **Verificación:** comprobación de los resultados obtenidos. En esta fase se analiza la *tasa de acierto* obtenida por el algoritmo de clasificación, utilizando para ello una matriz de confusión u otro método de verificación.

Los algoritmos de clasificación de imágenes de satélite se pueden dividir en dos grandes grupos, dependiendo del método utilizado durante la fase de entrenamiento: clasificación no supervisada y clasificación supervisada, ambos tipos de clasificadores basados en las propiedades espectrales de los píxeles de la imagen. La utilización de procedimientos no supervisados o supervisados depende fundamentalmente del conocimiento que posee el analista experto sobre la zona de estudio [6]. Si se posee conocimiento humano experto de la zona de estudio, es recomendable usar siempre una clasificación supervisada [80]. La *Figura 1.15* muestra el proceso general de clasificación no supervisada y supervisada, cada uno en un diagrama de flujo.

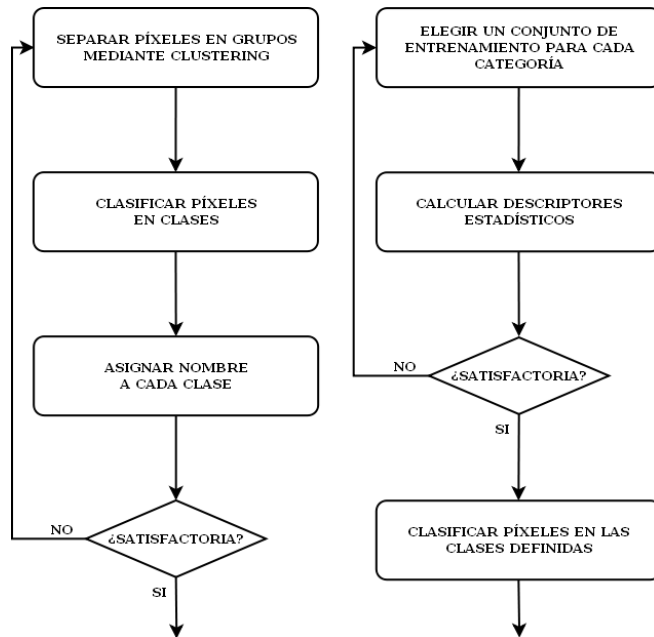


Figura 1.15: (Izq.) Clasificación no supervisada. (Dcha.) Clasificación supervisada.



En los **algoritmos de clasificación no supervisados**, el analista experto sólo debe especificar el número de clases en las que desea dividir la imagen, y el propio algoritmo se encarga de agrupar los píxeles similares basándose únicamente en la información espectral almacenada en las bandas de cada píxel. En este tipo de algoritmos, por tanto, no es necesario que el analista experto conozca la zona de estudio. Los algoritmos clasificadores *k-medias* e *isodata* son dos de los procedimientos más utilizados para la clasificación no supervisada de imágenes de satélite.

En los **algoritmos de clasificación supervisados**, el analista experto selecciona varias muestras de los píxeles de cada clase, formando el denominado conjunto de entrenamiento, para que el algoritmo pueda realizar una clasificación más precisa basada en el conocimiento humano. En este tipo de algoritmos, el conocimiento que posee el analista experto sobre el área de estudio determina la calidad del conjunto de entrenamiento, y por lo tanto es un factor muy influyente en el resultado final de la clasificación. Los píxeles son etiquetados en la clase a la que más se asemejan a nivel espectral [9], teniendo en cuenta el conjunto de entrenamiento. Existen muchos algoritmos de clasificación supervisados, como *paralelepípedos*, *mínima distancia* y *máxima verosimilitud*. La *Figura 1.16* muestra la descripción del proceso general de clasificación (a), así como los resultados obtenidos en una clasificación no supervisada (b) y una clasificación supervisada (c) sobre una imagen de satélite de la provincia de Almería.

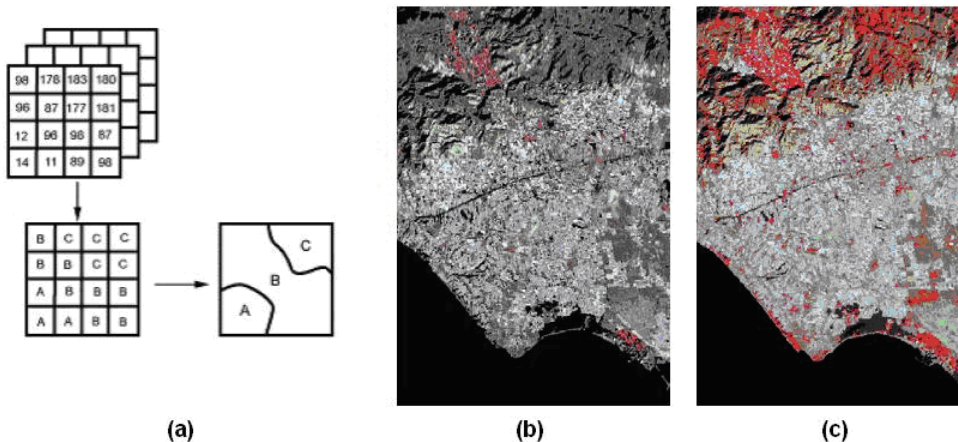


Figura 1.16: (a) Descripción del proceso general de clasificación. (b) Resultados de una clasificación no supervisada. (c) Resultados de una clasificación supervisada.

Este tipo de algoritmos de clasificación de imágenes de satélite se denominan clasificadores *espectrales*, ya que clasifican cada píxel teniendo en cuenta sus valores espectrales propios, con o sin la ayuda del conjunto de entrenamiento definido por el analista experto. Tanto los algoritmos de clasificación espectrales supervisados como los no supervisados funcionan bastante bien en imágenes que no están alteradas por el ruido y donde las propiedades espectrales de los píxeles pueden establecer las clases con suficiente certeza, sin que existan clases muy próximas entre sí.



Sin embargo, si las imágenes se encuentran alteradas por ruido de tipo *impulsivo-gaussiano* o existen variaciones sustanciales en las propiedades de los píxeles, la imagen resultante puede tener muchas pequeñas regiones (a menudo de un píxel) que estén mal clasificadas. Para solucionar este error de clasificación se pueden utilizar técnicas adicionales a las espectrales, como el uso de **algoritmos contextuales**. En estos algoritmos no sólo se tienen en cuenta los valores espectrales de cada píxel de la imagen, sino que también se tienen en cuenta los valores de los píxeles de alrededor, creando una vecindad para cada píxel. Existen varios algoritmos de clasificación contextual que utilizan valores *promedios*, valores *máximos* o descripción de las *texturas* para mejorar la clasificación espectral. En las siguientes secciones se explica el funcionamiento de dos algoritmos de clasificación no supervisados (*k-medias* e *isodata*) y tres algoritmos supervisados (*paralelepípedos*, *mínima distancia* y *máxima verosimilitud*).

### 1.4.2. Clasificador *k-medias*

El algoritmo *k-medias* [69] es uno de los clasificadores no supervisados más sencillos. Se trata de un método de análisis de grupos que tiene como objetivo realizar una partición de un conjunto de elementos de entrada  $E$  en un número de *clusters* (o grupos)  $k$  determinado de antemano, de tal modo que cada *cluster* incorpora los elementos con la media más cercana. A continuación se muestran los pasos del algoritmo *k-medias*:

- #1. Se determina el número de clusters  $k$ .
- #2. Se seleccionan  $k$  centros iniciales tomados al azar.
- #3. Se asigna cada uno de los píxeles al cluster cuyo centro se encuentre más cercano.
- #4. Se calcula el valor de los centros con los nuevos píxeles asignados como el promedio de los elementos del cluster.
- #5. Si las fronteras de los clusters cambian significativamente, volver al paso 3.

Como se puede apreciar, en cada iteración se va refinando el proceso de *clustering*, de tal modo que se van agrupando los píxeles con similares características en la misma clase. Como salida, el algoritmo *k-medias* ofrece un conjunto formado por los centros de las  $k$  clases, así como los elementos de entrada  $E$  etiquetados con la clase correspondiente. La *Tabla 1.4* muestra el pseudocódigo del algoritmo *k-medias*.

### 1.4.3. Clasificador *isodata*

El algoritmo *isodata* [11] es una mejora del algoritmo *k-medias*, ya que permite no sólo especificar un número máximo de iteraciones para realizar el proceso de *clustering*, sino además establecer un umbral de convergencia que determine el porcentaje de píxeles mínimo que se debe modificar entre dos iteraciones consecutivas para continuar con el algoritmo. Gracias a estas mejoras se consigue mejorar el proceso de creación de dicho agrupamiento. A continuación se muestran los pasos del algoritmo *isodata*:

---

<b>Algoritmo <math>k</math>-medias</b> ( $E, k$ )	
<hr/>	
<b>Entrada:</b>	
	$E = \{e_1, e_2, \dots, e_n\}$ : conjunto de píxeles a clasificar
	$k$ : número de clases
<b>Salida:</b>	
	$C = \{c_1, c_2, \dots, c_k\}$ : conjunto de centros de las clases
	$L = \{l(e)   e = 1, 2, \dots, n\}$ : píxeles clasificados
<hr/>	
01	<b>foreach</b> $c_i \in C$ <b>do</b>
02	$c_i \leftarrow e_j \in E$
03	<b>end</b>
04	<b>foreach</b> $e_i \in E$ <b>do</b>
05	$l(e_i) \leftarrow \min \text{Distance}(e_i, c_j)   j \in \{1..k\}$
06	<b>end</b>
07	$changed \leftarrow false$ ;
08	<b>repeat</b>
09	<b>foreach</b> $c_i \in C$ <b>do</b>
10	$UpdateCluster(c_i)$
11	<b>end</b>
12	<b>foreach</b> $e_i \in E$ <b>do</b>
13	$minDist \leftarrow \min \text{Distance}(e_i, c_j)   j \in \{1..k\}$
14	<b>if</b> $minDist \neq l(e_i)$ <b>then</b>
15	$l(e_i) \leftarrow minDist$
16	$changed \leftarrow true$
17	<b>end</b>
18	<b>end</b>
19	<b>until</b> $changed = true$

---

Tabla 1.4: Algoritmo de clasificación  $k$ -medias.

- #1. Se introducen los siguientes parámetros:  $k$  (número máximo de clusters),  $T$  (umbral de convergencia) y  $maxIters$  (número máximo de iteraciones).
- #2. Se determinan de forma arbitraria los centros de los  $k$  clusters.
- #3. Se calcula la distancia espectral de cada uno de los píxeles de la imagen a los centros. El píxel se asigna a la clase cuyo centro esté más cercano.
- #4. Con los nuevos píxeles asignados a clusters, se recalculan los nuevos centros.
- #5. Si alguno de los clusters es demasiado pequeño se puede eliminar, disminuyendo el número de clusters determinado en un principio por el usuario. Si los centros de dos clusters están muy cercanos, éstos se pueden unir, y si un cluster contiene muchos objetos y su varianza en alguna variable es extremadamente grande, este cluster se divide en dos, siempre que no se supere el número máximo de clusters introducidos por el usuario,  $k$ .
- #6. Si no se ha alcanzado el número máximo de iteraciones  $maxIters$ , o el porcentaje de píxeles que se modifica entre dos iteraciones está por debajo del umbral de convergencia  $T$ , se vuelve al paso 3.

El algoritmo *isodata* es uno de los clasificadores no supervisados más utilizados, por sus buenos resultados. La *Tabla 1.5* muestra el pseudocódigo del algoritmo *isodata*.

<b>Algoritmo <i>isodata</i></b> ( $E, k, T, maxIters$ )	
<b>Entrada:</b>	
	$E = \{e_1, e_2, \dots, e_n\}$ : conjunto de píxeles a clasificar
	$k$ : número de clases
	$T$ : umbral de convergencia
	$maxIters$ : máximo de iteraciones
<b>Salida:</b>	
	$C = \{c_1, c_2, \dots, c_k\}$ : conjunto de centros de las clases
	$L = \{l(e)   e = 1, 2, \dots, n\}$ : píxeles clasificados
01	<b>foreach</b> $c_i \in C$ <b>do</b>
02	$c_i \leftarrow e_j \in E$
03	<b>end</b>
04	<b>foreach</b> $e_i \in E$ <b>do</b>
05	$l(e_i) \leftarrow \minDistance(e_i, c_j)   j \in \{1..k\}$
06	<b>end</b>
07	$iter \leftarrow 0$ ;
08	<b>repeat</b>
09	$changed \leftarrow 0$ ;
10	<b>foreach</b> $c_i \in C$ <b>do</b>
11	$UpdateCluster(c_i)$
12	<b>end</b>
13	<b>foreach</b> $e_i \in E$ <b>do</b>
14	$minDist \leftarrow \minDistance(e_i, c_j)   j \in \{1..k\}$
15	<b>if</b> $minDist \neq l(e_i)$ <b>then</b>
16	$l(e_i) \leftarrow minDist$
17	$changed \leftarrow changed + 1$
18	<b>end</b>
19	<b>end</b>
20	$iter ++$ ;
21	<b>until</b> $changed < T$ and $iter \leq maxIters$

Tabla 1.5: Algoritmo de clasificación *isodata*.

#### 1.4.4. Clasificador *paralelepípedos*

En el caso del clasificador *paralelepípedos* [38], el usuario fija un área de dominio para cada categoría, teniendo en cuenta sus valores de centralidad y dispersión. Posteriormente cada píxel es asignado a una clase si sus niveles digitales  $ND$  están dentro del área de dominio de esa clase, para todas las bandas que intervienen en el proceso de clasificación. El píxel  $x$  es asignado a la clase  $A$  si sus  $ND$  en las distintas bandas ( $ND_{x,n}$ ) están incluidos en el área de dominio de esa clase, como muestra la siguiente expresión:

$$\overline{VC}_{A,n} - RD_{A,n} \leq ND_{x,n} \leq \overline{VC}_{A,n} + RD_{A,n} \quad (1.4)$$

donde:

–  $\overline{VC}_{A,n}$ : valor central de la clase  $A$  en la banda  $n$ .

- $RD_{A,n}$ : rango de dispersión de la clase  $A$  en la banda  $n$ .
- $ND_{x,n}$ : nivel digital del píxel  $x$  en la banda  $n$ .
- $n = 1, 2, \dots, N$ : bandas de la imagen de satélite.

El clasificador *paralelepípedos* tiene dos **ventajas**: considera la dispersión de las clases y es rápido en ejecución. Sin embargo también tiene dos **limitaciones**: algunos píxeles pueden clasificarse erróneamente en varias clases y pueden quedar píxeles sin clasificar. El algoritmo *paralelepípedos*, por tanto, tiene dos limitaciones en su funcionamiento. Limitaciones que, como veremos a lo largo del *Capítulo 3*, pueden solucionarse mediante el uso de autómatas celulares. La *Figura 1.17* muestra el proceso general de clasificación mediante el algoritmo *paralelepípedos* en el espacio de características de la imagen de satélite, considerando tres clases.

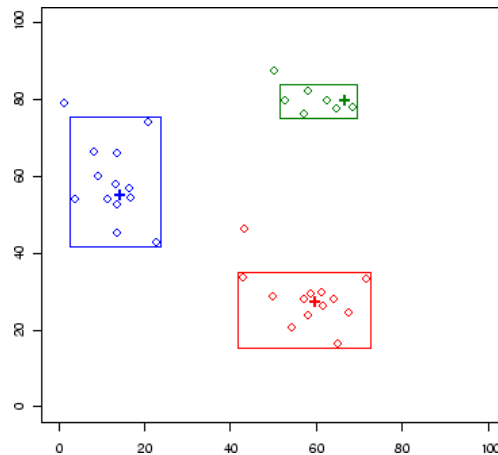


Figura 1.17: Clasificación mediante el algoritmo *paralelepípedos*.

#### 1.4.5. Clasificador *mínima distancia*

El clasificador *mínima distancia* [108] asigna a cada píxel la clase que se encuentra a menor distancia espectral respecto a su centroide, teniendo en cuenta todas las bandas que intervienen en el análisis. Para calcular la distancia entre el píxel y las clases se utiliza la fórmula de la *distancia euclídea*:

$$dist_{x,A} = \sqrt{\sum_{n=1}^{N_{bandas}} (ND_{x,n} - \overline{VC}_{A,n})^2} \quad (1.5)$$

donde:

- $dist_{x,A}$ : distancia entre el píxel  $x$  y la clase  $A$ .

- $ND_{x,n}$ : nivel digital del píxel  $x$  en la banda  $n$ .
- $\overline{VC}_{A,n}$ : valor central de la clase  $A$  en la banda  $n$ .
- $n = 1, 2, \dots, N$ : bandas de la imagen de satélite.

Cuando se han calculado todas las distancias entre el píxel y los centroides de las clases, el algoritmo asigna al píxel la clase más cercana siguiendo la siguiente expresión:

$$clase(x) = \{A | dist_{x,A} = \min\} \quad (1.6)$$

El clasificador *mínima distancia* también tiene dos **ventajas**: todos los píxeles de la imagen son clasificados y es rápido en ejecución. Y también tiene dos **limitaciones**: es proclive a errores de comisión (asignar un píxel a una clase errónea) y no considera la varianza de las clases. Estas limitaciones también pueden solucionarse mediante el uso de autómatas celulares, como veremos a lo largo del *Capítulo 3*. La *Figura 1.18* muestra el proceso general de clasificación mediante el algoritmo *mínima distancia* en el espacio de características de la imagen de satélite, considerando tres clases.

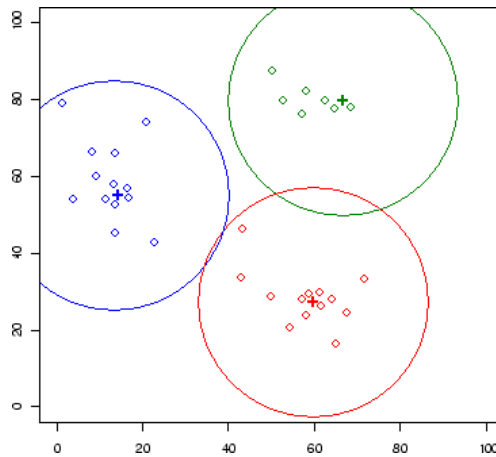


Figura 1.18: Clasificación mediante el algoritmo *mínima distancia*.

#### 1.4.6. Clasificador *máxima verosimilitud*

El clasificador *máxima verosimilitud* [61] considera que los  $ND$  de cada categoría se ajustan a una distribución normal. Esto nos permite describir esa categoría mediante una función de probabilidad, a partir de su vector de medias y matriz de covarianza. De este modo, podemos calcular la probabilidad de que un determinado píxel pertenezca a una categoría: el cálculo se realiza para todas las categorías definidas y el píxel se clasifica en la clase a la cual tenga más probabilidad de pertenecer.

Para calcular la probabilidad asociada a un cierto  $ND$  se necesita la media y la desviación típica de cada categoría. En el caso de trabajar con una sola banda, la probabilidad se calcula mediante la siguiente expresión:

$$p(x/A) = \frac{1}{\sqrt{2\pi\sigma_A^2}} e^{\{-(ND_x - \bar{M}_A)^2 / 2\sigma_A^2\}} \quad (1.7)$$

donde:

- $p(x/A)$ : probabilidad de que el píxel  $x$  pertenezca a la clase  $A$ .
- $ND_x$ : nivel digital del píxel  $x$ .
- $\bar{M}_A$ : media de la clase  $A$ .
- $\sigma_A^2$ : varianza de la clase  $A$ .

Una vez calculada la probabilidad para cada clase, un píxel es asignado a la clase  $A$  si se cumple la siguiente condición:

$$p(x/A) \geq p(x/B) \quad (1.8)$$

$A \neq B$ , con  $B = 1, \dots, numClases$ .

Este clasificador es más fiable que los dos anteriores, ya que considera la variabilidad. Sin embargo, tiene como *inconvenientes* que requiere un gran coste computacional y sobreclasifica las firmas. La *Figura 1.19* muestra el proceso general de clasificación mediante el algoritmo *máxima verosimilitud* en el espacio de características de la imagen de satélite, considerando tres clases.

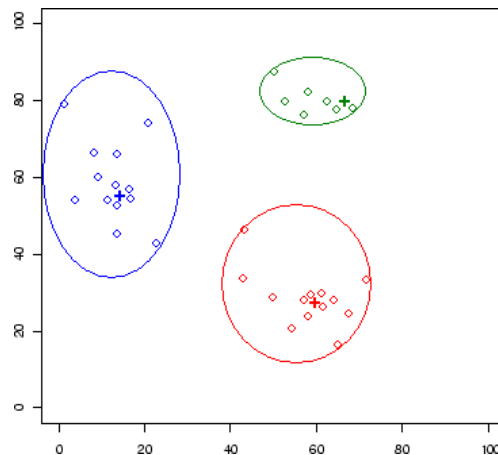


Figura 1.19: Clasificación mediante el algoritmo *máxima verosimilitud*.

## 1.5. RESUMEN Y CONCLUSIONES DEL CAPÍTULO

La **teledetección** es una ciencia aplicada que se utiliza en la actualidad para resolver numerosos problemas de tipo medioambiental, militar, social y político a través de las imágenes de satélite. En este capítulo nos hemos centrado en las características de las imágenes del satélite *Landsat 7*, porque son las que se han utilizado para llevar a cabo la mayoría de los experimentos de esta tesis doctoral, como veremos más adelante. Una de las técnicas más importantes empleadas en la teledetección son los **algoritmos de clasificación** de imágenes de satélite. A pesar del elevado número de algoritmos supervisados y no supervisados que existe en la actualidad, ninguno es 100% eficiente, ya que todos presentan una serie de **limitaciones** en su funcionamiento, incluso trabajando con imágenes de satélite que no presentan problemas añadidos.

Además, como vimos en el *Prólogo*, existen algunos píxeles, denominados **inciertos** a lo largo de este documento, que se encuentran a una distancia espectral casi equivalente de dos o más clases diferentes. Debido a este hecho, dichos píxeles pueden ser erróneamente clasificados si sólo tenemos en cuenta la información espectral de sus bandas. También podemos encontrarnos con píxeles **ruidosos** en una imagen, aunque en un porcentaje de frecuencia mucho menor que los píxeles inciertos. Estos dos tipos de píxeles, utilizando los algoritmos clásicos descritos a lo largo de este capítulo, seguramente serán erróneamente clasificados.

Por lo tanto es necesario utilizar algunas técnicas complementarias para ayudar a mejorar dicho proceso de clasificación, como pueden ser los **algoritmos contextuales**. Hasta el momento, las imágenes que incorporan una tasa de ruido, como consecuencia de errores ocurridos durante el proceso de captación de los sensores del satélite, son tratadas mediante un determinado tipo de filtros aplicados antes del proceso de clasificación (*pre-clasificación*), y los píxeles inciertos pueden ser correctamente clasificados mediante algoritmos contextuales que se aplican a la clasificación obtenida previamente (*post-clasificación*). Sería interesante fusionar estos tres procesos en uno solo, para mejorar la tasa de acierto y reducir el coste en tiempo.

El *Capítulo 2* ofrece una introducción teórica al modelo matemático del autómata celular, y ofrece una descripción detallada dos algoritmos determinados, ambos basados en los autómatas celulares e implementados desde el proyecto *SOLERES* durante el desarrollo de esta tesis doctoral, llamados *RACA* y *TACA*, para la simulación de precipitaciones sobre imágenes de satélite *DEM* y la caracterización de texturas de imágenes-AC y naturales, respectivamente.





---

## CAPÍTULO 2

# AUTÓMATAS CELULARES

---



# Capítulo 2

## AUTÓMATAS CELULARES

### Contenidos

---

<b>2.1.</b>	<b>INTRODUCCIÓN A LOS AUTÓMATAS CELULARES . . . . .</b>	<b>65</b>
2.1.1.	Definición conceptual de autómata celular . . . . .	66
2.1.2.	Definición formal de autómata celular . . . . .	67
2.1.3.	Clasificación de los autómatas celulares . . . . .	68
<b>2.2.</b>	<b>APLICACIONES DE LOS AUTÓMATAS CELULARES . . . . .</b>	<b>68</b>
2.2.1.	Autómatas celulares unidimensionales. <i>Stephen Wolfram</i> . . . . .	69
2.2.2.	Autómatas celulares bidimensionales. <i>Juego de la vida</i> . . . . .	72
2.2.3.	Autómatas celulares aplicados a la teledetección . . . . .	75
<b>2.3.</b>	<b>SIMULACIÓN DE PRECIPITACIONES MEDIANTE RACA . . . . .</b>	<b>76</b>
2.3.1.	Modelado y simulación de precipitaciones . . . . .	76
2.3.2.	Objetivos propuestos . . . . .	78
2.3.3.	Algoritmo RACA . . . . .	78
2.3.4.	Características experimentales . . . . .	82
2.3.5.	Casos de estudio del algoritmo RACA . . . . .	83
<b>2.4.</b>	<b>CARACTERIZACIÓN DE TEXTURAS MEDIANTE TACA . . . . .</b>	<b>87</b>
2.4.1.	Números decimales de las reglas de <i>Stephen Wolfram</i> . . . . .	87
2.4.2.	Caracterización de texturas de imágenes-AC . . . . .	88
2.4.3.	Caracterización de texturas naturales. Algoritmo TACA . . . . .	90
2.4.4.	Casos de estudio del algoritmo TACA . . . . .	92
<b>2.5.</b>	<b>RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .</b>	<b>94</b>

---



**U**n *autómata celular* es un modelo matemático que evoluciona en pasos discretos, y que normalmente se utiliza para simular sistemas de entidades naturales o artificiales que interactúan entre sí. Desde el origen de la teoría del autómata celular, gracias a *von Neumann*, los autómatas celulares han sido ampliamente utilizados para resolver problemas relacionados con los sistemas complejos y, sobre todo, para implementar todo tipo de simulaciones naturales o artificiales. Este capítulo ofrece una introducción teórica al modelo matemático del autómata celular, y está dividido en los siguientes apartados principales:

- **Introducción a los autómatas celulares:** en este apartado se ofrece una definición conceptual de autómata celular así como una posterior definición formal matemática, y además se muestra una clasificación de los autómatas celulares unidimensionales.
- **Aplicaciones de los autómatas celulares:** en este apartado se muestran las principales aplicaciones de los autómatas celulares unidimensionales así como la notación de *Stephen Wolfram*, las aplicaciones de los autómatas celulares bidimensionales, como por ejemplo el *juego de la vida* de *Conway*, y las aplicaciones más importantes de los autómatas celulares en el ámbito de la teledetección.
- **Simulación de precipitaciones mediante el algoritmo RACA:** en este apartado se ofrece una introducción al modelado y la simulación de precipitaciones, se exponen los objetivos propuestos para este método de simulación basado en autómatas celulares, se realiza una descripción del algoritmo RACA, y para finalizar se muestran las características experimentales y los casos de estudio sobre los que se ha aplicado.
- **Caracterización de texturas mediante el algoritmo TACA:** en este apartado se explica cómo obtener números decimales a partir de las reglas de *Stephen Wolfram* a través de un proceso de ingeniería inversa, se expone cómo usar los números decimales obtenidos para caracterizar imágenes-AC y texturas naturales ampliando el procedimiento anterior, y se muestran los casos de estudio sobre los que se ha aplicado.
- **Resumen y conclusiones del capítulo:** en este apartado se destacan los aspectos de los autómatas celulares en los que se centra esta tesis doctoral.

## 2.1. INTRODUCCIÓN A LOS AUTÓMATAS CELULARES

En este apartado se ofrece una definición conceptual de autómata celular así como una posterior definición formal matemática, y además se muestra una clasificación de los autómatas celulares unidimensionales.

### 2.1.1. Definición conceptual de autómatas celulares

Un **autómata celular** es un modelo matemático que consiste en una malla de celdas distribuidas normalmente en forma de matriz bidimensional. Su precursor fue *John von Neumann* [105] a mediados del siglo XX, y tiene las siguientes características básicas:

- **Estados:** cada celda toma un valor alfanumérico que corresponde con su estado actual. Existe un conjunto finito de estados posibles.
- **Vecindad:** conjunto de celdas que interactúan con la actual. La *Figura 2.1* muestra los tipos de vecindad más comunes de los autómatas celulares: 4 vecinos (vecindad de *von Neumann*), 8 vecinos (vecindad de *Moore*) y 24 vecinos (vecindad de *Moore extendida*).
- **$f$ :** función de transición, que toma como entrada el estado de la celda actual y de la vecindad, y devuelve un nuevo estado para la celda actual.
- **Reglas:** la función de transición  $f$  utiliza un conjunto de reglas, que especifican cómo cambian los estados de las celdas dependiendo de los valores de entrada que se introduzcan.
- **Iteraciones:** la función de transición  $f$  se aplica a cada celda de la malla a través de varias iteraciones.

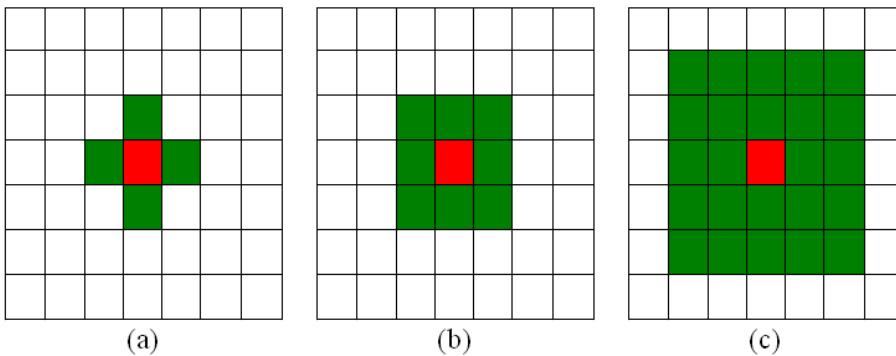


Figura 2.1: Tipos de vecindad: (a) *von Neumann*. (b) *Moore*. (c) *Moore extendida*.

Las celdas del autómata celular cambian su estado de manera discreta en el tiempo. En cada iteración, se comprueban todas las celdas y se aplican las reglas. Dichas reglas se aplican a través de la función de transición  $f$  a cada celda, teniendo en cuenta no sólo su estado, sino también el de su vecindad [82]. Por lo tanto, los autómatas celulares sufren un proceso de evolución a lo largo de la ejecución de las distintas iteraciones, ya que los estados de las celdas cambian durante dicha ejecución. Desde este punto de vista, durante las últimas décadas, los autómatas celulares se han convertido en una técnica muy potente para resolver problemas de simulación.

Los autómatas celulares se han aplicado para resolver una gran variedad de problemas de diversos tipos: un modelo basado en autómatas celulares para crear terapias contra la infección del virus VIH [99], un sistema para comprender el crecimiento urbano [24], un modelo de migración de bancos de peces [95], un modelo basado autómatas celulares aplicado al mundo de los videojuegos [17], un modelo que simula una población infectada por una plaga periódica [37], un sistema basado en autómatas celulares para simular fluidos de lava [100], un modelo de ecosistema de vida artificial [113], un sistema para realizar simulaciones de enfermedades [25], un modelo que simula el flujo de agua en un canal de riego [72], un modelo para la competición y evolución de especies [26], sistemas para el estudio de la enfermedad del cáncer basados en autómatas celulares [102] [50], simulación del tráfico tanto de vehículos en ciudades [3] como de trenes en vías de ferrocarril [103], simulación de avalanchas de nieve [5], un sistema de iluminación modular [13], nuevos sistemas criptográficos basados en autómatas celulares [48] [33] y autómatas celulares cuánticos aplicados tanto a la simulación de campos magnéticos macroscópicos [34] como a la mejora de la eficiencia de algoritmos [59].

### 2.1.2. Definición formal de autómata celular

**Definición 2.1 Autómata celular.** *Un autómata celular es una séxtupla formada por los siguientes elementos  $(d, r, Q, \#, V, f)$ , donde:*

- $d|d > 0$ : *dimensión espacial del autómata celular. La posición de cada celda se expresa mediante un vector  $Z^d$ . Con  $d = 1$  tenemos un autómata unidimensional y con posición de las celdas en  $Z$ ; con  $d = 2$  tenemos un autómata bidimensional y con posición de las celdas en  $Z \times Z$ , etc.*
- $r$ : *índice que marca el tamaño de la vecindad, es decir, el número de vecinos que interactúan con la celda actual.*
- $Q$ : *conjunto de estados que puede tomar cada celda. El conjunto de estados es finito e igual para todas las celdas.*
- $\#$ : *estado de  $Q$  llamado quiescente, que indica la ausencia de actividad.*
- $V$ : *vector de vecindad, que contiene  $r$  elementos distintos de  $Z^d$ .  $V \subset (Z^d)^r$ .*
- $f$ : *función de transición del autómata celular. Toma como argumentos de entrada los estados de la celda actual y de los vecinos, y devuelve un nuevo estado para la celda actual. La función de transición  $f$  utiliza un conjunto de reglas que especifican los procesos de cambio de los estados de las celdas. La función de transición  $f$  se aplica a cada celda de la matriz a través de varias iteraciones, como muestra la siguiente expresión matemática:*

$$f : Q^{r+1} \rightarrow Q$$

$$q_i(t) = f(q_{i-r}(t-1), q_{i-r+1}(t-1), \dots, q_{i+r}(t-1)) \quad (2.1)$$

*siendo  $q_i(t)$  el estado de la celda  $i$  en el tiempo  $t$ .*

**Definición 2.2** *Configuración actual de un autómata celular.* Definimos la configuración actual de un autómata celular  $C$  cualquiera a través de la siguiente expresión matemática:

$$\sigma : Z^d \rightarrow Q \quad (2.2)$$

**Definición 2.3** *Configuración siguiente de un autómata celular.* Definimos la configuración siguiente de un autómata celular en el tiempo  $t$  al resultado de aplicar las reglas del autómata a sus celdas con los estados de la configuración anterior  $t - 1$ , como muestra la siguiente expresión:

$$\sigma_t : f(\sigma_{t-1}) \quad (2.3)$$

Existen algunos trabajos que se han encargado de estudiar el aspecto más teórico de los autómatas celulares, donde se ofrecen, desde un punto de vista matemático, diversos teoremas y corolarios que se cumplen en este modelo [58].

### 2.1.3. Clasificación de los autómatas celulares

*Stephen Wolfram*, basándose en el comportamiento dinámico de los autómatas celulares unidimensionales, los clasificó en cuatro tipos distintos [110] [111]:

- **Tipo 1:** el comportamiento del autómata celular es *predecible*, ya que todas las configuraciones evolucionan hacia un estado estable, independientemente de cual sea la configuración inicial, con lo que los patrones desaparecen.
- **Tipo 2:** el autómata celular tiene un comportamiento *periódico*, ya que evoluciona hacia estructuras periódicas. Si se produce un cambio en un valor de la situación inicial, sólo afecta a una región finita a su alrededor.
- **Tipo 3:** el comportamiento del autómata celular es *caótico*, y pequeños cambios en la configuración inicial provoca evoluciones distintas.
- **Tipo 4:** es el tipo más *impreciso*, donde se engloban sobre todo autómatas celulares que caen en un atractor después de pasar por una amplia fase evolutiva.

## 2.2. APLICACIONES DE LOS AUTÓMATAS CELULARES

En este apartado se muestran las principales aplicaciones de los autómatas celulares unidimensionales así como la notación de *Stephen Wolfram*, las aplicaciones de los autómatas celulares bidimensionales, como por ejemplo el *juego de la vida* de *Conway*, así como las aplicaciones más importantes que podemos encontrar dentro del ámbito específico de la teledetección.



### 2.2.1. Autómatas celulares unidimensionales. *Stephen Wolfram*

Los **autómatas celulares unidimensionales** son aquellos que tienen una sola dimensión espacial, aunque realmente se representan en forma matricial. Cada generación del autómata celular unidimensional está representada por un renglón de la matriz, donde la primera generación corresponde con el primer renglón y las siguientes se ubican de manera correlativa [109]. A lo largo del presente documento, se denominará **imagen-AC** a la representación matricial de las distintas generaciones producidas por un autómata celular unidimensional. A continuación se muestra un ejemplo de autómata celular unidimensional, siguiendo la nomenclatura presentada a lo largo de este capítulo:

$$C = (1, 2, Q, 0, V, f), \text{ donde:}$$

$$Q = \{0, 1\} \text{ y } V = \{-1, 1\}$$

Se trata de un autómata celular con una dimensión espacial unidimensional y 2 vecinos por celda. Cada celda puede tomar dos estados posibles: 0 y 1. El estado *quiescente* es el 0, y la vecindad de cada celda corresponde a la celda de la izquierda y de la derecha. La función de transición  $f$  del autómata celular se muestra a continuación:

$$f(q_{i-1}(t-1), q_i(t-1), q_{i+1}(t-1)) = (q_{i-1}(t-1) + q_{i+1}(t-1)) \bmod 2 \quad (2.4)$$

La *Tabla 2.1* muestra la evolución que experimenta dicho autómata celular unidimensional a través de 4 iteraciones, partiendo del estado inicial  $t=0$ .

t=0	0	0	0	0	1	0	0	0	0
t=1	0	0	0	1	0	1	0	0	0
t=2	0	0	1	0	0	0	1	0	0
t=3	0	1	0	0	0	0	0	1	0
t=4	1	0	0	0	0	0	0	0	1

Tabla 2.1: Evolución del autómata celular unidimensional del ejemplo.

El precursor de los autómatas celulares unidimensionales fue el científico *Stephen Wolfram* [112], creador del software *Mathematica*. Creó una numeración específica para este tipo de autómata celular, donde se representa cada función de transición como una terna o vector formado por el estado actual y los estados de las dos celdas vecinas:

$$\delta_i: (q_h, [e_1, e_2]) \Rightarrow q_t \text{ es equivalente a } \delta_i: ([e_1, q_h, e_2]) \Rightarrow q_t$$

donde  $q_h$  es el estado de la celda actual, y  $e_1, e_2$  son los estados de las celdas adyacentes. A continuación se muestra un ejemplo concreto de función de transición. Si tomamos todas las combinaciones posibles, podemos representar dicha función de transición  $f$  de la siguiente manera:

$$\begin{aligned}
\delta_1: (0, [0, 0]) &\Rightarrow 0 \text{ es equivalente a } \delta_1:(0, 0, 0) \Rightarrow 0 \\
\delta_2: (0, [0, 1]) &\Rightarrow 1 \text{ es equivalente a } \delta_2:(0, 0, 1) \Rightarrow 1 \\
\delta_3: (1, [0, 0]) &\Rightarrow 0 \text{ es equivalente a } \delta_3:(0, 1, 0) \Rightarrow 0 \\
\delta_4: (1, [0, 1]) &\Rightarrow 1 \text{ es equivalente a } \delta_4:(0, 1, 1) \Rightarrow 1 \\
\delta_5: (0, [1, 0]) &\Rightarrow 1 \text{ es equivalente a } \delta_5:(1, 0, 0) \Rightarrow 1 \\
\delta_6: (0, [1, 1]) &\Rightarrow 1 \text{ es equivalente a } \delta_6:(1, 0, 1) \Rightarrow 1 \\
\delta_7: (1, [1, 0]) &\Rightarrow 1 \text{ es equivalente a } \delta_7:(1, 1, 0) \Rightarrow 1 \\
\delta_8: (1, [1, 1]) &\Rightarrow 0 \text{ es equivalente a } \delta_8:(1, 1, 1) \Rightarrow 0
\end{aligned}$$

El vector generado por cada combinación anterior se puede expresar como un número binario de 3 dígitos, y podemos asignar a cada valor binario su estado final. A continuación podemos crear un vector de estados finales ordenados y expresarlo en sistema de numeración decimal, de tal forma que el ejemplo anterior quedaría expresado de la siguiente manera:

$$R = [0,1,0,1,1,1,0] \text{ (notación binaria)} \Rightarrow R = 94 \text{ (notación decimal)}$$

Por lo tanto, según la notación de *Stephen Wolfram*, la función de transición  $f$  del ejemplo anterior se puede resumir con la regla  $R94$  obtenida al pasar a decimal el número binario formado por los estados finales ordenados, producidos por las 8 combinaciones anteriormente realizadas. Como partimos de un número formado por 8 dígitos binarios, podremos crear un total de 256 reglas básicas. Cuando aplicamos una regla específica a un renglón de estados iniciales, va surgiendo un nuevo renglón en cada iteración, que unidos forman la matriz de generaciones a la que hemos llamado imagen-AC, es decir, se forma una imagen aplicando un autómata celular unidimensional a un estado inicial. La *Figura 2.2* muestra la imagen-AC obtenida mediante la regla  $R94$  tras aplicar 15 iteraciones. La *Figura 2.3* muestra la imagen-AC creada por otra regla de *Stephen Wolfram*, en este caso la regla  $R54$ .

Estas reglas, además, pueden expresarse mediante expresiones matemáticas. Si comenzamos con una sola celda negra en la regla  $R94$ , podemos codificar en binario las generaciones sucesivas y comprobamos que forman el siguiente patrón: 1, 111, 11011, 1110111, 110101011, etc, que en sistema de numeración decimal corresponde al patrón: 1, 7, 27, 119, 427, ... Dicha sucesión numérica puede expresarse, desde un punto de vista matemático, a través de la siguiente función:

$$a(n) = \begin{cases} 1 & \text{si } n = 0 \\ 7 & \text{si } n = 1 \\ \frac{1}{6}(10 + 11 \cdot 4^n) & \text{si } n > 1 \text{ es impar} \\ \frac{1}{3}(1 + 5 \cdot 4^n) & \text{si } n > 0 \text{ es par} \end{cases} \quad (2.5)$$

También podemos comprobar, en la siguiente expresión, que dicha regla sigue un patrón numérico decimal en la generación de nuevos estados:

$$a(n) = \begin{cases} \frac{7}{15}(4^{n+1} - 1) & \text{si } n \text{ es impar} \\ \frac{1}{15}(4^{n+2} - 1) & \text{si } n \text{ es par} \end{cases} \quad (2.6)$$

**Regla 94**

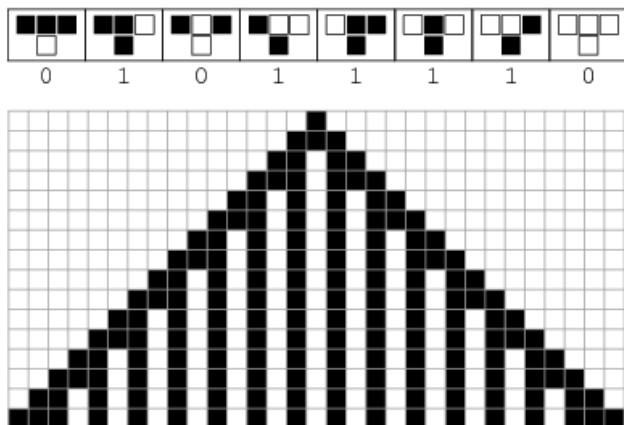


Figura 2.2: Imagen-AC generada por la regla *R94*.

**Regla 54**

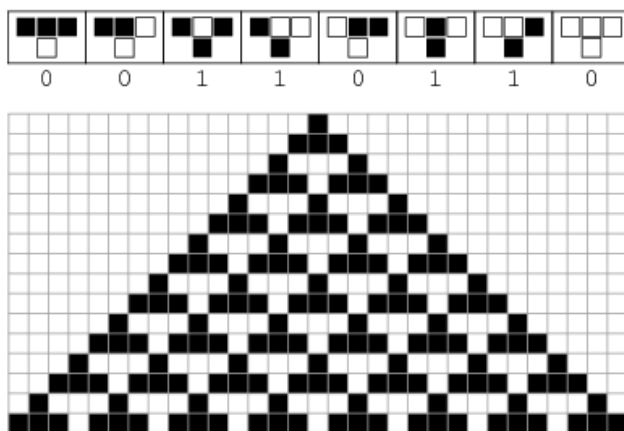


Figura 2.3: Imagen-AC generada por la regla *R54*.

La *Figura 2.4* muestra las reglas más conocidas con la notación expuesta de *Stephen Wolfram*, así como los patrones que crea cada regla concreta. La *Figura 2.5* muestra las imágenes-AC generadas por dichas reglas de *Stephen Wolfram* tras aplicar 15 iteraciones al autómata celular unidimensional.

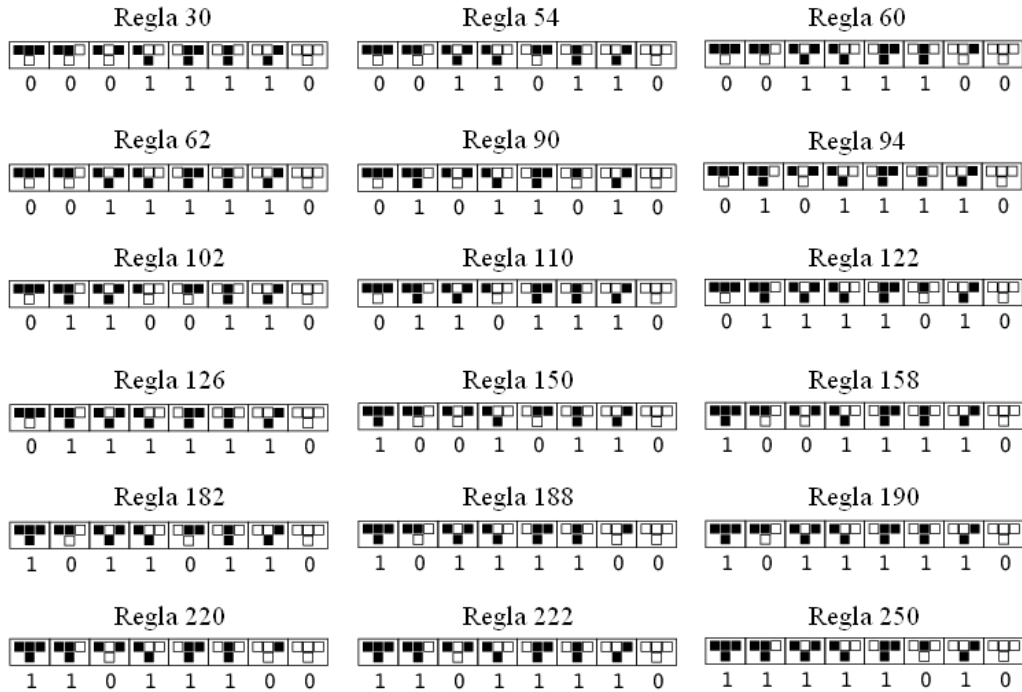


Figura 2.4: Reglas que generan las imágenes-AC más conocidas.

### 2.2.2. Autómatas celulares bidimensionales. *Juego de la vida*

Los **autómatas celulares bidimensionales** son aquellos en los que las celdas se sitúan en un universo plano de dos dimensiones, es decir, una cuadrícula bidimensional, de tal modo que cada celda tiene 8 vecinos adyacentes. Una de las aplicaciones más conocidas de los autómatas celulares bidimensionales es el llamado **juego de la vida** diseñado por el matemático *John Horton Conway* en 1970, y publicado en la columna de juegos matemáticos de la revista *Scientific American* por *Martin Gardner* [49]. Dicha publicación fue de gran interés para diversos campos de la ciencia, sobre todo en el ámbito de las matemáticas y la economía, debido a la gran variabilidad que ofrece el algoritmo en la evolución de los patrones. Como su propio nombre indica, el *juego de la vida* es un algoritmo basado en un autómata celular bidimensional que simula el proceso mismo de la vida, donde cada celda de la matriz puede realizar las acciones fundamentales de cualquier ser vivo.

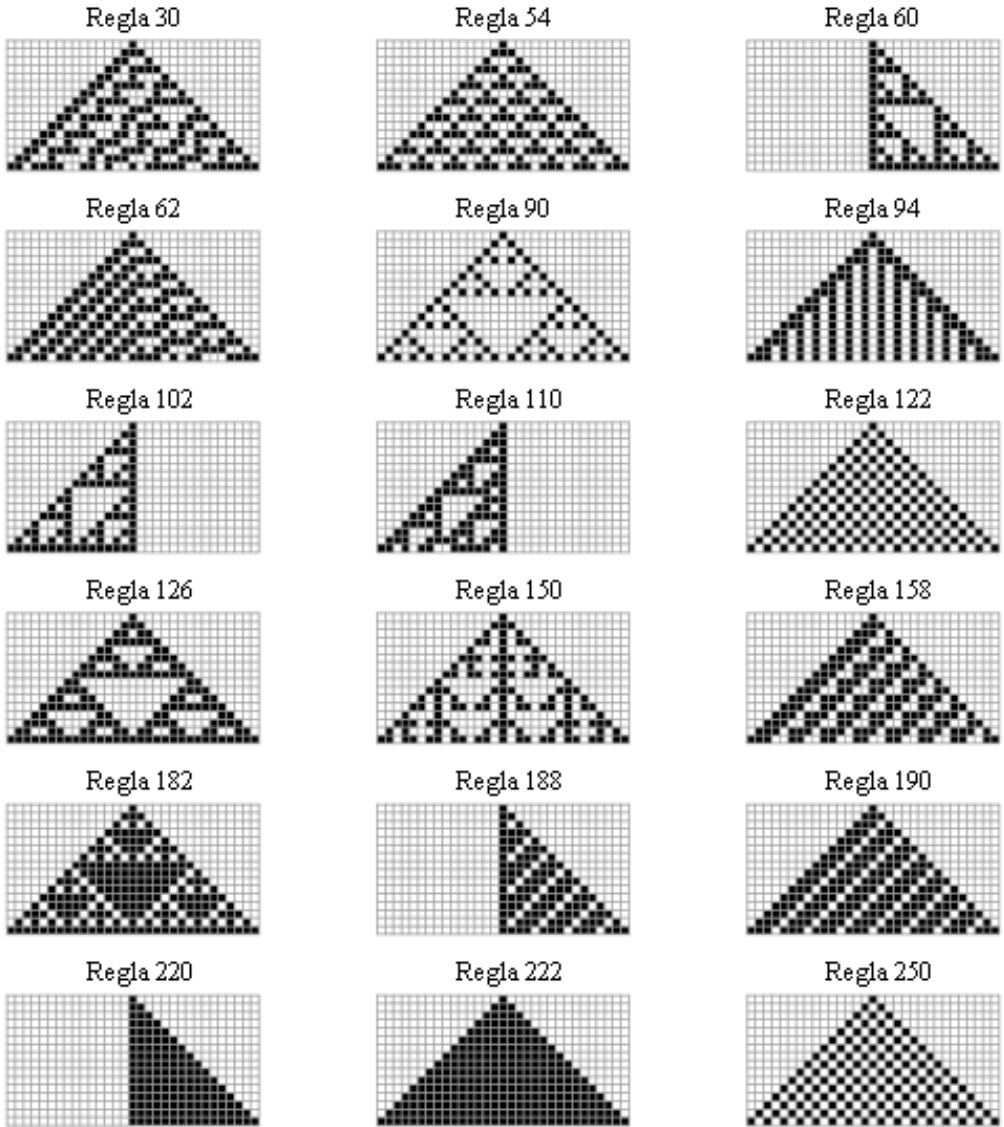


Figura 2.5: Imágenes-AC más conocidas de *Stephen Wolfram*.

Cada celda corresponde a un ser vivo que, como tal, tiene la posibilidad de nacer, reproducirse, mantenerse con vida y morir por soledad o superpoblación. La definición del autómata celular que implementa el *juego de la vida* de *Conway* es la siguiente:

$$C = (2, 8, Q, 0, V, f), \text{ donde:}$$

$$Q = \{0, 1\} \text{ y } V = \{(-1, -1), (0, -1), (1, -1), (-1, 0), (1, 0), (-1, 1), (0, 1), (1, 1)\}$$

Se trata de un autómata celular con una dimensión espacial bidimensional y 8 vecinos por celda. Cada celda puede tomar 2 estados posibles: 0 y 1, que corresponde con los estados *muerta* y *viva*, respectivamente. El estado *quiescente* es el 0, y la vecindad de cada celda es de tipo *Moore*. La función de transición  $f$  del autómata celular del *juego de la vida* de *Conway* se muestra a continuación:

$$f(q_0(t-1), q_1(t-1), \dots, q_8(t-1)) = \begin{cases} q_0 & \text{si } \sum_{i=1}^8 q_i(t-1) = 2 \\ 1 & \text{si } \sum_{i=1}^8 q_i(t-1) = 3 \\ 0 & \end{cases} \quad (2.7)$$

Como se puede observar en la primera opción de la función de transición  $f$  del autómata celular, cuando una celda posee 2 vecinos con vida, ésta permanece en el mismo estado en el que se encontraba (proceso de *estabilidad*). La segunda opción de la función de transición  $f$  indica que, cuando una celda está rodeada por 3 celdas vivas, nace (proceso de *reproducción*). La tercera opción indica que, en cualquier otro caso, la celda muere (procesos de *soledad* y *superpoblación*). La *Figura 2.6* muestra un ejemplo de cada tipo de proceso del *juego de la vida*.

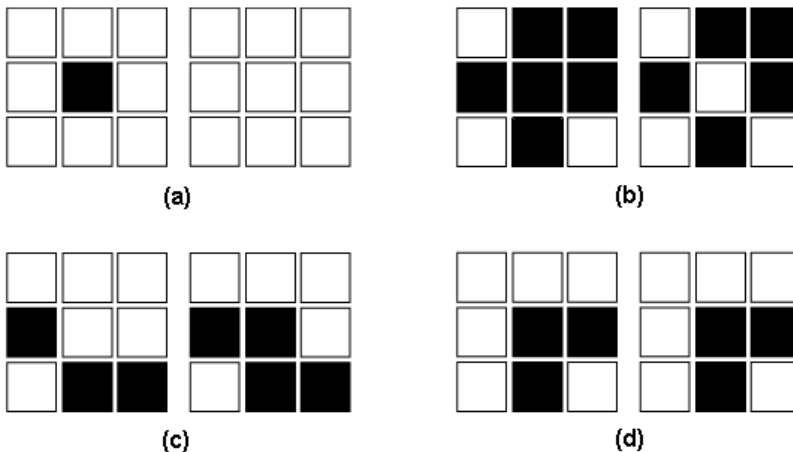


Figura 2.6: (a) *Soledad*: una celda viva con menos de 2 celdas vivas adyacentes muere. (b) *Sobreproducción*: una celda viva con más de 3 celdas vivas adyacentes también muere. (c) *Reproducción*: una celda muerta con 3 celdas vivas adyacentes nace. (d) *Estabilidad*: una celda viva con dos celdas vivas adyacentes permanece viva.

La Figura 2.7 muestra una simulación del *juego de la vida* de Conway a través de 75 iteraciones del autómata celular.

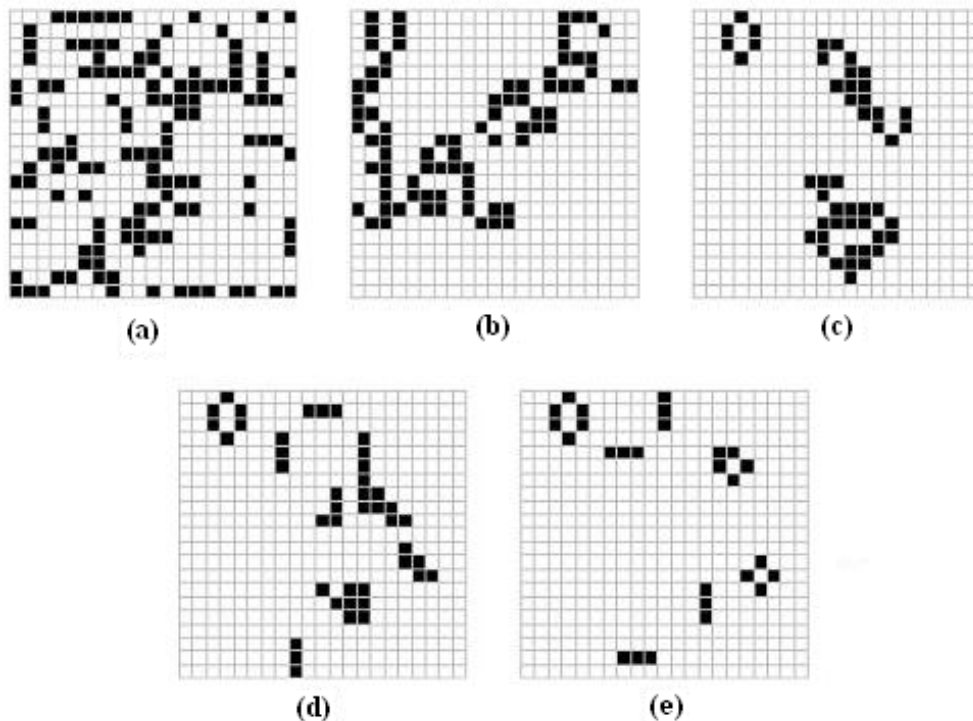


Figura 2.7: (a) Condición inicial. (b) Iteración 10. (c) Iteración 25. (d) Iteración 50. (e) Iteración 75.

### 2.2.3. Autómatas celulares aplicados a la teledetección

Cuando aplicamos los autómatas celulares a las imágenes de satélite, consideramos que cada píxel de la imagen es una celda de un autómata celular bidimensional con una distribución espacial en forma de matriz, y la vecindad de cada píxel o celda del autómata suele estar formada por los 8 píxeles vecinos (vecindad de *Moore*), aunque dicha vecindad puede variar dependiendo del objetivo que se pretende conseguir y puede estar formada por sólo 4 píxeles vecinos (vecindad de *von Neumann*), o incluso los 24 píxeles de alrededor (vecindad de *Moore* extendida).

Los cambios en los estados de las celdas del autómata celular (píxeles de la imagen) ocurren de forma discreta en el tiempo. En cada iteración del autómata celular se comprueba, para cada celda, su estado y los estados de su vecindad, para posteriormente aplicar las reglas de la función de transición  $f$  y obtener de esta forma el nuevo estado de cada celda del autómata.

Por lo tanto, los autómatas celulares tienen un proceso de evolución, ya que todas las celdas están cambiando continuamente de estado a través de la función de transición  $f$ , a lo largo de las distintas iteraciones. Desde este punto de vista, los autómatas celulares se han convertido en las últimas décadas en una herramienta de gran potencia en el ámbito de la teledetección, y han sido utilizados sobre todo en aplicaciones que implican desarrollar algún tipo de simulación en imágenes de satélite, como pueden ser las simulaciones medioambientales, la resolución de fenómenos sociales complejos y las aplicaciones variadas relacionadas con el ámbito de la visión artificial.

En el ámbito **medioambiental** se han realizado numerosos trabajos combinando la teledetección con la técnica de los autómatas celulares: simulaciones avanzadas del proceso de derretimiento de la nieve con influencia de varios factores externos [64], modelización dinámica de sistemas de vegetación [12], estudio de la propagación de brotes de cólera mediante las mediciones indirectas del terreno [68], simulaciones de propagación de incendios forestales [78] [79] [57], modelado de procesos de deforestación con elevada biocomplejidad [77], simulaciones dinámicas destinadas al control del empleo de la tierra [63] [74] o animaciones 3D de nubes [60].

También se han realizado varios trabajos aplicando las propiedades que ofrecen los autómatas celulares a la teledetección para modelar distintos **fenómenos sociales** de elevada complejidad [106], como por ejemplo el control del ruido del tráfico de una ciudad [86], o incluso técnicas avanzadas para la generación de mallas tanto en el diseño urbano como en el arquitectónico [22].

Dentro del ámbito de la **visión artificial**, los autómatas celulares se han aplicado para la obtención de mejoras en imágenes, implementando filtros de reducción de ruido y detección de bordes [81].

## 2.3. SIMULACIÓN DE PRECIPITACIONES MEDIANTE RACA

En este apartado se ofrece una introducción al modelado y la simulación de precipitaciones, se exponen los objetivos propuestos para este método de simulación basado en autómatas celulares, se realiza una descripción del algoritmo RACA, y se muestran las características experimentales y los casos de estudio sobre los que se ha aplicado.

### 2.3.1. Modelado y simulación de precipitaciones

Las características físicas de los *fenómenos meteorológicos* han sido ampliamente modeladas y simuladas por la comunidad científica de todo el mundo. Una investigación profunda de la naturaleza de dichos procesos nos permite conocer, y sobre todo predecir, los efectos que se producen sobre la superficie terrestre. En la actualidad, y gracias a dichas predicciones, nos podemos anticipar a los efectos de los fenómenos meteorológicos, antes de que ocurran. De este modo, podemos conocer las futuras consecuencias del cambio climático, advertir a la población antes del paso de un huracán con el objetivo de salvar el máximo número posible de vidas humanas, o utilizar las gráficas de precipitaciones históricas de una región para realizar una previsión de futuras inundaciones sobre un terreno poblado determinado.



Uno de los fenómenos meteorológicos más estudiados a lo largo de la historia ha sido el proceso de las *precipitaciones*, y su modelado virtual es un factor clave a la hora de estimar los diferentes cambios medioambientales producidos por dicho fenómeno: variación de la cantidad de agua remanente en un pantano, formación de nuevos ríos, erosión del suelo, etc. El modelado matemático de sus propiedades físicas es un proceso bastante complejo, ya que influyen una enorme cantidad de variables que complican mucho la implementación de una simulación totalmente fiel a la realidad. Sin embargo, se pueden realizar aproximaciones bastante precisas. Existen muchos estudios que han investigado el fenómeno de las precipitaciones, utilizando un gran diversidad de metodologías relacionadas con diversos ámbitos de la ciencia.

Como hemos visto a lo largo de este capítulo, los autómatas celulares suelen utilizarse para resolver problemas relacionados con los sistemas complejos [90]. Además, han sido ampliamente utilizados en el ámbito de la teledetección para implementar procesos de simulación de fenómenos meteorológicos y medioambientales en imágenes de satélite 2D. Sin embargo, existen pocos estudios que utilicen las propiedades de los autómatas celulares sobre imágenes de satélite *DEM* (*Digital Elevation Model*), que almacenan en cada píxel la altura correspondiente a dicha superficie, y por lo tanto nos ofrecen la posibilidad de representar de manera tridimensional el modelado de la simulación realizada sobre la imagen de satélite.

Entre los trabajos de investigación que están relacionados con el propósito del algoritmo RACA, y a su vez están basados en los autómatas celulares, podemos citar los siguientes ejemplos: un modelo que simula la erosión que provoca el agua en el suelo [32], un algoritmo que simula el flujo del agua [91], un modelo que simula el proceso de las inundaciones [35], y un algoritmo que analiza el drenaje y la asimilación de los datos de las precipitaciones [29]. Todos estos trabajos de investigación obtienen unos resultados muy realistas, ya que utilizan un gran número de estados y reglas para su autómata celular correspondiente. Sin embargo, debido a la complejidad de los autómatas celulares que utilizan, tienen el inconveniente de que es necesario conocer previamente una gran cantidad de parámetros para configurar su funcionamiento y, por lo tanto, es obligatorio realizar un estudio preliminar de la zona de estudio. En muchos casos, no existe suficiente información para llevar a cabo la simulación, o es muy costoso en tiempo conseguir dicha información. Sin embargo, el algoritmo RACA utiliza sólo dos estados y reglas en su autómata celular con el objetivo de simplificar el proceso de simulación, implementando sólo la fuerza de la gravedad, y proporcionando resultados numéricos y gráficos 3D del nivel final de agua acumulada. Por lo tanto, antes de la aplicación del algoritmo RACA *no es necesario realizar un estudio preliminar* de la zona de estudio y, aunque sus resultados no tendrán un acierto del 100 %, sí serán bastante aproximados, y se obtendrán con un *coste computacional bajo*.

Las siguientes secciones presentan una descripción detallada del algoritmo RACA, implementado en el marco del proyecto *SOLERES*, en el que se ha aplicado la metodología de los autómatas celulares para modelar y simular el proceso meteorológico de las precipitaciones sobre una imagen de satélite *DEM* ofrecida por la *NASA*, mostrando resultados hidrológicos tanto numéricos como tridimensionales del caudal de agua remanente de la zona de estudio.

### 2.3.2. Objetivos propuestos

Los principales objetivos que se pretenden cubrir con el desarrollo del algoritmo RACA son los siguientes:

- **Objetivo #1. Prevención de desastres naturales en regiones urbanas:** se puede calcular el índice de probabilidad de desastres provocados por riadas e inundaciones en zonas urbanas, teniendo en cuenta las gráficas de precipitaciones registradas históricamente en dicha región. En los resultados visuales ofrecidos por el algoritmo RACA se puede observar claramente qué regiones de la imagen *DEM* estarán cubiertas por el agua tras realizar la simulaciones correspondientes. Por lo tanto, si una zona que ya ha sido urbanizada aparece cubierta de agua utilizando una gráfica de precipitaciones registrada históricamente en dicha zona, el riesgo de inundación es muy elevado y por lo tanto se deben poner medios para evitar pérdidas humanas.
- **Objetivo #2. Proyección de futuros proyectos de urbanismo:** se pueden establecer regiones urbanas seguras en torno a zonas geográficas con riesgo de inundación. Teniendo en cuenta todas las gráficas de precipitaciones registradas históricamente en una región determinada, el algoritmo RACA puede calcular cuál fue la cota máxima que alcanzó el agua en los distintos puntos de la imagen de satélite tridimensional. Teniendo en cuenta los resultados obtenidos, nunca se deberán crear zonas urbanas en cotas inferiores a dicha altura de riesgo, para evitar futuras pérdidas de vidas humanas.
- **Objetivo #3. Estimación del suministro de agua de una población:** teniendo en cuenta las gráficas de precipitaciones correspondientes al tipo de clima de una zona determinada, se pueden realizar cálculos muy aproximados de la estimación del caudal de agua acumulado en una región que posee un pantano artificial. El algoritmo RACA también considera la cantidad inicial de agua que pudiera tener dicho pantano (condiciones iniciales antes de la simulación). Además, también se pueden calcular las zonas geográficas más indicadas para la construcción del muro de una presa, con el objetivo de optimizar el proceso de acumulación de agua con una cantidad mínima de muro construido, y de este modo ahorrar material de construcción.

### 2.3.3. Algoritmo RACA

El algoritmo *RACA* (*Rainfall Algorithm with Cellular Automata*) [42] toma como argumentos de entrada una o varias gráficas de precipitaciones medidas en litros por metro cuadrado de distintos momentos puntuales, que pueden ser días, semanas, meses, años, décadas, etc. Posteriormente se aplican las distintas reglas establecidas del autómata celular sobre la imagen de satélite *DEM*, realizando una simulación completa para cada una de las muestras de precipitaciones, y empleando para cada simulación tantas iteraciones como sean necesarias para que el agua descienda y se acumule en los niveles menos elevados de la superficie de la imagen.

La presentación de resultados no sólo se realiza después de cada iteración del autómata celular, sino también al final del proceso completo. Por lo tanto, podemos saber en cada momento qué cantidad de caudal de agua remanente existe en dicha región de la superficie terrestre. Las consideraciones previas que se deben tener en cuenta antes de aplicar el algoritmo RACA son las siguientes:

- **Condiciones iniciales:** antes de aplicar la simulación 3D, se parte de unas condiciones iniciales (*caudal de agua* ya existente en el pantano/río, o región montañosa).
- **Gráficas de precipitaciones:** para llevar a cabo la simulación es necesario introducir el volumen de las *precipitaciones* medido en litros por metro cuadrado para cada unidad de tiempo correspondiente (día, semana, mes, año, etc).
- **Estados del autómata celular:** cada píxel de la imagen *DEM* corresponde a una celda del autómata celular, y cada celda del autómata celular tiene dos estados: el primero es la altitud original a la que se encuentra el píxel correspondiente de la imagen *DEM* (*estado #1*), y el segundo es el nivel al que llega el agua acumulada por encima (*estado #2*). La altitud total de un píxel será la suma de ambos estados, es decir, la altitud del terreno más el nivel de agua.

Para cada una de las muestras de las distintas gráficas de precipitaciones introducidas al algoritmo RACA, caerá sobre cada píxel de la imagen *DEM* el volumen de litros por metro cuadrado establecido, y se realizará una simulación completa del descenso y acumulación del agua de la lluvia en los niveles menos elevados de la imagen. La función de transición  $f$ , a través de las reglas del autómata celular, deberá ir pasando el volumen correspondiente de agua de cada celda a su celda vecina con menor elevación, teniendo en cuenta la altura total (*estado #1* + *estado #2*), ya que por pura física, el agua siempre tiende a descender.

Cuando el agua vaya llegando a los píxeles menos elevados, que poseen celdas vecinas con elevación mayor o igual a la suya propia, va aumentando el caudal sobre dicha superficie: altitud del píxel (*estado #1*) más nivel de agua acumulada sobre el mismo (*estado #2*). Este paso se irá repitiendo en cada iteración del autómata celular hasta que todo el agua de la imagen se quede estancada y deje de desplazarse entre las celdas, momento en que termina la simulación del argumento de entrada actual y comienza la simulación de la siguiente muestra de precipitaciones. Para finalizar, se muestra el caudal de agua acumulada después de cada iteración del autómata celular, así como el total tras el proceso completo de simulación. El caudal de agua que existe sobre un determinado píxel de la imagen *DEM* se calcula multiplicando el nivel del agua por 1000, ya que, como cada píxel de la imagen *DEM* sobre la que vamos a aplicar el algoritmo RACA tiene una resolución espacial de  $1\text{m}^2$ , entonces cada cubo tridimensional de la representación virtual 3D tiene una resolución de  $1\text{m}^3$ , y cada metro cúbico tiene un volumen de 1000 litros. Para calcular el caudal total de la zona geográfica se suma el caudal almacenado sobre cada uno de los píxeles de la imagen *DEM*. Dicho caudal se muestra tanto de manera numérica como en gráficos virtuales 3D.

El **autómata celular** multiestado del algoritmo RACA se puede expresar, desde un punto de vista matemático y utilizando la misma nomenclatura que la usada a lo largo de este capítulo, a través de la siguiente expresión:

$$\mathbf{RACA} = (d, r, Q, \#, V, f) = \quad (2.8)$$

$$(3, \{4|8\}, [q_{alt\_terreno}, q_{nivel\_agua}], [alt\_DEM, agua\_inicial], V, f)$$

donde:

- $d = 3$ : la dimensión espacial del autómata celular del algoritmo RACA es 3, es decir, las celdas están distribuidas de manera tridimensional.
- $r = \{4|8\}$ : el tamaño de la vecindad puede ser la vecindad de *von Neumann* o la vecindad de *Moore*.
- $Q = [q_{alt\_terreno}, q_{nivel\_agua}]$ : cada celda tiene un conjunto de dos estados, la altitud del terreno y el nivel del agua.
- $\# = [alt\_DEM, agua\_inicial]$ : el estado *quiescente*, o valor inicial, es *alt\\_DEM* para  $q_{alt\_terreno}$  y *agua\\_inicial* para  $q_{nivel\_agua}$ .
- $V$ : el vector de vecindad se puede configurar para 4 u 8 vecinos de cada una de las celdas.
- $f$ : la función de transición aplica las reglas a cada una de las celdas a lo largo de las iteraciones con el propósito de cambiar sus estados, teniendo en cuenta la vecindad elegida, como se muestra en la siguiente expresión:

$$f : Q^{\{4|8\}+1} \rightarrow Q \quad (2.9)$$

La función de transición  $f$  del autómata celular del algoritmo RACA aplica las siguientes reglas en cada iteración:

- **Regla #1.** Si  $flujo\_agua < q_{(i)nivel\_agua}$ :

$$\begin{aligned} q_{(i)nivel\_agua} &= q_{(i)nivel\_agua} - flujo\_agua \\ q_{(j)nivel\_agua} &= q_{(j)nivel\_agua} + flujo\_agua \end{aligned}$$

- **Regla #2.** Si  $flujo\_agua \geq q_{(i)nivel\_agua}$ :

$$\begin{aligned} q_{(i)nivel\_agua} &= 0 \\ q_{(j)nivel\_agua} &= q_{(j)nivel\_agua} + q_{(i)nivel\_agua} \end{aligned}$$

donde  $q_{(i)}$  es la celda actual,  $q_{(j)}$  es la celda vecina con menor altitud (si existen varias vecinas con la menor altitud, se toma una al azar) y  $flujo\_agua$  se define a través de la siguiente expresión:

$$flujo\_agua = \frac{(q_{(i)alt\_terreno} + q_{(i)nivel\_agua}) - (q_{(j)alt\_terreno} + q_{(j)nivel\_agua})}{2} \quad (2.10)$$

La *Tabla 2.2* muestra una descripción del algoritmo RACA creado para la simulación de precipitaciones con autómatas celulares.

**Algoritmo RACA: simulación de precipitaciones con autómatas celulares**

**Entrada:**

*imagenDEM* = imagen DEM  
*condiciones* = condiciones iniciales  
*muestras* = muestras de precipitaciones

**Salida:**

*imagenDEM* = imagen DEM con el agua acumulada  
*resultados* = resultados numéricos

- 01 Partimos de unas condiciones iniciales de acumulación de agua. Asignamos dichas *condiciones* a los píxeles de la *imagenDEM*.
- 02 Para cada *muestra* de la gráfica de precipitaciones, realizar paso 3:
- 03 Se asigna a cada píxel el volumen de litros por metro cuadrado establecido por la *muestra* actual. Hasta que el agua de toda la *imagenDEM* se quede estancada, repetir paso 4:
- 04 Recorrer los píxeles de la *imagenDEM* y asignar el agua a los vecinos con menor altura. Este paso corresponde a las iteraciones del autómata celular.
- 05 Mostrar *resultados* numéricos en formato texto.
- 06 Visualizar resultados 3D de la *imagenDEM*.

Tabla 2.2: Descripción del algoritmo RACA.

La *Figura 2.8* muestra la arquitectura detallada del algoritmo RACA para simulación de precipitaciones con autómatas celulares.

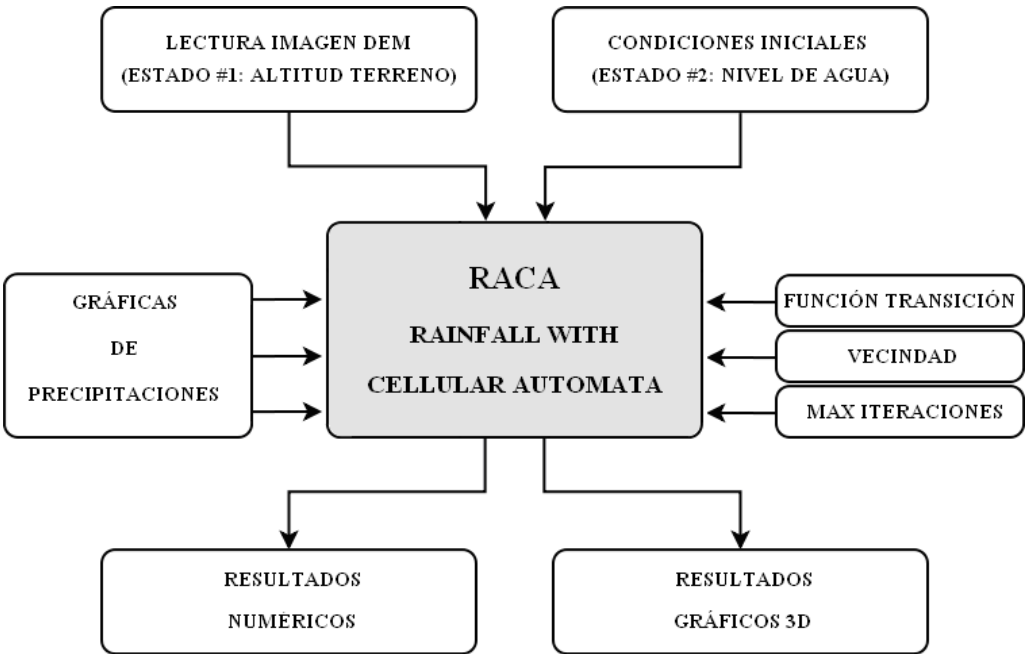


Figura 2.8: Arquitectura del algoritmo RACA.

### 2.3.4. Características experimentales

Esta sección muestra las características experimentales del modelado y simulación, basados en las propiedades que ofrecen los autómatas celulares, del caudal de agua remanente de pantanos y ríos en base al volumen de precipitaciones recibidas en un intervalo de tiempo determinado. Para llevar a cabo este estudio, se ha utilizado una imagen de satélite *DEM* ofrecida por la *NASA*, proporcionada por el *Shuttle Radar Topography Mission (SRTM)*, con el objetivo de tener en cuenta las características topográficas del terreno de la zona de estudio.

La imagen de satélite *DEM*, nombrada *S32W070* por la *NASA*, corresponde a una región montañosa de una zona concreta de Argentina, con las siguientes coordenadas geográficas: sur  $32^\circ$  y oeste  $070^\circ$ . A la imagen original ofrecida por la *NASA* se le ha aplicado un tratamiento previo: ha sido degradada en resolución espacial y corregida de algunos huecos vacíos que ofrecía. El *Apéndice A* muestra una ampliación de la información de la imagen *S32W070*.

La *Figura 2.9* muestra la imagen resultante de tal proceso, con una resolución de  $600 \times 600$  píxeles, donde cada píxel tiene una resolución espacial de  $1\text{m}^2$  y una altitud que oscila entre 20 y 100 metros.

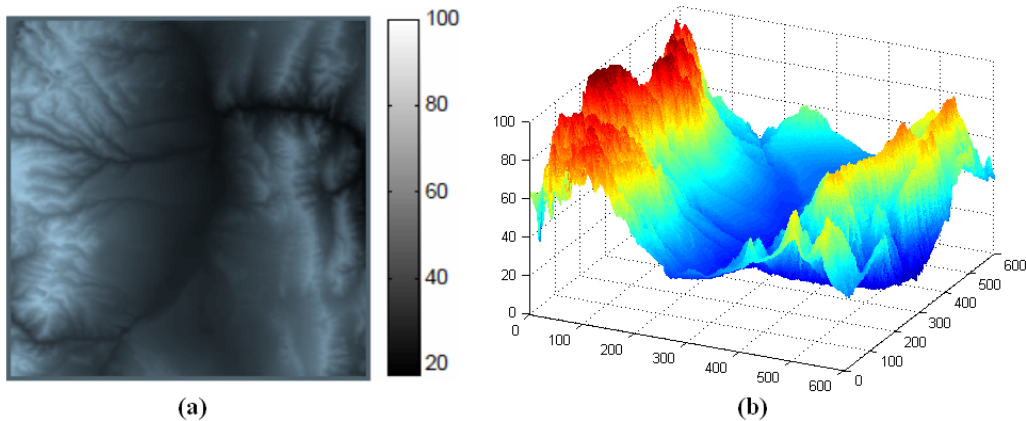


Figura 2.9: Imagen *DEM S32W070* (misión *SRTM* de la *NASA*): (a) Vista en planta. (b) Vista en perspectiva.

Como se puede observar, la *Figura 2.9 (a)* también muestra los valores de elevación en la barra lateral incluida en la imagen, donde los colores negros indican los píxeles con menor elevación y los blancos indican los píxeles con mayor altitud. Como se puede observar en dicha barra lateral, la altura varía entre 20 y 100 metros. Por lo tanto podemos representar, sin ningún tipo de problema, dicha imagen *DEM* de manera tridimensional, con el objetivo de poder visualizarla desde cualquier ángulo concreto.

### 2.3.5. Casos de estudio del algoritmo RACA

Con el objetivo de comparar los resultados ofrecidos por distintos tipos de climas sobre la misma imagen de satélite *DEM*, la simulación de precipitaciones mediante autómatas celulares se ha llevado a cabo utilizando cuatro gráficas de precipitaciones anuales diferentes, cada una correspondiente a un clima distinto. Como la imagen de satélite *DEM* sobre la que se ha experimentado pertenece a una región de Argentina, se han tomado cuatro tipos de climas característicos de distintos lugares del país: *clima árido de sierras y bolsones* (La Rioja), *clima frío húmedo* (Ushuaia, Tierra del Fuego), *clima templado pampeano húmedo* (Ciudad de Buenos Aires) y *clima subtropical sin estación seca* (Posadas, Misiones). La *Figura 2.10* muestra las gráficas de precipitaciones anuales de los cuatro tipos de climas distintos que se han tomado de Argentina.

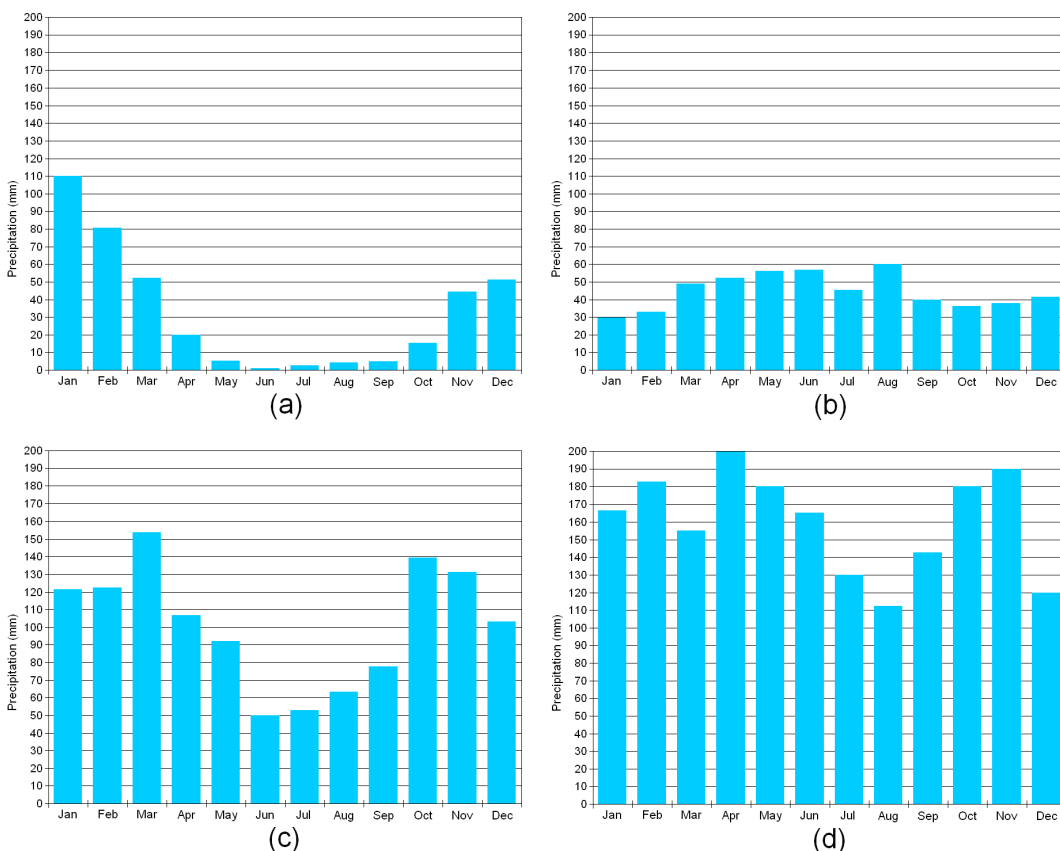


Figura 2.10: Gráficas de precipitaciones anuales de los 4 tipos de climas: (a) Clima árido de sierras y bolsones. (b) Clima frío húmedo. (c) Clima templado pampeano húmedo. (d) Clima subtropical sin estación seca.

Como se puede observar, las gráficas de precipitaciones anuales están ordenadas de manera creciente, tomando como referencia la cantidad total de litros por metro cuadrado anuales, de tal modo que la primera gráfica corresponde al clima más seco, en el que apenas se registran precipitaciones durante los meses de verano, y la última gráfica al clima más húmedo, en el que se registran las mayores precipitaciones durante todos los meses del año. Para cada gráfica de precipitaciones anuales se han realizado *12 simulaciones* (una por cada mes del año). Para cada simulación, se han utilizado tantas iteraciones del autómata celular como han sido necesarias hasta que el agua ha dejado de desplazarse. Como es lógico, los primeros meses, en los que los niveles de agua son inferiores, el número de iteraciones del autómata celular han sido mayores que en los meses consecutivos, en los que los niveles de agua son superiores, y como consecuencia el agua tiene que descender una menor altura por la zona montañosa. Por lo tanto, los resultados que se ofrecen se han obtenido tras realizar un total de *48 simulaciones* (12 por cada tipo de clima). Para simplificar la enorme cantidad de resultados ofrecidos por las 48 simulaciones, se ha tomado como referencia sólo el resultado anual final de cada tipo de clima tras realizar la simulación de todos los meses del año, descartando los resultados individuales de cada mes del año.

Tras realizar las simulaciones, el **clima árido de sierras y bolsones** registra unas precipitaciones anuales totales de 392 litros por metro cuadrado. Con dicha cantidad de precipitaciones anuales, el agua alcanza una cota de altura máxima de 33,19 metros sobre el nivel del mar, o lo que es lo mismo, 13,19 metros sobre el nivel mínimo de la imagen *DEM* (20 metros). El caudal total acumulado para este tipo de clima es de 141 millones de litros.

En segundo lugar, el **clima frío húmedo** registra unas precipitaciones anuales totales de 539 litros por metro cuadrado, alcanzando el agua una cota de altura máxima de 34,35 metros sobre el nivel del mar (14,35 sobre el nivel mínimo de la imagen *DEM*). El caudal total acumulado para este tipo de clima es de 194 millones de litros.

En tercer lugar, el **clima templado pampeano** húmedo registra unas precipitaciones anuales totales de 1231 litros por metro cuadrado (más del doble que el clima anterior), alcanzando el agua una cota de altura máxima de 38,03 metros sobre el nivel del mar (18,03 sobre el nivel mínimo de la imagen *DEM*). El caudal total acumulado para este tipo de clima es de 443 millones de litros. Nótese que, a pesar de que el caudal es más del doble que el del clima anterior, la cota de altura máxima sólo varía en apenas 4 metros, debido a que, a mayor altura en la imagen *DEM*, suele haber menos cantidad de montaña ocupando el espacio, por lo que existe más volumen libre para que sea ocupado por el agua.

Por último, el **clima subtropical sin estación seca** registra unas precipitaciones anuales totales de 1926 litros por metro cuadrado, alcanzando el agua una cota de altura máxima de 40,50 metros sobre el nivel del mar (20,50 sobre el nivel mínimo de la imagen *DEM*). El caudal total acumulado para este tipo de clima es de 693 millones de litros. La *Tabla 2.3* muestra un resumen de los resultados numéricos anuales obtenidos tras realizar las simulaciones en los cuatro tipos de clima, y la *Figura 2.11* muestra los resultados gráficos 3D anuales, donde el agua se representa de color azul marino, las zonas más bajas con colores claros (azul o verde claro) y las zonas más elevadas con colores más oscuros (como naranja o rojo oscuro).



Tipo de clima	Precipitaciones	Cota máxima	Caudal acumulado
(a) Clima árido de sierras y bolsones	392 litros/m <sup>2</sup>	13,19 m	141×10 <sup>6</sup> litros
(b) Clima frío húmedo	539 litros/m <sup>2</sup>	14,35 m	194×10 <sup>6</sup> litros
(c) Clima templado pampeano húmedo	1231 litros/m <sup>2</sup>	18,03 m	443×10 <sup>6</sup> litros
(d) Clima subtropical sin estación seca	1926 litros/m <sup>2</sup>	20,50 m	693×10 <sup>6</sup> litros

Tabla 2.3: Resultados numéricos anuales obtenidos en los 4 tipos de climas.

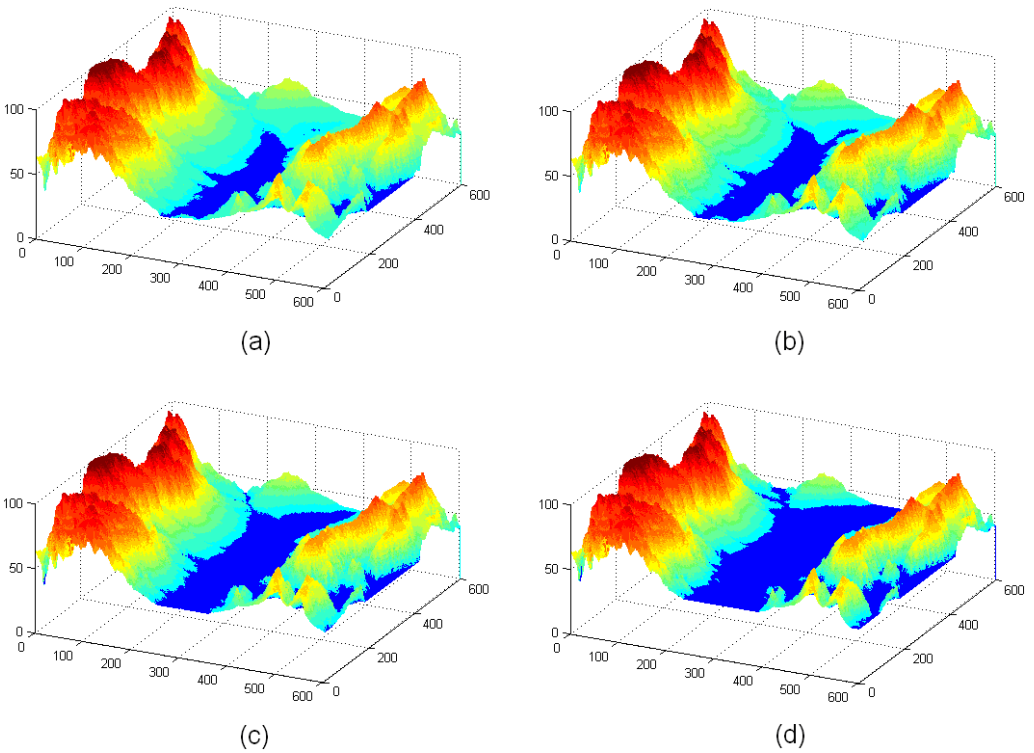


Figura 2.11: Resultados gráficos 3D anuales obtenidos en los 4 tipos de climas: (a) Clima árido de sierras y bolsones. (b) Clima frío húmedo. (c) Clima templado pampeano húmedo. (d) Clima subtropical sin estación seca.

Estos resultados gráficos 3D facilitan el proceso de análisis de los valores obtenidos ya que, en la imagen, podemos comprobar visualmente la diferencia del caudal de agua existente entre el clima árido de sierras y bolsones, el más seco de los cuatro, y el clima subtropical sin estación seca, el más húmedo.

Además, dichas imágenes 3D se pueden configurar para ofrecer la mejor perspectiva posible, modificando la posición de la cámara a través del teclado, no sólo para poder observar la imagen desde el ángulo deseado, sino también para poder navegar libremente por la superficie de la imagen de satélite *DEM*. Gracias a la navegación 3D podemos visualizar de manera sencilla cómo se ha distribuido el caudal de agua acumulado por la superficie de la imagen de satélite *DEM*, analizando con más detalle qué regiones han sido cubiertas por el agua y cuáles no. Esto puede tener multitud de aplicaciones prácticas, como por ejemplo localizar cuál sería el lugar apropiado para situar el muro de una presa con el objetivo de maximizar el caudal de agua acumulado, o comprobar el riesgo de inundación de poblados próximos a pantanos dependiendo de la cantidad de precipitaciones recibidas. La *Figura 2.12* muestra, de manera ampliada, los resultados 3D del clima más húmedo: clima subtropical sin estación seca.

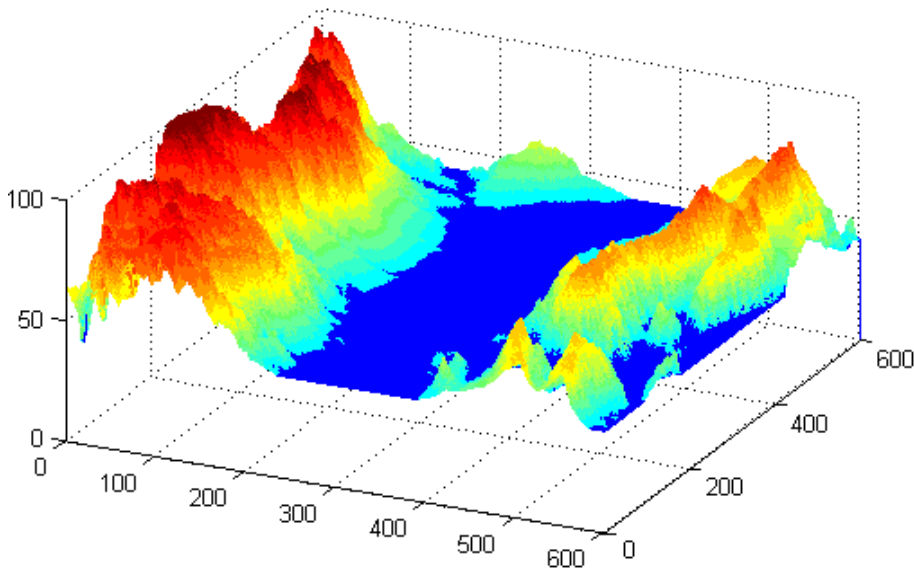


Figura 2.12: Resultados del clima subtropical sin estación seca.

Los principales trabajos futuros que pueden derivar de este trabajo de investigación son los siguientes:

- Añadir a la simulación el proceso de evaporación del agua almacenada en el pantano, teniendo en cuenta las diferentes variables que influyen en dicho proceso.
- Simular el proceso de erosión del terreno tras la caída de lluvia durante un largo intervalo de tiempo, con el objetivo de comprobar cómo modifica el paso del agua el aspecto del paisaje.
- Crear un sistema de posicionamiento óptimo automatizado del muro de una presa que maximice el caudal de agua acumulado tras el proceso de lluvia.

## 2.4. CARACTERIZACIÓN DE TEXTURAS MEDIANTE TACA

En este apartado se explica cómo obtener números decimales a partir de las reglas de *Stephen Wolfram*, se expone cómo usar los números decimales obtenidos para caracterizar imágenes-AC y texturas naturales ampliando dicho procedimiento, y se muestran los casos de estudio sobre los que se ha aplicado.

### 2.4.1. Números decimales de las reglas de *Stephen Wolfram*

La interpretación computerizada de imágenes digitales es una de las ramas más importantes de la informática, y tiene como objetivo la adquisición de información útil a partir de imágenes procedentes de diversos campos de la ciencia, como la medicina, las aplicaciones industriales, el medio ambiente y la teledetección, a través de técnicas de reconocimiento de patrones e inteligencia artificial. La percepción visual de la rugosidad o suavidad de la superficie de los objetos, que es una característica conocida como **textura**, se puede utilizar como información adicional para la caracterización de dichas imágenes digitales. A pesar de la importancia de esta característica, en la actualidad no existe una definición satisfactoria y precisa para la textura [53]. En la actualidad existen numerosos grupos de investigación que están estudiando las características, análisis e interpretación de las texturas de imágenes digitales.

El proceso de detección y análisis de la textura en imágenes digitales puede dividirse en dos grandes grupos: estructurales y estadísticos [52]. Los métodos de análisis *estructurales* asumen que las texturas se componen de primitivas de textura [54] [104]. Los métodos *estadísticos* se basan en valores estadísticos de los niveles de gris de la imagen, y han sido ampliamente estudiados desde la aparición del artículo de *R. Haralick* [53], que propuso varias características basadas en la *matriz de coocurrencia*. Otros métodos más recientes han investigado la *combinación* de técnicas estadísticas y estructurales [104], enfoques basados en modelos relacionados con la transformada *wavelet* [65], los campos aleatorios de *Markov (MRF)* y técnicas *fractales* [66].

Desde el proyecto *SOLERES* se ha implementado el algoritmo TACA para caracterizar diferentes tipos de texturas, basándose en las propiedades de los autómatas celulares lineales. El método propuesto deriva de las propiedades de las imágenes-AC producidas por un modelo de autómata celular unidimensional lineal de 3 celdas, que es uno de los modelos más sencillos en este campo, como hemos visto a lo largo de este capítulo. El objetivo principal de este estudio consiste en mostrar la viabilidad de las técnicas basadas en autómatas celulares para caracterizar texturas naturales. Las pruebas se han realizado con éxito en imágenes binarias proporcionadas por el *Álbum Brodatz* [18].

Como se ha expuesto a lo largo de este capítulo, dadas dos líneas consecutivas de una imagen-AC con valores 0 y 1, la segunda línea toma sus valores a partir de la aplicación de una regla de 3 celdas específica a la primera línea. Generalizando dicho proceso, cada celda con valor 1 de una línea cualquiera se obtiene aplicando la regla de 3 celdas a las tres celdas vecinas de la línea anterior, que a su vez representan un número binario que puede tomar los valores 0-7 al cambiar a base decimal. La *Tabla 2.4* muestra el proceso de adquisición de los números decimales de la regla de 3 celdas en el que se obtienen los números decimales 1, 2, 5 y 6.

Primera línea	1	0	1	0	1	1	1	0	1	1	0	1	0	0	1
Segunda línea	1	1	1	1	0	0	1	1	0	1	1	1	0	1	0
Número de 3-celdas	-	5	2	5	-	-	6	5	-	6	5	2	-	1	-

Tabla 2.4: Adquisición de los números decimales de la regla de 3 celdas.

A partir de dichos valores decimales se deduce que la segunda fila se ha obtenido a partir de la primera mediante la aplicación de una regla lineal de 3 celdas que en la notación binaria tiene un 1 en las posiciones 1, 2, 5 y 6, es decir, la regla es  $01100110_2 = 102_{10}$ , conocida como la regla *R102* de *Stephen Wolfram*. Como se puede observar, el procedimiento no proporciona información acerca de los sitios ubicados en las fronteras de la imagen o cuando existe un 0 presente, sólo presenta información de los sitios con el valor 1. Cabe señalar que, en cierto sentido, el método está relacionado con las propiedades de inversión de los autómatas celulares [101]. De este modo, realizando un proceso similar al de ingeniería inversa en la rama de la programación informática, se puede obtener la regla que describe la generación de una imagen-AC a partir de dos líneas consecutivas de la imagen.

## 2.4.2. Caracterización de texturas de imágenes-AC

Cuando este tipo de análisis se extiende a una imagen-AC completa, tenemos tantas líneas como número de filas tiene la imagen, y para calcular la regla de generación correspondiente es necesario calcular la cantidad de números decimales que se encuentran presentes en la imagen completa. Si aplicamos el método propuesto a la imagen-AC que se muestra en la *Figura 2.13 (a)*, la cantidad de números decimales que se obtiene, o *histograma* correspondiente, es el que se muestra en la *Figura 2.13 (b)*, donde se puede observar que los únicos números decimales de la gráfica son 0, 5, 6 y 7, que en la notación binaria es  $11100001_2 = 225_{10}$ . La *Figura 2.13* muestra la imagen y el histograma de las primitivas de la imagen-AC resultante de la regla *R225* de *Stephen Wolfram*.

El histograma de los números decimales obtenido del procedimiento propuesto juega un papel importante en la caracterización de la textura de una imagen binaria, ya sea una imagen-AC o una textura natural. Ya que el histograma mencionado está asociado al número de primitivas de una imagen-AC, lo vamos a definir como *histograma de primitivas (HP)*. La *Figura 2.14* muestra las *primitivas de textura* correspondientes a la imagen generada por la regla *R102* de *Stephen Wolfram*. El método propuesto también se puede aplicar a una imagen compuesta por varias imágenes-AC, como se muestra en la *Figura 2.15 (izquierda)*. Esta imagen, de tamaño  $256 \times 256$  píxeles, es en realidad la composición de cuatro imágenes-AC de  $128 \times 128$  píxeles cada una, generadas por las reglas *R121* y *R153* en la fila superior, y las reglas *R60* y *R90* en la fila inferior. El análisis de la textura de la imagen compuesta se ha realizado aplicando el método propuesto por bloques de  $16 \times 16$  píxeles, con un total de 256 bloques en la imagen. Cada bloque de salida tiene un nivel de gris asociado a la primitiva de textura presente en el HP. La *Figura 2.15 (derecha)* muestra el resultado de construir una imagen con los bloques resultantes en las respectivas posiciones de la imagen compuesta original.

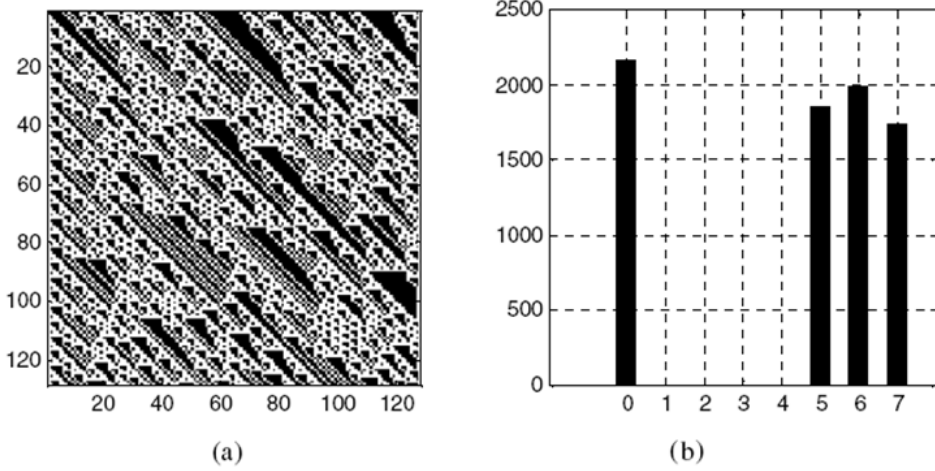


Figura 2.13: (a) Imagen-AC obtenida mediante la regla  $R225$  de *Stephen Wolfram*. (b) Histograma de números decimales obtenidos aplicando el método propuesto.

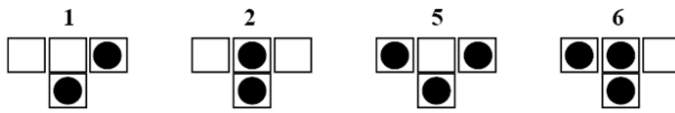


Figura 2.14: Primitivas de textura correspondientes a la imagen-AC generada por la regla  $R102$  de *Stephen Wolfram*.

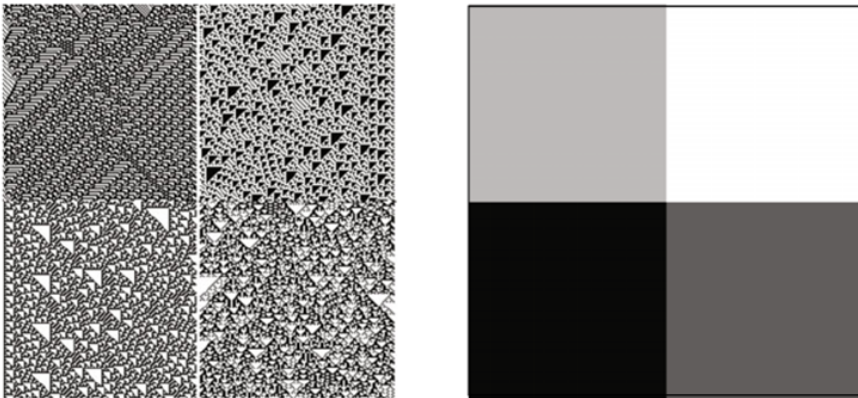


Figura 2.15: (Izquierda) Imagen ( $256 \times 256$ ) compuesta por 4 texturas distintas: sectores superiores ( $R121$  y  $R153$ ), sectores inferiores ( $R60$  y  $R90$ ). (Derecha) Imagen en escala de grises resultante de la aplicación del método propuesto a la imagen original.

### 2.4.3. Caracterización de texturas naturales. Algoritmo TACA

El algoritmo *TACA* (*Texture Algorithm with Cellular Automata*) [62] permite caracterizar texturas naturales. Para aplicar este algoritmo al análisis de texturas naturales vamos a suponer que los patrones de textura mostrados por estas imágenes son el resultado de un proceso similar al que se produce en las imágenes-AC. Sin embargo, con el fin de llevar a cabo el método propuesto a las texturas naturales, tenemos que superar previamente los siguientes problemas:

- En primer lugar, el número de niveles de gris en este tipo de imágenes es normalmente mayor que dos. Este problema puede solucionarse aplicando una *binarización* a la imagen en estudio, con la admisión de una pérdida natural de detalles en la textura de la imagen.
- En segundo lugar, cuando calculamos el *HP* de la imagen binaria correspondiente aparece un número de *primitivas* variado, pero con una frecuencia de ocurrencia dispersa. De hecho, en el *HP* que resulta de estas imágenes no pueden estar presentes las 256 primitivas, por lo tanto, sólo se deben elegir las más importantes de todas las existentes.

Hasta ahora, cuando se ha comentado el cálculo del *HP*, nos hemos referido al histograma de primitivas que resulta al aplicar el método propuesto fila por fila a una textura determinada. A este histograma le vamos a llamar *HP Horizontal (HPH)*. Con el objetivo de tener en cuenta otras primitivas, que no están presentes en el *HPH*, vamos a realizar el mismo proceso en la dirección vertical de la imagen, es decir, columna por columna. A este histograma se le denominará *HP Vertical (HPV)*. Por supuesto, la información adicional proporcionada por el *HPV* mejorará la caracterización de la textura de estudio, pero eso sí, a costa de aumentar considerablemente el número total de opciones posibles para la selección de las mejores primitivas. No obstante, la selección de primitivas horizontales y verticales es la parte más compleja del algoritmo TACA ya que, en la versión actual del algoritmo, dichas primitivas no se seleccionan de manera automática, sino que es el analista experto el que debe elegir las manualmente.

Con estas condiciones, el algoritmo TACA puede caracterizar texturas naturales. Para ello, en primer lugar deberá aplicar un proceso de binarización a la imagen con la textura natural, y suponemos que dicha imagen binaria, representada por ceros y unos, ha sido generada por un autómata celular lineal. A continuación se calcula el *HPH* y se seleccionan las primitivas más importantes, para generar a continuación la imagen-AC correspondiente a la regla de *Stephen Wolfram* obtenida. Después se realiza el mismo procedimiento para el *HPV*. Para finalizar, se genera una imagen que caracteriza la textura natural a partir de la composición de las dos imágenes-AC obtenidas por las reglas de *Stephen Wolfram*. Gracias a la imagen final que se obtiene como resultado, se puede obtener información adicional de la textura natural inicial y relacionarla con las dos reglas generadoras de *Stephen Wolfram*.

La *Figura 2.16* muestra 3 imágenes del *Álbum Brodatz* y sus imágenes binarias correspondientes, y la *Tabla 2.5* muestra una descripción detallada de los distintos pasos del algoritmo TACA, propuesto desde el proyecto *SOLERES* para caracterizar la textura de las imágenes naturales.

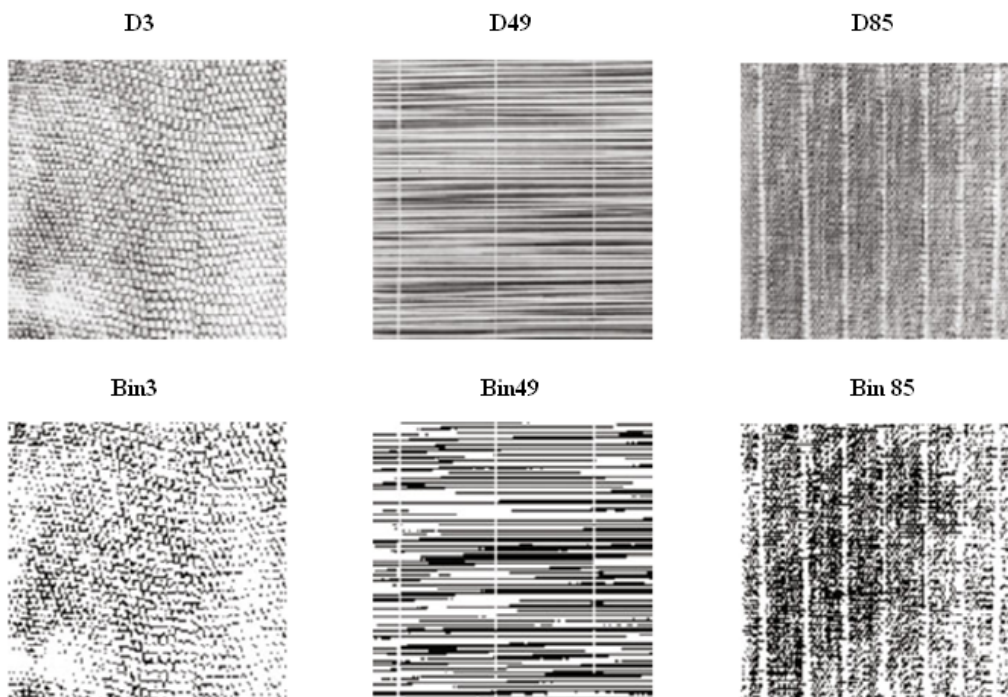


Figura 2.16: (Arriba) Tres texturas naturales del *Álbum Brodatz*:  $D3$ ,  $D49$  y  $D85$ . (Abajo) Imágenes binarias correspondientes.

---

**Algoritmo TACA: caracterización de texturas naturales con autómatas celulares**

---

**Entrada:**

$imagenOriginal$  = imagen con una textura natural

**Salida:**

$imagen - AC$  = imagen generada por reglas de *Stephen Wolfram*

---

- 01 Partimos de una  $imagenOriginal$  con una textura natural.
  - 02 Se aplica un proceso de *binarización* a la imagen con la textura natural. Suponemos que la imagen resultante (representada por ceros y unos) ha sido generada a partir de un autómata celular lineal.
  - 03 Se calcula el  $HPH$  de la imagen binaria a través del procedimiento propuesto.
  - 04 Se seleccionan las *primitivas* más importantes del  $HPH$  (de 1 a 8).
  - 05 Se genera la  $imagen-AC$  correspondiente a la *regla* que surge del paso anterior (características horizontales).
  - 06 Se calcula el  $HPV$  de la imagen binaria a través del procedimiento propuesto.
  - 07 Se seleccionan las *primitivas* más importantes del  $HPV$  (de 1 a 8).
  - 08 Se genera la  $imagen-AC$  correspondiente a la *regla* que surge del paso anterior (características verticales).
  - 09 Se genera una imagen que caracteriza la textura natural de la que partimos en el paso 1. Esta imagen es una *composición* (normalmente la suma) de las imágenes-AC obtenidas en los pasos 5 y 8.
- 

Tabla 2.5: Descripción del algoritmo TACA.



### 2.4.4. Casos de estudio del algoritmo TACA

Como ejemplo ilustrativo se ha aplicado el algoritmo TACA a la versión binaria de la imagen *Brodatz D85*, denotada en la *Figura 2.16* como *bin85*. En la *Figura 2.17 (a)* se muestra el *HPH* obtenido, y en la *Figura 2.17 (b)* el *HPV*.

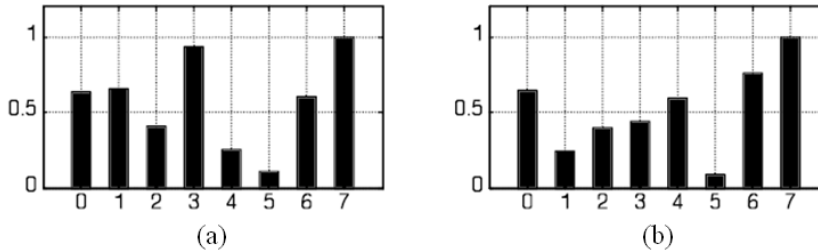


Figura 2.17: Histogramas de primitivas normalizados para la imagen *Brodatz D85*: (a) *HPH* (dirección horizontal). (b) *HPV* (dirección vertical).

Los pasos 4 y 7 del algoritmo TACA son los más difíciles de aplicar ya que, en la etapa actual de la investigación, no se ha encontrado un método práctico para superar este problema. El número de primitivas horizontales y verticales es elegido convenientemente mediante la observación de cada histograma. Una vez elegidas las primitivas más importantes, se obtiene el número de la regla para cada sentido, y es posible generar los *indicadores* horizontal y vertical de la textura original. Estos indicadores de textura se muestran en la *Figura 2.18*.

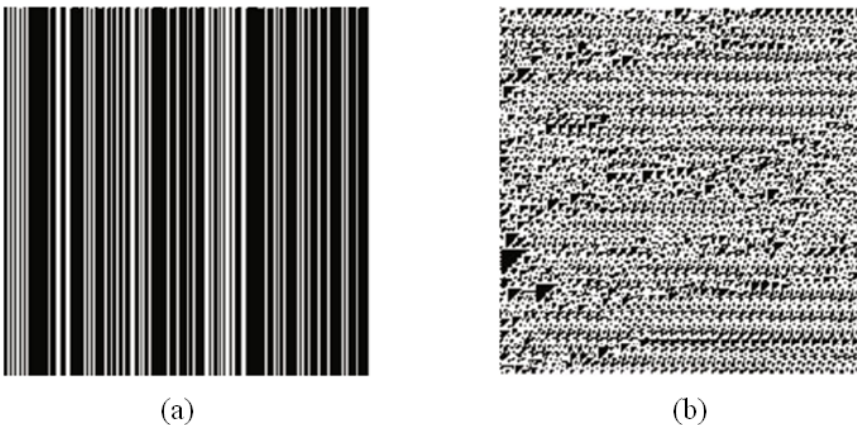


Figura 2.18: Imágenes-AC obtenidas como resultado de elegir las primitivas más importantes de los histogramas anteriores. Del histograma vertical se han tomado 6 primitivas y del horizontal 3. Aplicando el método propuesto se han obtenido las siguientes reglas: (a) *R207* (horizontal). (b) *R193* (vertical).



Las Figuras 2.19, 2.20 y 2.21 muestran los resultados obtenidos en la aplicación del algoritmo TACA a las imágenes de *Brodatz D3*, *D49* y *D85*. La composición que se utilizan en estos ejemplos es una suma de cada componente.

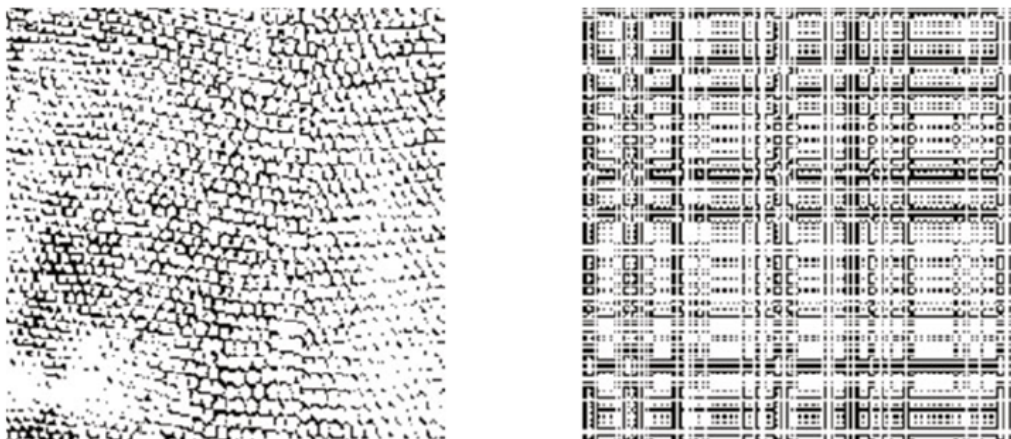


Figura 2.19: (Izquierda) Imagen *bin3* ( $256 \times 256$ ). (Derecha) Aproximación de la imagen *bin3* obtenida por la composición de las imágenes-AC *R95* (H) y *R95* (V), obtenidas usando 6 componentes de cada histograma primitivo (H y V).

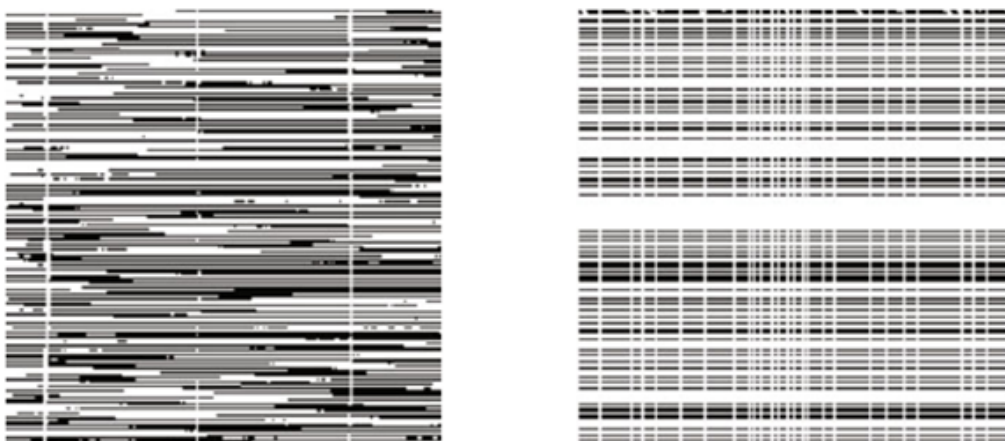


Figura 2.20: (Izquierda) Imagen *bin49* ( $256 \times 256$ ). (Derecha) Aproximación de la imagen *bin49* obtenida por la composición de las imágenes-AC *R217* (H) y *R76* (V), obtenidas usando 5 componentes del *HPH* y 3 componentes del *HPV*.

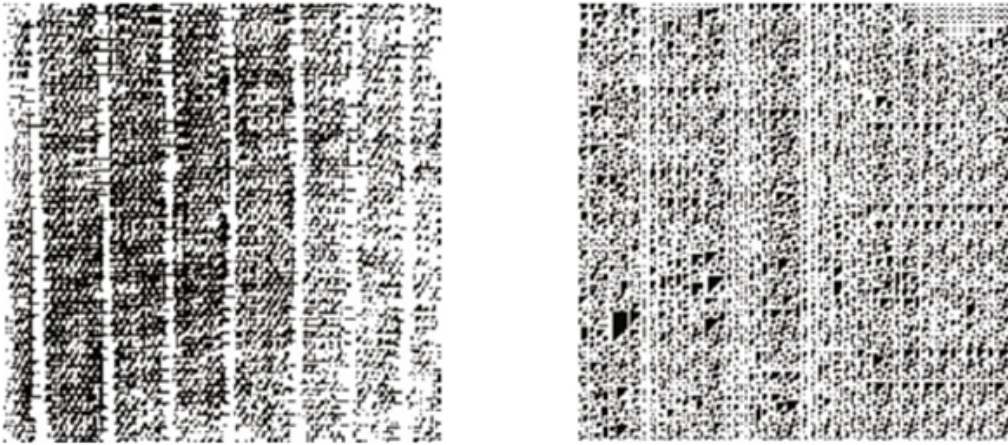


Figura 2.21: (Izquierda) Imagen *bin85* ( $256 \times 256$ ). (Derecha) Aproximación de la imagen *bin85* obtenida por la composición de las imágenes-AC *R207* (H) y *R193* (V), obtenidas usando 6 componentes del *HPH* y 3 componentes del *HPV*.

Como futuros trabajos de este trabajo de investigación se propone, por un lado, mejorar los pasos 4 y 7 del algoritmo TACA con el objetivo de automatizar el proceso de selección de primitivas horizontales y verticales, y por otro lado, extender la aplicación del algoritmo propuesto a texturas naturales no binarias, y por lo tanto más realistas.

## 2.5. RESUMEN Y CONCLUSIONES DEL CAPÍTULO

Los **autómatas celulares** han resultado ser un modelo matemático muy útil para resolver todo tipo de problemas relacionados con los sistemas complejos y los procesos de simulación. En el ámbito concreto de la teledetección, los autómatas celulares se han utilizado sobre todo para realizar simulaciones de fenómenos naturales. Como veremos más adelante, los autómatas celulares permiten solventar todas las limitaciones de los algoritmos *paralelepípedos* y *mínima distancia* descritos a lo largo del *Capítulo 1*. Aunque el trabajo principal de este tesis doctoral consiste en el desarrollo del algoritmo ACA, también se han implementado, desde el marco de trabajo del proyecto *SOLERES*, otros dos algoritmos basados en autómatas celulares: los algoritmos RACA y TACA.

El **algoritmo RACA**, por un lado, tiene como objetivo la simulación de precipitaciones sobre imágenes *DEM*, y nos permite visualizar los resultados obtenidos desde dos puntos de vista: numérico y gráficos 3D. Gracias a este algoritmo, se pueden establecer estudios relacionados con la prevención de desastres naturales en regiones urbanas con probabilidad de inundaciones, proyección de futuros proyectos de urbanismo y estimación del futuro suministro de agua de una población determinada. El algoritmo RACA ha demostrado ser muy útil a la hora de realizar simulaciones de precipitaciones sobre imágenes de satélite *DEM*, ya que *no requiere realizar un estudio preliminar* de la zona de estudio antes de llevar a cabo el proceso de simulación.

Por el contrario, otros algoritmos sí necesitan recopilar una gran cantidad de parámetros de configuración, lo que en ocasiones es demasiado costoso en tiempo o directamente inviable. Por lo tanto, aunque los resultados del algoritmo RACA no son tan precisos como los de otros algoritmos existentes, sí son bastante aproximados, y además con un *coste computacional muy bajo*, debido a que se ha implementado con un autómata celular de estados y reglas simplificadas. El algoritmo RACA ofrece tanto resultados *numéricos* como resultados *gráficos 3D*, con lo que podemos elegir la perspectiva virtual adecuada para centrarnos en regiones concretas de la zona de estudio.

El **algoritmo TACA**, por otro lado, tiene como objetivo la caracterización de texturas tanto de imágenes-AC como de imágenes naturales, y obtiene unos resultados bastante aproximados a las texturas naturales binarias utilizando la *composición* correspondiente de las *imágenes-AC horizontal y vertical*, obtenidas a través de las componentes de cada histograma primitivo. De este modo, podemos caracterizar texturas naturales utilizando un proceso de ingeniería inversa para averiguar las *reglas* de *Stephen Wolfram* que la componen.

No obstante, los algoritmos RACA y TACA son complementarios al trabajo principal implementado durante el desarrollo de esta tesis doctoral, que se centra en la creación de un nuevo algoritmo de clasificación de imágenes de satélite supervisado basado en los autómatas celulares, que tiene como objetivo mejorar los resultados de los clasificadores clásicos desde tres puntos de vista: acierto, flexibilidad y personalización, así como agrupar en un solo proceso las tareas de pre-clasificación, clasificación y post-clasificación. El *Capítulo 3* ofrece una descripción detallada del algoritmo ACA.



---

## CAPÍTULO 3

# CLASIFICACIÓN MEDIANTE AUTÓMATAS CELULARES

---



# Capítulo 3

## CLASIFICACIÓN MEDIANTE AUTÓMATAS CELULARES

### Contenidos

---

<b>3.1.</b>	<b>PROBLEMAS DE LOS ALGORITMOS DE CLASIFICACIÓN . . . . .</b>	<b>101</b>
3.1.1.	Limitaciones de <i>paralelepípedos</i> y <i>mínima distancia</i> . . . . .	102
3.1.2.	Análisis de la tasa de acierto: matriz de confusión . . . . .	104
3.1.3.	Análisis de errores durante el proceso de clasificación . . . . .	105
<b>3.2.</b>	<b>CLASIFICACIÓN DE IMÁGENES DE SATÉLITE CON ACA . . . . .</b>	<b>109</b>
3.2.1.	Trabajo previo existente . . . . .	109
3.2.2.	Objetivos principales del algoritmo ACA . . . . .	110
3.2.3.	Información complementaria de los objetivos principales . . . . .	111
3.2.4.	Consideraciones previas del algoritmo ACA . . . . .	112
<b>3.3.</b>	<b>DESCRIPCIÓN DEL ALGORITMO ACA . . . . .</b>	<b>113</b>
3.3.1.	Estados y reglas del autómata celular . . . . .	113
3.3.2.	Descripción matemática del autómata celular . . . . .	115
3.3.3.	Pseudocódigo del algoritmo ACA . . . . .	116
3.3.4.	Fórmulas complementarias del algoritmo ACA . . . . .	119
3.3.5.	Clasificación jerárquica del algoritmo ACA . . . . .	121
3.3.6.	Arquitectura del algoritmo ACA . . . . .	122
<b>3.4.</b>	<b>HERRAMIENTAS UTILIZADAS PARA LA INVESTIGACIÓN . . . . .</b>	<b>124</b>
3.4.1.	Erdas Imagine . . . . .	124
3.4.2.	Developers Toolkit . . . . .	126
3.4.3.	Weka . . . . .	128
3.4.4.	MATLAB . . . . .	129
<b>3.5.</b>	<b>RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .</b>	<b>130</b>

---





**E**n la actualidad existe una gran cantidad de algoritmos de clasificación de imágenes de satélite, como por ejemplo los clasificadores *paralelepípedos* y *mínima distancia*. Sin embargo, dichos algoritmos presentan una serie de limitaciones, ya que no existe un clasificador que sea 100% eficiente, en términos de tasa de acierto. Desde el marco del proyecto *SOLERES* se ha desarrollado un nuevo algoritmo de clasificación de imágenes de satélite basado en autómatas celulares llamado *ACA* [41] [43] [44] [45] [46], que no sólo mejora la tasa de acierto obtenida por otros algoritmos clásicos, sino que además ofrece una gran cantidad de información útil complementaria. Este capítulo ofrece una descripción detallada del algoritmo *ACA*, y está dividido en los siguientes apartados principales:

- **Problemas de los algoritmos de clasificación:** en este apartado se muestra una descripción de las limitaciones de los algoritmos de clasificación *paralelepípedos* y *mínima distancia*, se realiza el cálculo de la tasa de acierto a través de un ejemplo práctico, y se analizan los errores cometidos durante la clasificación.
- **Clasificación de imágenes de satélite con ACA:** en este apartado se cita el trabajo previo existente relacionado con la clasificación de imágenes de satélite mediante autómatas celulares, se enumeran los objetivos principales del algoritmo *ACA* así como la información complementaria de cada objetivo, y se exponen las consideraciones previas a tener en cuenta antes de aplicar dicho algoritmo.
- **Descripción del algoritmo ACA:** en este apartado se muestran los estados y las reglas del autómata celular utilizado en el algoritmo *ACA* así como su descripción matemática, el pseudocódigo de los algoritmos *ACA principal*, *ACA espectral* y *ACA contextual*, las fórmulas complementarias utilizadas, una explicación detallada de la clasificación jerárquica y la arquitectura principal del algoritmo *ACA*.
- **Herramientas utilizadas para la investigación:** en este apartado se describen las herramientas más importantes utilizadas para desarrollar el trabajo de esta tesis doctoral. Por un lado el programa de tratamiento de imágenes de satélite *Erda's Imagine* y su módulo *Developers Toolkit*, y por otro lado *Weka* y *MATLAB*.
- **Resumen y conclusiones del capítulo:** en este apartado se destacan los aspectos más importantes del algoritmo *ACA*.

### 3.1. PROBLEMAS DE LOS ALGORITMOS DE CLASIFICACIÓN

En este apartado se muestra una descripción de las limitaciones de los algoritmos de clasificación *paralelepípedos* y *mínima distancia*, se realiza el cálculo de la tasa de acierto a través de un ejemplo práctico, y se analizan los errores cometidos durante la clasificación.

### 3.1.1. Limitaciones de *paralelepíedos* y *mínima distancia*

En el *Capítulo 1* se detallaban las características fundamentales de algunos de los algoritmos de clasificación de imágenes de satélite clásicos más conocidos en el ámbito de la teledetección, destacando las ventajas y limitaciones de cada uno. Esta sección se centra en estudiar las limitaciones de dos algoritmos supervisados clásicos concretos: *paralelepíedos* y *mínima distancia*, para posteriormente mejorar su funcionamiento mediante el uso de autómatas celulares a lo largo de los siguientes apartados de este capítulo. La *Tabla 3.1* muestra un resumen de las ventajas y limitaciones que presentan dichos algoritmos de clasificación.

Clasificador	Ventajas	Limitaciones
<i>Paralelepíedos</i>	Considera la dispersión Rápido de ejecutar	Píxeles clasificados en varias clases Quedan píxeles sin clasificar
<i>Mínima distancia</i>	Se clasifican todos los píxeles Rápido de ejecutar	Proclive a errores de comisión No considera la varianza

Tabla 3.1: Ventajas y limitaciones de los algoritmos *paralelepíedos* y *mínima distancia*.

El algoritmo *paralelepíedos* presenta dos limitaciones importantes. Por un lado, algunos píxeles pueden ser clasificados en varias clases, por encontrarse dentro del área de varios paralelepíedos. Por otro lado, en muchas ocasiones pueden quedar píxeles sin clasificar, por encontrarse fuera del paralelepípedo de la clase a la que pertenecen.

El algoritmo *mínima distancia* también presenta dos limitaciones importantes. Por un lado, es proclive a los errores de comisión, es decir, en muchas ocasiones los píxeles son etiquetados en una clase a la que no pertenecen. Por otro lado, no considera la varianza de las clases, ya que sólo tiene en cuenta la distancia que existe entre el centro de las clases y cada píxel de la imagen.

Si nos centramos en las limitaciones de dichos algoritmos, podemos observar que la mayoría se pueden solucionar si aplicamos posteriormente algún *algoritmo contextual* que mejore la clasificación obtenida. Este tipo de algoritmos no sólo utilizan la información espectral de cada píxel para asignarlo a una clase determinada, sino también los valores de los vecinos de alrededor en aquellos píxeles que se encuentran espectralmente cercanos a varias clases y por lo tanto son más complejos de clasificar. A este tipo de píxeles, a los que estamos llamando *píxeles inciertos* a lo largo de este documento, son los causantes de la limitación “píxeles clasificados en varias clases” del algoritmo *paralelepíedos* y la limitación “proclive a errores de comisión” del algoritmo *mínima distancia*. Por otro lado, la limitación “quedan píxeles sin clasificar” del algoritmo *paralelepíedos* también se puede solucionar con el uso de algún algoritmo contextual de post-clasificación, asignando a cada píxel no clasificado la clase mayoritaria del conjunto de todos sus píxeles vecinos.

La *Figura 3.1* muestra las limitaciones del algoritmo *paralelepíedos*. A los píxeles que han quedado sin clasificar se les ha asignado el color de sus clases correspondientes, y a los píxeles inciertos se les ha asignado el color negro.

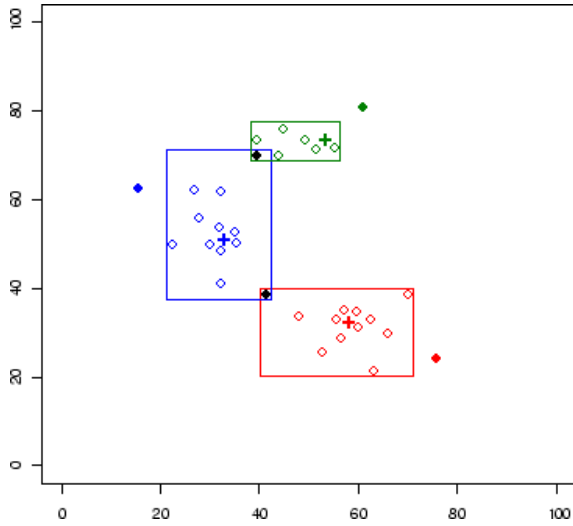


Figura 3.1: Limitaciones del algoritmo *paralelepípedos*.

La *Figura 3.2* muestra las limitaciones del algoritmo *mínima distancia*. A los píxeles inciertos también se les ha asignado el color negro.

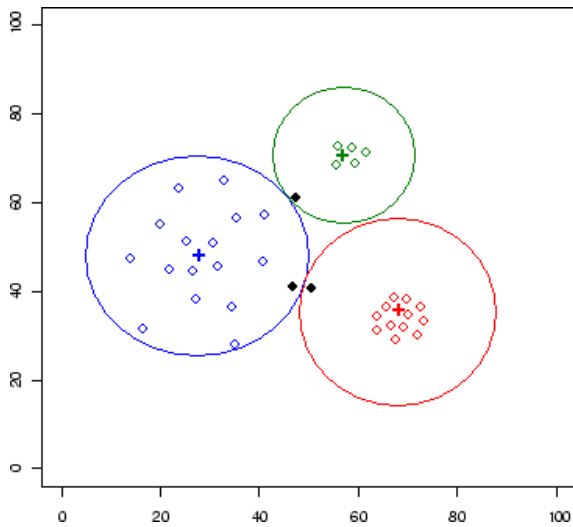


Figura 3.2: Limitaciones del algoritmo *mínima distancia*.

Las siguientes secciones se centran en el análisis de los errores que suele cometer el algoritmo *mínima distancia* con los píxeles inciertos de una imagen de satélite, a través de un ejemplo práctico de clasificación.

### 3.1.2. Análisis de la tasa de acierto: matriz de confusión

En esta sección vamos a analizar la **tasa de acierto** obtenida por parte del algoritmo *mínima distancia* clásico a través de un ejemplo práctico ilustrativo. Las pruebas se han llevado a cabo sobre una imagen de satélite *Landsat TM* multispectral de 7 bandas, con una resolución espacial de  $30 \times 30$  m y un tamaño de  $300 \times 300$  píxeles. A la imagen se le ha añadido un porcentaje significativo de ruido. La *Figura 3.3* muestra la zona de estudio, que pertenece a una región geográfica de las provincias de Almería y Granada, situadas en el sureste de España.

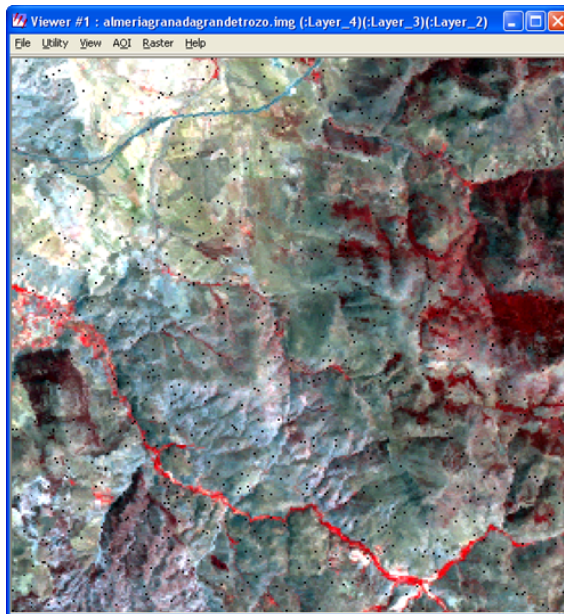


Figura 3.3: Zona de estudio.

El algoritmo *mínima distancia* ha etiquetado los píxeles de la imagen de satélite en 8 clases distintas, teniendo en cuenta un conjunto de entrenamiento elegido previamente. A continuación, se han utilizado los cálculos de una matriz de confusión para determinar qué píxeles han sido clasificados de manera correcta y cuáles se han clasificado erróneamente. Una **matriz de confusión** es una tabla de contingencia donde se muestran los resultados del muestreo de control de la calidad de un proceso de clasificación, comparando los resultados de la clasificación con los valores obtenidos mediante trabajo de campo. Los píxeles bien clasificados son aquellos que aparecen en la diagonal principal, y el resto de píxeles son los han sido clasificados de manera errónea. Debido a la enorme heterogeneidad del terreno de la zona de estudio, la tasa de acierto obtenida por parte del algoritmo *mínima distancia* ha sido muy baja. La siguiente sección analiza detalladamente los *errores* que ha cometido el algoritmo *mínima distancia*.

### 3.1.3. Análisis de errores durante el proceso de clasificación

En esta sección vamos a analizar los *errores cometidos* por parte del algoritmo *mínima distancia* clásico durante el proceso de clasificación del ejemplo práctico ilustrativo de la sección anterior, con el objetivo de poder corregirlos posteriormente mediante el uso de los *autómatas celulares*.

En primer lugar, vamos a comenzar detectando el origen del principal problema al que tiene que enfrentarse el proceso de clasificación: la existencia de **píxeles inciertos** a lo largo de toda la imagen de satélite. Para ello, hemos comenzado obteniendo una lista de 200 píxeles mal clasificados por parte del algoritmo *mínima distancia*, tomados al azar. La *Tabla 3.2*, que se encuentra en la siguiente página, muestra dicha lista de píxeles, donde se muestra la posición [x, y] de cada píxel, y a continuación la clase a la que pertenece (según el trabajo de campo realizado por expertos ecólogos) y la clase que se le ha asignado de manera errónea (según los resultados del algoritmo de clasificación *mínima distancia*).

Como se puede observar en dicha tabla, se han destacado en negrita los píxeles que poseen la relación de clases “5, 7” (píxeles inciertos que pertenecen a la clase 5, pero que se han clasificado erróneamente en la clase 7). De los 200 píxeles mal clasificados tomados al azar, 45 son de este tipo (un 22,5 % del total). Eso nos lleva a deducir que ambas clases se encuentran muy próximas entre sí desde un punto de vista espectral, y por lo tanto existe cierta ambigüedad durante el proceso de clasificación. Tomamos a modo de ejemplo dos píxeles del tipo “5, 7”, concretamente los píxeles 2 y 3 de la *Tabla 3.2*, cuya posición es [003, 161] y [003, 162], respectivamente. La *Figura 3.4* muestra la clasificación realizada con *trabajo de campo* y la obtenida mediante el algoritmo *mínima distancia*, centrada en dichos píxeles.

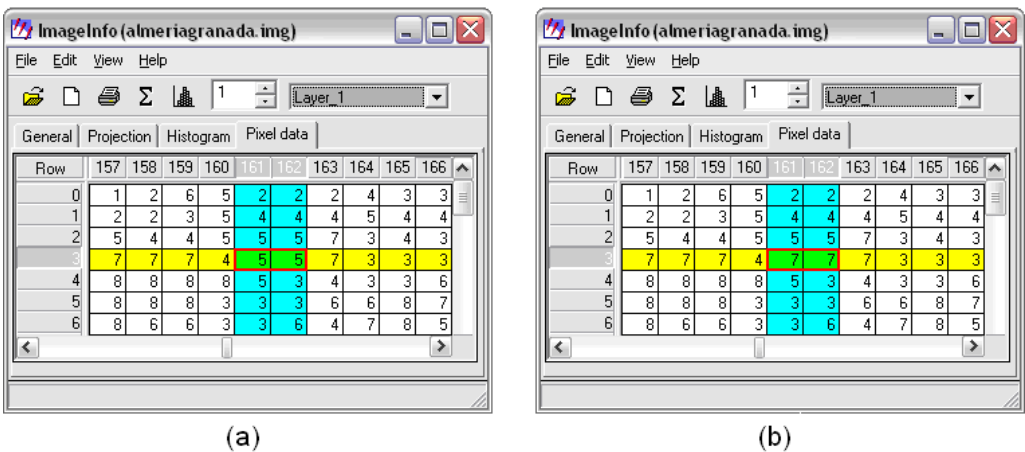


Figura 3.4: (a) Clasificación realizada por *trabajo de campo* centrada en los dos píxeles. (b) Clasificación obtenida por *mínima distancia* centrada en los dos píxeles de estudio. En ambas imágenes, los píxeles de estudio ocupan la posición [003, 161] y [003, 162].

Píxel	Clases	Píxel	Clases	Píxel	Clases	Píxel	Clases
[002, 218]	<b>5, 7</b>	[093, 092]	7, 8	[174, 032]	4, 5	[238, 259]	<b>5, 7</b>
[003, 161]	<b>5, 7</b>	[094, 109]	7, 8	[174, 253]	7, 8	[239, 034]	7, 8
[003, 162]	<b>5, 7</b>	[094, 248]	4, 5	[175, 159]	3, 5	[242, 173]	6, 3
[003, 293]	4, 5	[094, 249]	4, 5	[175, 169]	4, 5	[243, 296]	7, 8
[006, 241]	1, 2	[095, 296]	1, 2	[178, 146]	8, 6	[246, 097]	7, 8
[007, 009]	5, 3	[096, 123]	7, 8	[179, 032]	4, 5	[247, 259]	7, 8
[007, 245]	1, 2	[097, 282]	4, 5	[182, 021]	2, 4	[248, 288]	1, 2
[008, 206]	2, 4	[099, 146]	<b>5, 7</b>	[182, 169]	4, 5	[250, 082]	<b>5, 7</b>
[008, 207]	2, 4	[103, 108]	4, 5	[185, 030]	7, 8	[251, 117]	4, 5
[008, 271]	2, 4	[104, 108]	4, 5	[186, 187]	4, 5	[251, 136]	2, 4
[010, 190]	2, 4	[104, 134]	7, 8	[186, 267]	4, 5	[252, 157]	<b>5, 7</b>
[011, 164]	4, 5	[104, 276]	<b>5, 7</b>	[189, 300]	4, 5	[253, 134]	2, 4
[012, 168]	<b>5, 7</b>	[115, 270]	2, 4	[190, 199]	<b>5, 7</b>	[254, 116]	<b>5, 7</b>
[014, 294]	4, 5	[116, 191]	4, 5	[194, 050]	<b>5, 7</b>	[255, 161]	7, 8
[018, 144]	<b>5, 7</b>	[117, 191]	4, 5	[195, 274]	2, 4	[256, 081]	8, 6
[020, 169]	4, 5	[118, 035]	8, 6	[195, 291]	2, 4	[257, 081]	8, 6
[021, 169]	4, 5	[118, 194]	2, 4	[196, 232]	<b>5, 7</b>	[258, 300]	2, 4
[026, 145]	7, 8	[118, 260]	2, 4	[196, 291]	2, 4	[260, 107]	7, 3
[027, 251]	4, 5	[119, 260]	2, 4	[197, 057]	5, 3	[260, 241]	7, 8
[028, 285]	<b>5, 7</b>	[123, 275]	3, 5	[197, 073]	4, 5	[260, 242]	7, 8
[029, 174]	4, 5	[126, 038]	4, 5	[197, 300]	1, 2	[262, 071]	2, 4
[034, 230]	2, 4	[126, 282]	1, 2	[198, 290]	4, 5	[262, 205]	<b>5, 7</b>
[034, 250]	<b>5, 7</b>	[127, 239]	<b>5, 7</b>	[198, 300]	1, 2	[263, 017]	4, 5
[036, 213]	2, 4	[127, 240]	<b>5, 7</b>	[199, 016]	2, 4	[263, 048]	2, 4
[038, 206]	<b>5, 7</b>	[133, 178]	4, 5	[199, 083]	<b>5, 7</b>	[264, 070]	2, 4
[043, 236]	4, 5	[133, 225]	<b>5, 7</b>	[199, 198]	<b>5, 7</b>	[270, 210]	7, 8
[043, 260]	4, 5	[137, 156]	4, 5	[200, 022]	4, 5	[271, 055]	4, 5
[044, 286]	<b>5, 7</b>	[141, 090]	8, 6	[200, 026]	4, 5	[274, 121]	2, 4
[045, 278]	<b>5, 7</b>	[141, 174]	<b>5, 7</b>	[203, 076]	2, 4	[278, 166]	8, 6
[050, 230]	2, 4	[142, 174]	<b>5, 7</b>	[203, 238]	4, 5	[279, 279]	<b>5, 7</b>
[051, 243]	<b>5, 7</b>	[144, 202]	4, 5	[204, 061]	<b>5, 7</b>	[280, 093]	4, 5
[052, 197]	<b>5, 7</b>	[145, 231]	<b>5, 7</b>	[205, 018]	2, 4	[281, 045]	2, 4
[055, 052]	7, 8	[148, 134]	7, 8	[205, 059]	4, 5	[282, 226]	8, 6
[056, 251]	<b>5, 7</b>	[149, 186]	5, 3	[209, 208]	<b>5, 7</b>	[283, 176]	7, 8
[059, 167]	2, 4	[150, 081]	4, 5	[209, 241]	4, 5	[284, 010]	<b>5, 7</b>
[059, 238]	2, 4	[150, 122]	4, 5	[214, 096]	3, 5	[284, 168]	4, 5
[059, 268]	5, 3	[150, 224]	2, 4	[215, 120]	4, 5	[285, 242]	<b>5, 7</b>
[060, 282]	4, 5	[150, 250]	2, 4	[215, 263]	7, 8	[286, 065]	4, 5
[061, 085]	7, 8	[150, 251]	2, 4	[217, 019]	<b>5, 7</b>	[287, 065]	4, 5
[062, 021]	7, 8	[151, 224]	2, 4	[217, 064]	8, 6	[290, 110]	4, 5
[063, 144]	7, 8	[151, 225]	2, 4	[218, 192]	4, 5	[290, 180]	<b>5, 7</b>
[069, 102]	7, 8	[152, 225]	2, 4	[220, 088]	<b>5, 7</b>	[294, 065]	2, 4
[069, 247]	4, 5	[153, 240]	4, 5	[220, 089]	<b>5, 7</b>	[294, 223]	<b>5, 7</b>
[070, 235]	4, 5	[153, 244]	<b>5, 7</b>	[224, 040]	7, 8	[294, 257]	7, 8
[071, 117]	7, 8	[154, 120]	<b>5, 7</b>	[225, 026]	<b>5, 7</b>	[297, 120]	4, 5
[076, 192]	4, 5	[154, 240]	4, 5	[229, 012]	6, 3	[298, 075]	2, 4
[076, 211]	7, 8	[157, 281]	1, 2	[229, 028]	<b>5, 7</b>	[298, 120]	4, 5
[078, 067]	7, 8	[162, 197]	4, 5	[230, 202]	4, 5	[299, 263]	4, 5
[079, 246]	4, 5	[163, 283]	1, 2	[231, 226]	4, 5	[300, 212]	1, 2
[093, 017]	7, 8	[169, 210]	4, 5	[233, 233]	7, 8	[300, 217]	7, 8

Tabla 3.2: Lista de 200 píxeles mal clasificados por parte del algoritmo *mínima distancia* clásico, tomados al azar. En negrita, los píxeles que poseen la relación de clases “5, 7”.

Si nos fijamos con detenimiento en los dos píxeles de estudio, nos damos cuenta de que están rodeados en su mayoría por píxeles de la clase 5, en lugar de píxeles de la clase 7. Vamos a analizar por qué se han clasificado erróneamente dichos píxeles. La *Tabla 3.3* muestra los valores espectrales de ambos píxeles, que curiosamente son iguales. Por lo tanto, por simplificación, vamos a tratar a ambos píxeles como uno solo.

Píxel	Banda1	Banda2	Banda3	Banda4	Banda5	Banda6	Banda7
[003, 161]	72	39	52	84	111	160	54
[003, 162]	72	39	52	84	111	160	54

Tabla 3.3: Valores espectrales de los dos píxeles de estudio.

Por otro lado, la *Tabla 3.4* muestra los valores espectrales del conjunto de entrenamiento de las 8 clases en las que se ha clasificado la zona de estudio, para cada una de las 7 bandas de la imagen *Landsat*.

Clase	Banda1	Banda2	Banda3	Banda4	Banda5	Banda6	Banda7
1	56,7658	25,5007	28,0801	48,5983	53,8074	142,911	24,5118
2	65,6057	31,5801	38,1911	53,6608	74,2405	150,921	36,6283
3	91,8313	46,3399	59,8819	62,9785	96,1951	167,733	51,7341
4	73,7848	36,7980	46,8314	56,1121	89,0304	156,462	45,9838
<b>5</b>	<b>76,6276</b>	<b>39,2358</b>	<b>51,7107</b>	<b>59,9246</b>	<b>104,636</b>	<b>160,299</b>	<b>54,5456</b>
6	100,406	53,8322	71,6157	72,6973	114,660	170,330	62,3255
<b>7</b>	<b>81,3447</b>	<b>42,9589</b>	<b>58,5874</b>	<b>64,9728</b>	<b>117,165</b>	<b>165,610</b>	<b>61,9089</b>
8	84,8517	46,2963	65,4349	70,5442	130,463	170,348	69,9688

Tabla 3.4: Valores espectrales del conjunto de entrenamiento.

Teniendo en cuenta estos datos, usados por el algoritmo *mínima distancia* durante el proceso de clasificación, las *Tablas 3.5* y *3.6* resumen los valores calculados por dicho algoritmo para las clases 5 y 7, respectivamente: diferencia al cuadrado de los valores espectrales del píxel y las clases, para cada banda de la imagen.

	Banda1	Banda2	Banda3	Banda4	Banda5	Banda6	Banda7
Clase 5	76,6276	39,2358	51,7107	59,9246	104,636	160,299	54,5456
Píxel	72	39	52	84	111	160	54
<b>Diferencia<sup>2</sup></b>	<b>21,41</b>	<b>0,06</b>	<b>0,08</b>	<b>579,62</b>	<b>40,50</b>	<b>0,09</b>	<b>0,30</b>

Tabla 3.5: Cálculo de la distancia del píxel de estudio a la clase 5.

	Banda1	Banda2	Banda3	Banda4	Banda5	Banda6	Banda7
Clase 7	81,3447	42,9589	58,5874	64,9728	117,165	165,610	61,9089
Píxel	72	39	52	84	111	160	54
Diferencia <sup>2</sup>	<b>87,32</b>	<b>15,67</b>	<b>43,39</b>	<b>362,03</b>	<b>38,01</b>	<b>31,47</b>	<b>62,55</b>

Tabla 3.6: Cálculo de la distancia del píxel de estudio a la clase 7.

Si sumamos los valores *diferencia*<sup>2</sup> de las tablas anteriores, obtenemos una distancia de 642,06 a la clase 5 y 640,44 a la clase 7. Como se puede observar, la distancia espectral de dicho píxel a ambas clases es muy parecida, tan sólo existe una diferencia de 1,62. Por lo tanto se trata de un **píxel incierto**, ya que se encuentra muy próximo a ambas clases desde el punto de vista espectral. Se podría utilizar *información contextual* durante el proceso de clasificación para ayudar a este tipo de píxeles, ya que como hemos observado anteriormente en la *Figura 3.4*, los dos píxeles de estudio, ambos con los mismos valores espectrales, se encontraban rodeados en su mayoría por píxeles de la clase 5, en lugar de píxeles de la clase 7. Ayudado con técnicas contextuales, estos píxeles se hubieran clasificado correctamente.

Aparte de los píxeles inciertos, también existen los **píxeles ruidosos**. No son muy frecuentes, pero también pueden ser tratados mediante algoritmos contextuales, en procesos de pre-clasificación de la imagen de satélite. La *Figura 3.5* muestra los valores espectrales de un píxel ruidoso de la imagen de satélite de la zona de estudio, con un porcentaje significativo de ruido introducido artificialmente.

Row	0	1	2	3	4	5	6	7	8	9	10	11	12
0	15.000	12.000	16.000	11.000	13.000	10.000	06.000	08.000	01.000	06.000	09.000	08.000	99.000
1	15.000	10.000	10.000	05.000	15.000	14.000	09.000	00.000	03.000	03.000	12.000	06.000	05.000
2	13.000	05.000	07.000	03.000	10.000	11.000	08.000	11.000	06.000	09.000	15.000	98.000	98.000
3	06.000	02.000	04.000	13.000	11.000	09.000	09.000	11.000	10.000	07.000	04.000	85.000	99.000
4	97.000	95.000	99.000	09.000	10.000	09.000	04.000	09.000	97.000	96.000	94.000	00.000	97.000
5	94.000	95.000	00.000	12.000	14.000	11.000	09.000	10.000	02.000	83.000	85.000	94.000	97.000

Figura 3.5: Valor espectral de un píxel ruidoso.

Los *autómatas celulares* nos permiten implementar técnicas de *pre-clasificación*, para la eliminación de los píxeles ruidosos, y *post-clasificación*, para el refinamiento de los píxeles inciertos, agrupadas en una junto con el proceso de clasificación, como se muestra en el siguiente apartado.



## 3.2. CLASIFICACIÓN DE IMÁGENES DE SATÉLITE CON ACA

En este apartado se cita el trabajo previo existente relacionado con la clasificación de imágenes de satélite mediante autómatas celulares, se enumeran los objetivos principales del algoritmo ACA así como la información complementaria de cada objetivo, y se exponen las consideraciones previas a tener en cuenta antes de aplicar dicho algoritmo.

### 3.2.1. Trabajo previo existente

Como vimos en el *Capítulo 1*, a pesar del gran número de algoritmos de clasificación de imágenes de satélite que existen disponibles, en la actualidad existen numerosos grupos de investigación estudiando nuevos métodos clasificadores [97] [98] [107] [114] [115], ya que todavía no se ha creado ningún clasificador que ofrezca una tasa de acierto del 100% para las imágenes de satélite complejas [7] [8]. Por otro lado, como vimos en el *Capítulo 2*, hasta ahora los autómatas celulares han sido utilizados por la comunidad científica principalmente para implementar todo tipo de simulaciones [14]. En el ámbito de la teledetección, las simulaciones han sido sobre todo de tipo medioambiental y social. La aplicación de los autómatas celulares en el proceso de clasificación de imágenes de satélite es un campo nuevo de estudio, en el que existen muy pocos trabajos previos significativos. En este contexto, se han utilizado principalmente para implementar algoritmos de post-clasificación y reconocimiento de patrones, en combinación con otras técnicas. Los trabajos existentes se pueden dividir en tres grandes categorías:

- **Post-clasificación de imágenes de satélite mediante autómatas celulares:** los autómatas celulares se han utilizado en procesos de post-clasificación para mejorar los resultados obtenidos previamente por algún algoritmo de clasificación. Como ejemplos podemos encontrar la aplicación de autómatas celulares con autoaprendizaje a la post-clasificación de imágenes de satélite [76] o la post-clasificación de imágenes *Landsat* aplicada en países en desarrollo [94].
- **Combinación de autómatas celulares con otras técnicas:** los autómatas celulares se han utilizado, combinados con algoritmos genéticos, para el reconocimiento de patrones [71]. También existen trabajos de investigación relacionados con los autómatas celulares combinados con redes neuronales artificiales, que permiten definir reglas con un mayor grado de objetividad, con el objetivo de simular el uso de la tierra utilizando un *SIG* [67] [70].
- **Clasificación de imágenes mediante autómatas celulares:** en este aspecto destaca un método de clasificación de imágenes basado en las transformaciones de los autómatas celulares [16]. No existen demasiados trabajos en esta tercera categoría, que es precisamente en la que se centra esta tesis doctoral.

Como se puede observar, a pesar de las enormes ventajas que implica la utilización de autómatas celulares en los procesos de clasificación, existen pocos trabajos previos relacionados específicamente con la clasificación de imágenes de satélite mediante autómatas celulares. Este capítulo se centra en la descripción detallada del algoritmo ACA, y la siguiente sección comienza describiendo sus objetivos principales.

### 3.2.2. Objetivos principales del algoritmo ACA

Esta tesis doctoral propone un nuevo algoritmo de clasificación de imágenes de satélite supervisado basado en autómatas celulares multiestado, llamado *ACA* (*classification Algorithm based on Cellular Automata*) [41] [43] [44] [45] [46]. El algoritmo ACA se basa en la aplicación de las propiedades de los autómatas celulares sobre las imágenes de satélite, con el propósito de implementar un proceso de clasificación supervisado espectral-contextual. Los tres objetivos principales del algoritmo ACA son los siguientes:

- **Objetivo #1: mejorar la tasa de acierto de la clasificación.** Este objetivo pretende aumentar la *tasa de acierto* obtenida por parte de otros algoritmos de clasificación clásicos a través del uso de *información contextual*. Para ello, el algoritmo ACA clasificará los píxeles problemáticos teniendo en cuenta no sólo sus valores espectrales (ambiguos, en el caso de los píxeles inciertos, y erróneos, en el caso de los píxeles ruidosos), sino también los valores de los píxeles vecinos, para afinar más en la tasa de acierto final de la clasificación. Gracias a este objetivo, se fusionaran tres procesos distintos que normalmente se utilizan por separado y en el siguiente orden: pre-clasificación (eliminación de píxeles ruidosos), clasificación y post-clasificación (refinamiento de píxeles inciertos).
- **Objetivo #2: construir una clasificación jerárquica.** Este objetivo pretende obtener una *clasificación jerárquica* dividida en niveles o capas de fiabilidad basadas en la proximidad espectral de los píxeles a sus clases correspondientes, de tal manera que las primeras capas sean más fiables que las últimas, en términos de tasa de acierto. Para conseguir este objetivo, en cada iteración del autómata celular serán clasificados sólo aquellos píxeles que estén dentro de una distancia espectral máxima permitida en el espacio de características respecto del centro de sus clases correspondientes, y esta distancia se incrementará en cada iteración. A partir de ahora, esta distancia variable recibirá el nombre de **threshold**. De este modo obtendremos una clasificación jerárquica dividida en *niveles de calidad*, donde los primeros niveles ofrecerán mayor fiabilidad que los últimos. Además, la clasificación jerárquica ayudará al primer objetivo, ya que los píxeles inciertos y ruidosos, que están más alejados normalmente del centro de sus clases, serán clasificados en las últimas iteraciones del autómata celular, utilizando como vecinos a píxeles que con una gran probabilidad se habrán clasificado en iteraciones anteriores y por lo tanto ofrecerán mayor fiabilidad, mejorando así la tasa de acierto final.
- **Objetivo #3: detectar bordes y píxeles inciertos/ruidosos.** Este objetivo pretende conseguir una lista detallada de los píxeles *inciertos* y *ruidosos* para tener un mayor control sobre los píxeles problemáticos, así como una lista de los píxeles que determinen los *bordes espaciales* de las clases de la imagen de satélite. Esta segunda opción será bastante interesante, ya que la detección de bordes espaciales (no espectrales) de las clases de una imagen también es un problema clásico dentro del ámbito de la teledetección. Gracias a esta información, en futuros análisis podremos establecer, por ejemplo, cuántas agrupaciones de una misma clase existen en una imagen de satélite, como contar el número de edificios de una ciudad.

### 3.2.3. Información complementaria de los objetivos principales

El algoritmo ACA también pretende ofrecer *información complementaria* para cada uno de los tres objetivos principales vistos en la sección anterior. El propósito de toda esta información complementaria consiste en poder analizar con detenimiento no sólo el comportamiento del algoritmo ACA a lo largo de las distintas iteraciones del autómata celular, sino además, estudiar la evolución de las clases y de los píxeles de la imagen de satélite, así como representar gráficamente la evolución de los valores de determinados parámetros, como la tasa de acierto, la precisión y el *recall*. La información complementaria que ofrece el algoritmo ACA para cada uno de los tres objetivos principales establecidos es la siguiente:

- **Información complementaria #1: evolución de las matrices de confusión.** Esta primera información complementaria pretende proporcionar una matriz de confusión intermedia para cada iteración del autómata celular, con el propósito de que el analista experto pueda estudiar detenidamente no sólo el comportamiento del algoritmo ACA en cada imagen de satélite concreta, sino que además pueda analizar la evolución de la tasa de acierto a lo largo del proceso de clasificación. Dicha información complementa al *objetivo #1*, ya que nos permitirá obtener la *evolución de la tasa de acierto*.
- **Información complementaria #2: evolución de la precisión y el *recall*.** Esta segunda información complementaria pretende obtener el valor de la precisión y el *recall* de cada clase para cada iteración del autómata celular, con el propósito de estudiar la evolución de dichos valores a lo largo del proceso de clasificación a través de las posteriores gráficas de evolución. Dicha información complementa al *objetivo #2*, ya que nos permitirá obtener la *evolución de la precisión y el recall* a través de la clasificación jerárquica.
- **Información complementaria #3: evolución de las tablas de píxeles.** Esta tercera información complementaria pretende obtener una tabla intermedia de píxeles con información desglosada de cada tipo de píxel para cada iteración del autómata celular, con el propósito de completar la información ofrecida por las matrices de confusión intermedias. Dicha información complementa al *objetivo #3*, ya que incrementará la información proporcionada por dicho objetivo a través de la *evolución de las tablas intermedias de píxeles*. La *Tabla 3.7* muestra un índice de los tres objetivos principales propuestos, así como la información complementaria ofrecida por parte del algoritmo ACA para cada uno de los objetivos.

#	Objetivos principales	Información complementaria
1	Mejorar la tasa de acierto de la clasificación	Evolución de las matrices de confusión
2	Construir una clasificación jerárquica	Evolución de la precisión y el <i>recall</i>
3	Detectar bordes y píxeles inciertos/ruidosos	Evolución de las tablas de píxeles

Tabla 3.7: Objetivos principales e información complementaria del algoritmo ACA.

### 3.2.4. Consideraciones previas del algoritmo ACA

Como hemos visto en las dos secciones anteriores, el principal objetivo del algoritmo ACA consiste en clasificar los píxeles de la imagen de satélite basándose no sólo en la información espectral de cada píxel, sino también en la información contextual de sus píxeles vecinos, y por lo tanto mejorar los resultados obtenidos por otros algoritmos de clasificación existentes en la literatura desde varios puntos de vista: aumentar la tasa de acierto, conseguir una clasificación jerárquica, y obtener la máxima cantidad de información extra derivada del proceso de clasificación. Para implementar el algoritmo ACA y conseguir estos objetivos, debemos tener en cuenta las siguientes consideraciones previas entre el modelo general de autómatas celulares multiestado y los elementos básicos de un proceso genérico de clasificación de imágenes de satélite:

- Cada *celda* del autómatas celulares corresponde a un *píxel* de la imagen de satélite.
- Cada celda tiene *tres estados*, uno por objetivo, y nos indican: la *clase* a la que pertenece cada píxel, la *iteración* del autómatas celulares en la que se clasificó y el *tipo* concreto de píxel.
- La *vecindad* de cada celda puede estar formada por 4 píxeles (vecindad *von Neumann*), 8 píxeles (vecindad de *Moore*) ó 24 píxeles (vecindad de *Moore* extendida).
- La función de transición  $f$  debe clasificar cada píxel de la imagen en su clase correspondiente basándose en las características de la celda actual y de su vecindad, y aplicando las *reglas* del autómatas celulares a lo largo de las distintas iteraciones.

El algoritmo ACA está basado en los dos algoritmos de clasificación supervisados *paralelepípedos* y *mínima distancia* clásicos. Por lo tanto, existen dos algoritmos ACA: **ACA paralelepípedos** y **ACA mínima distancia**, cada uno basado en su algoritmo clásico correspondiente. Por simplificación, a lo largo de este documento se usará **algoritmo ACA** para referirse a ambos algoritmos. El algoritmo ACA modifica el comportamiento de los algoritmos clásicos a través del uso de autómatas celulares, aplicando las reglas de la función de transición  $f$  para optimizar los resultados obtenidos. Otra consideración previa del algoritmo ACA consiste en que la función de transición  $f$  debe tener en cuenta las siguientes entradas para cada píxel de la imagen de satélite, en cada iteración del autómatas celulares:

- **Clases posibles ofrecidas por el algoritmo clásico:** conjunto de clases posibles del píxel actual, que puede ser *una* sola clase, o *varias* en el caso de los píxeles inciertos. Las clases espectrales son proporcionadas por los algoritmos *paralelepípedos* o *mínima distancia*, modificados mediante técnicas de autómatas celulares.
- **Iteración del autómatas celulares:** valor que especifica el *nivel de calidad* en el que se clasifica el píxel actual, y que posteriormente permite obtener el *grado de pertenencia* de cada píxel de la imagen a su clase correspondiente.
- **Estados de la vecindad:** conjunto de estados de los píxeles *vecinos* al píxel actual. El tipo de vecindad del autómatas celulares es elegida antes de aplicar el proceso de clasificación para personalizar los resultados obtenidos.

### 3.3. DESCRIPCIÓN DEL ALGORITMO ACA

En este apartado se muestran los estados y las reglas del autómata celular utilizado en el algoritmo ACA así como su descripción matemática, el pseudocódigo de los algoritmos *ACA principal*, *ACA espectral* y *ACA contextual*, las fórmulas complementarias utilizadas, una explicación detallada de la clasificación jerárquica y la arquitectura principal del algoritmo ACA.

#### 3.3.1. Estados y reglas del autómata celular

Para conseguir los objetivos principales del algoritmo ACA, citados en el apartado anterior, se ha utilizado un autómata celular multiestado en el que cada celda tiene 3 estados independientes, uno para cada uno de los objetivos principales. El primer estado es **clase**, y corresponde a la clase espectral en la que es clasificado cada píxel de la imagen de satélite utilizando no sólo sus valores espectrales, sino también la información contextual de los píxeles vecinos. Este estado nos permite mejorar la tasa de acierto de la clasificación (*objetivo #1*). El segundo estado es **calidad**, y corresponde al número de iteración del autómata celular en el que es clasificado cada píxel de la imagen. Este estado nos permite construir una clasificación jerárquica (*objetivo #2*). El tercer estado es **tipo**, y corresponde al tipo de píxel con el que estamos tratando: incierto, ruidoso, borde o foco, donde los píxeles foco son los que no son bordes espaciales de las clases. Este estado nos permite detectar bordes y píxeles inciertos/ruidosos (*objetivo #3*). A continuación se muestra todo el rango de valores que puede tomar cada uno de los 3 estados del autómata celular:

- **Estado #1 [clase]** = *claseEspectral* (clases espectrales definidas por el conjunto de entrenamiento) o *claseVacía* (píxeles que todavía no han sido clasificados).
- **Estado #2 [calidad]** =  $1 .. numIteraciones$  (iteración del autómata celular que indica el nivel de calidad de la capa dentro de la clasificación jerárquica).
- **Estado #3 [tipo]** = *incierto* (píxeles dudosos), *ruidoso* (píxeles ruidosos), *borde* (píxeles frontera de las clases) o *foco* (píxeles que no son bordes).

Como se puede observar, el estado #1 (*clase*) puede tomar cualquier clase espectral que haya sido definida anteriormente en el conjunto de entrenamiento, así como la clase vacía para aquellos píxeles que aún no han sido clasificados; en la primera iteración del autómata celular, todas las celdas tienen su estado 1 con el valor *claseVacía*, ya que aún no ha comenzado el proceso de clasificación. El estado #2 (*calidad*) toma como valor el número de iteración en la que el píxel es clasificado, un valor definido entre 1 y el número máximo de iteraciones. El estado #3 (*tipo*) indica el tipo de píxel con el que nos encontramos: incierto, ruidoso, borde o foco.

Aparte de los tres estados, también se han creado *cuatro reglas* que definen el comportamiento del autómata celular multiestado, una regla para cada tipo de píxel especificado en el estado #3 del algoritmo ACA:

- **Regla #1.** Si el número de posibles *claseEspectral* es igual a 0 porque el píxel actual tiene valores espectrales erróneos:  
 $[clase] [calidad] [tipo] = [clase\ mayoritaria\ de\ la\ vecindad] [iteración] [ruidoso]$
- **Regla #2.** Si el número de posibles *claseEspectral* es igual a 1, y todas las clases de la vecindad son *claseVacía* o iguales a la del píxel actual:  
 $[clase] [calidad] [tipo] = [claseEspectral] [iteración] [foco]$
- **Regla #3.** Si el número de posibles *claseEspectral* es igual a 1, y alguna clase de la vecindad es distinta a la del píxel actual:  
 $[clase] [calidad] [tipo] = [claseEspectral] [iteración] [borde]$
- **Regla #4.** Si el número de posibles *claseEspectral* es mayor que 1:  
 $[clase] [calidad] [tipo] = [clase\ mayoritaria\ de\ la\ vecindad\ entre\ las\ clases\ dudosas] [iteración] [incierto]$

Como se puede observar, cada regla corresponde a uno de los cuatro tipos de píxeles que se han especificado previamente: la regla #1 para los píxeles ruidosos, la regla #2 para los píxeles foco, la regla #3 para los píxeles borde y la regla #4 para los píxeles inciertos. A través de las reglas #1 y #4, el algoritmo ACA mejora la tasa de acierto de la clasificación a través de técnicas contextuales (*objetivo #1*), y a través de las reglas #2 y #3, el algoritmo ACA obtiene una lista de píxeles focos y bordes espaciales de las clases de la imagen de satélite (*objetivo #3*). A través del comportamiento iterativo del autómata celular, el algoritmo ACA construye una clasificación jerárquica basada en niveles de calidad (*objetivo #2*). Por otro lado, la primera regla se caracteriza por tener un error en el proceso de clasificación, por lo que el número de clases espectrales posibles es 0. En la segunda y tercera regla, la primera condición es la misma: el número de clases espectrales posibles es 1. En la cuarta regla, la condición es distinta al resto: el número de clases espectrales posibles es mayor que 1. A continuación se ofrece una explicación más detallada del funcionamiento de cada una de las reglas:

- **Regla #1:** un píxel será etiquetado como *ruidoso* si sus valores espectrales son erróneos en todas las bandas de la imagen de satélite, y por lo tanto tomará como clase la *clase mayoritaria de la vecindad* (información contextual).
- **Regla #2:** un píxel será etiquetado como *foco* si no es el borde espacial de una clase, es decir, si los píxeles de alrededor pertenecen a la misma clase o a la clase vacía porque aún no han sido clasificados.
- **Regla #3:** un píxel será etiquetado como *borde* si es la frontera espacial de una clase espectral, y por lo tanto existen píxeles vecinos que pertenecen a una clase distinta de la clase de dicho píxel.
- **Regla #4:** un píxel será etiquetado como *incierto* si el número de clases espectrales posibles es mayor que 1, y por lo tanto tomará como clase la *clase mayoritaria de la vecindad entre las clases dudosas* (información contextual).

### 3.3.2. Descripción matemática del autómata celular

El **autómata celular** multiestado del algoritmo ACA se puede expresar, desde un punto de vista matemático y utilizando la misma nomenclatura que la usada a lo largo del *Capítulo 2*, a través de la siguiente expresión:

$$\text{ACA} = (d, r, Q, \#, V, f) = \quad (3.1)$$

$$(2 \times N, \{4|8|24\}, [q_{clase}, q_{calidad}, q_{tipo}], [clase\ Vacía, \emptyset, \emptyset], V, f)$$

donde:

- $d = 2 \times N$ : la dimensión espacial del autómata celular del algoritmo ACA es 2, es decir, las celdas están distribuidas en forma de matriz bidimensional. Sin embargo, como cada píxel de la imagen de satélite tiene un total de  $N$  valores almacenados, uno para cada banda espectral, realmente estamos trabajando sobre una dimensión  $d = 2 \times N$ .
- $r = \{4|8|24\}$ : el tamaño de la vecindad puede ser de 4, 8 ó 24 vecinos de alrededor, con el objetivo de personalizar el proceso de clasificación.
- $Q = [q_{clase}, q_{calidad}, q_{tipo}]$ : el conjunto de estados por celda está formado por todos los conjuntos de valores que puede tomar cada uno de los tres estados, como se muestra en la siguiente expresión:

$$[q_{clase}, q_{calidad}, q_{tipo}] =$$

$$[\{clase\ Espectral|clase\ Vacía\}, \quad (3.2)$$

$$\{1|\dots|num\ Iteraciones\},$$

$$\{incierto|ruidoso|borde|foco\}]$$

- $\# = [clase\ Vacía, \emptyset, \emptyset]$ : el estado *quiescente*, o valor inicial, es la clase vacía para el primer estado y el conjunto vacío para el resto de estados.
- $V$ : el vector de vecindad se puede configurar a 4, 8 ó 24 vecinos de cada una de las celdas en todas las dimensiones disponibles, como se muestra en la siguiente expresión:

$$V \subset (\mathbb{Z}^{2 \times N})^{\{4|8|24\}} \quad (3.3)$$

- $f$ : la función de transición aplica las cuatro reglas a cada una de las celdas a lo largo de las iteraciones con el propósito de cambiar sus estados, teniendo en cuenta la vecindad elegida, como se muestra en la siguiente expresión:

$$f : Q^{\{4|8|24\}+1} \rightarrow Q \quad (3.4)$$

### 3.3.3. Pseudocódigo del algoritmo ACA

A nivel de implementación, el algoritmo ACA se compone de tres algoritmos complementarios entre sí: *ACA principal*, *ACA espectral* y *ACA contextual*. El **algoritmo ACA principal** se encarga de ejecutar las iteraciones del autómata celular. En cada iteración, en primer lugar se realiza una clasificación espectral provisional de todos los píxeles de la imagen de satélite que aún no han sido clasificados y que se encuentran dentro del *threshold* permitido de la iteración actual, y a continuación se realiza una clasificación contextual de dichos píxeles con el propósito de mejorar los resultados ofrecidos por la clasificación espectral. El valor del *threshold* aumenta en cada iteración del proceso de clasificación. La *Tabla 3.8* muestra el pseudocódigo del algoritmo *ACA principal*.

---

**Algoritmo ACA principal** ( $ND, \overline{VC}, RD, ACA.r, ACA.nIter, thr$ )

---

**Entradas:**

$ND$ : niveles digitales de todos los píxeles en todas las bandas

$\overline{VC}$ : valor central de todas las clases en todas las bandas

$RD$ : rango de dispersión de todas las clases en todas las bandas

$ACA.r$ : tamaño de la vecindad del autómata celular

$ACA.nIter$ : número de iteraciones del autómata celular

$thr$ : *threshold* para cada iteración del autómata celular

**Salida:**

$ACA.Q = \{q_{clase}, q_{calidad}, q_{tipo}\}$ : conjunto de estados de todas las celdas

---

```

01 for  $i \leftarrow 1$  to  $ACA.nIter$  do
02   foreach  $ACA.Q_j | j \in \{1..numPíxeles\}$  do
03     if  $ACA.Q_{j,clase} = \emptyset$  then
04        $ACA.Q_j \leftarrow ACAespectral(ND_j, \overline{VC}, RD, ACA.Q_j, thr)$ ;
05       if  $ACA.Q_{j,clase} \neq \emptyset$  or  $i = ACA.nIter$  then
06          $ACA.Q_j \leftarrow ACAcontextual(ACA.Q_j, ACA.r, i)$ ;
07       endif
08     endif
09   end
10    $thr \leftarrow thr + inc$ ;
11 end
12 return  $ACA.Q$ ;

```

---

Tabla 3.8: Pseudocódigo del algoritmo *ACA principal*.

El **algoritmo ACA espectral** se basa en los algoritmos supervisados *paralelepípedos* y *mínima distancia* clásicos, mejorados a través de las técnicas que ofrecen los autómatas celulares, dividiendo el proceso de clasificación en varias iteraciones a través del incremento del *threshold*. Esta división ofrece una *clasificación jerárquica* formada por diferentes niveles o capas de fiabilidad basadas en el grado de pertenencia de los píxeles a sus clases correspondientes, donde las primeras capas ofrecen mayor calidad que las últimas. El algoritmo *ACA espectral* es el responsable del *objetivo #2*. La *Tabla 3.9* muestra el pseudocódigo del algoritmo *ACA espectral*.



---

**Algoritmo ACA espectral** ( $ND_j, \overline{VC}, RD, ACA.Q_j, thr$ )

---

**Entradas:**

$ND_j$ : niveles digitales del píxel  $j$  en todas las bandas  
 $\overline{VC}$ : valor central de todas las clases en todas las bandas  
 $RD$ : rango de dispersión de todas las clases en todas las bandas  
 $ACA.Q_j = \{q_{clase}, q_{calidad}, q_{tipo}\}$ : conjunto de estados de la celda  $j$   
 $thr$ : *threshold* para cada iteración del autómata celular

**Salida:**

$ACA.Q_j = \{q_{clase}, q_{calidad}, q_{tipo}\}$ : conjunto de estados de la celda  $j$

---

```

01 foreach  $clase_A | A \in \{1..numClases\}$  do
02   if  $clasificaciónJerárquica(ND_j, \overline{VC}, RD, thr, clase_A) \in thr$ 
03     then  $ACA.Q_{j,clase} \leftarrow ACA.Q_{j,clase} + clase_A$ ;
04   endif
05 end
06 return  $ACA.Q_j$ ;

```

---

Tabla 3.9: Pseudocódigo del algoritmo *ACA espectral*.

La función *clasificaciónJerárquica()* del algoritmo *ACA espectral* utiliza uno de los dos clasificadores clásicos modificados a través de los autómatas celulares. Para ejecutar el algoritmo *ACA paralelepípedos* se usa la siguiente expresión, modificada a partir de la *Expresión (1.4)* del *Capítulo 1*:

$$\overline{VC}_{A,n} - RD_{A,n}^{thr} \leq ND_{j,n} \leq \overline{VC}_{A,n} + RD_{A,n}^{thr} \quad (3.5)$$

donde el argumento modificado de la *Expresión (1.4)* es el siguiente:

- $RD_{A,n}^{thr}$ : rango de dispersión de la clase  $A$  en la banda  $n$  para la iteración  $i$  del proceso de clasificación. Este valor aumenta en cada iteración, teniendo en cuenta el valor de  $thr$ .

Para ejecutar el algoritmo *ACA mínima distancia* se usa la siguiente expresión, modificada a partir de la *Expresión (1.6)* del *Capítulo 1*:

$$ACA.Q_{j,clase} = \{A, \forall A | dist_{j,A} \leq thr\} \quad (3.6)$$

El algoritmo *ACA espectral* modifica el funcionamiento de los algoritmos clásicos a través de las *Expresiones (3.5)* y *(3.6)*, dividiendo los resultados del proceso de clasificación en el número de iteraciones del autómata celular a través del parámetro  $thr$ .

Por otro lado, el **algoritmo ACA contextual** es el encargado de aplicar las *cuatro reglas* del autómata celular, y también es el responsable de los otros dos objetivos principales del algoritmo ACA. Por un lado, el algoritmo *ACA contextual* mejora la tasa de acierto de la clasificación mediante técnicas contextuales (*objetivo #1*) a través de las reglas #1 y #4 para los píxeles ruidosos e inciertos.

Por otro lado, el algoritmo *ACA contextual* indica el tipo de píxel: ruidoso, foco, borde o incierto (*objetivo #3*) a través de las *reglas #1, #2, #3 y #4*. La *Tabla 3.10* muestra el pseudocódigo del algoritmo *ACA contextual*.

---

**Algoritmo ACA contextual** ( $ACA.Q_j, ACA.r, i$ )

---

**Entradas:**  
 $ACA.Q_j = \{q_{clase}, q_{calidad}, q_{tipo}\}$ : conjunto de estados de la celda  $j$   
 $ACA.r$ : tamaño de la vecindad del autómata celular  
 $i$ : iteración actual del autómata celular

**Salida:**  
 $ACA.Q_j = \{q_{clase}, q_{calidad}, q_{tipo}\}$ : conjunto de estados de la celda  $j$

---

```

01  if obtenerClases( $ACA.Q_{j,clase}$ ) =  $\emptyset$  then
02       $ACA.Q_{j,clase} = claseMayoritaria(j, ACA.r)$ ;
03       $ACA.Q_{j,tipo} = \text{"ruidoso"}$ ;
04  endif
05  if obtenerClases( $ACA.Q_{j,clase}$ ) = 1 and
06       $clasesIguales(ACA.Q_{j,clase}, ACA.Q_{ACA.r,clase}) = true$ 
07      then  $ACA.Q_{j,tipo} = \text{"foco"}$ ;
08  endif
09  if obtenerClases( $ACA.Q_{j,clase}$ ) = 1 and
10       $clasesIguales(ACA.Q_{j,clase}, ACA.Q_{ACA.r,clase}) = false$ 
11      then  $ACA.Q_{j,tipo} = \text{"borde"}$ ;
12  endif
13  if obtenerClases( $ACA.Q_{j,clase}$ ) > 1 then
14       $ACA.Q_{j,clase} = claseMayoritariaDudosas(ACA.Q_j, ACA.r)$ ;
15       $ACA.Q_{j,tipo} = \text{"incierto"}$ ;
16  endif
17   $ACA.Q_{j,calidad} = i$ ;
18  return  $ACA.Q_j$ ;

```

---

Tabla 3.10: Pseudocódigo del algoritmo *ACA contextual*.

El algoritmo *ACA contextual* utiliza una serie de funciones complementarias para poder aplicar las cuatro reglas del autómata celular. A continuación se muestra el propósito de cada función:

- Función **obtenerClases()**: obtiene las clases posibles del píxel actual devueltas tras la clasificación espectral preliminar.
- Función **clasesIguales()**: comprueba si la clase del píxel actual es la misma que las clases de toda su vecindad.
- Función **claseMayoritaria()**: obtiene la clase mayoritaria de la vecindad del píxel actual.
- Función **claseMayoritariaDudosas()**: obtiene la clase mayoritaria de la vecindad del píxel actual, entre las clases dudosas.

### 3.3.4. Fórmulas complementarias del algoritmo ACA

Con el objetivo de analizar la evolución de las diferentes clases durante el proceso de clasificación de la imagen de satélite, el algoritmo ACA también calcula la precisión y el *recall* de las clases a partir de la matriz de confusión intermedia de cada iteración del autómata celular. La **precisión** es un valor que mide el acierto de que una clase específica haya sido predicha correctamente, y su fórmula clásica es la siguiente:

$$precisión_A = \frac{MC_{A,A}}{\sum_{n=1}^{N_{clases}} MC_{A,n}} \quad (3.7)$$

donde:

- $precisión_A$ : precisión de la clase  $A$ .
- $MC_{A,A}$ : valor de la clase  $A$  en la diagonal principal de la matriz de confusión.
- $\sum_{n=1}^{N_{clases}} MC_{A,n}$ : suma de los valores de la fila de la clase  $A$  en la matriz de confusión.

Se ha adaptado la fórmula clásica de la precisión al comportamiento iterativo del algoritmo ACA, como muestra la siguiente fórmula:

$$precisión_A^i = \frac{MC_{A,A}^i}{\sum_{n=1}^{N_{clases}} MC_{A,n}^i} \quad (3.8)$$

donde:

- $precisión_A^i$ : precisión de la clase  $A$  en la iteración  $i$ .
- $MC_{A,A}^i$ : valor de la clase  $A$  en la diagonal principal de la matriz de confusión intermedia de la iteración  $i$ .
- $\sum_{n=1}^{N_{clases}} MC_{A,n}^i$ : suma de los valores de la fila de la clase  $A$  en la matriz de confusión intermedia de la iteración  $i$ .

El **recall** es un valor que mide la capacidad del modelo de predicción para seleccionar una clase concreta del conjunto de entrenamiento, y su fórmula clásica es la siguiente:

$$recall_A = \frac{MC_{A,A}}{\sum_{n=1}^{N_{clases}} MC_{n,A}} \quad (3.9)$$

donde:

- $recall_A$ : *recall* de la clase  $A$ .
- $MC_{A,A}$ : valor de la clase  $A$  en la diagonal principal de la matriz de confusión.
- $\sum_{n=1}^{N_{clases}} MC_{n,A}$ : suma de los valores de la columna de la clase  $A$  en la matriz de confusión.

También se ha adaptado la fórmula clásica del *recall* al comportamiento iterativo del algoritmo ACA, como muestra la siguiente fórmula:

$$recall_A^i = \frac{MC_{A,A}^i}{\sum_{n=1}^{N_{clases}} MC_{n,A}^i} \quad (3.10)$$

donde:

- $recall_A^i$ : *recall* de la clase  $A$  en la iteración  $i$ .
- $MC_{A,A}^i$ : valor de la clase  $A$  en la diagonal principal de la matriz de confusión intermedia de la iteración  $i$ .
- $\sum_{n=1}^{N_{clases}} MC_{n,A}^i$ : suma de los valores de la columna de la clase  $A$  en la matriz de confusión intermedia de la iteración  $i$ .

La clasificación jerárquica del algoritmo ACA, dividida en el número de iteraciones establecidas para el autómata celular, simula además una *clasificación pseudodifusa* que permite al analista experto calcular el **grado de pertenencia** de cada píxel a su clase correspondiente. Dicho valor es calculado para cada píxel de la imagen a lo largo del proceso de clasificación, y ofrece un porcentaje entre el número de iteración en la que se ha clasificado cada píxel concreto y el número total de iteraciones usadas en su clase correspondiente para clasificar todos los píxeles, como muestra la siguiente fórmula:

$$gp_{x,A} = \frac{iteración_{A,fin} - iteración_{x,A}}{iteración_{A,fin}} \quad (3.11)$$

donde:

- $gp_{x,A}$ : grado de pertenencia del píxel  $x$  a la clase  $A$ .
- $iteración_{A,fin}$ : iteración en la que todos los píxeles de la clase  $A$  se han clasificado.
- $iteración_{x,A}$ : iteración en la que el píxel  $x$  ha sido clasificado en la clase  $A$ .

La clasificación jerárquica también permite conocer el **grado de heterogeneidad** de cada clase, valor que indica la dificultad que presentan sus píxeles a la hora de ser clasificados, como muestra la siguiente fórmula:

$$gh_A = \frac{iteración_{A,fin}}{iteración_{im,fin}} \quad (3.12)$$

donde:

- $gh_A$ : grado de heterogeneidad de la clase  $A$ .
- $iteración_{A,fin}$ : iteración en la que todos los píxeles de la clase  $A$  se han clasificado.
- $iteración_{im,fin}$ : iteración en la que todos los píxeles de la imagen se han clasificado.

### 3.3.5. Clasificación jerárquica del algoritmo ACA

Uno de los aspectos más novedosos del algoritmo ACA es la obtención de una *clasificación jerárquica*. Para comprender mejor en qué consiste este tipo de clasificación, vamos a analizar detenidamente el siguiente ejemplo. La *Figura 3.6* muestra el proceso de clasificación jerárquica de los algoritmos *ACA paralelepípedos* y *ACA mínima distancia* mediante 3 iteraciones del autómatas celular, representado en el plano espectral, es decir, el espacio de características.

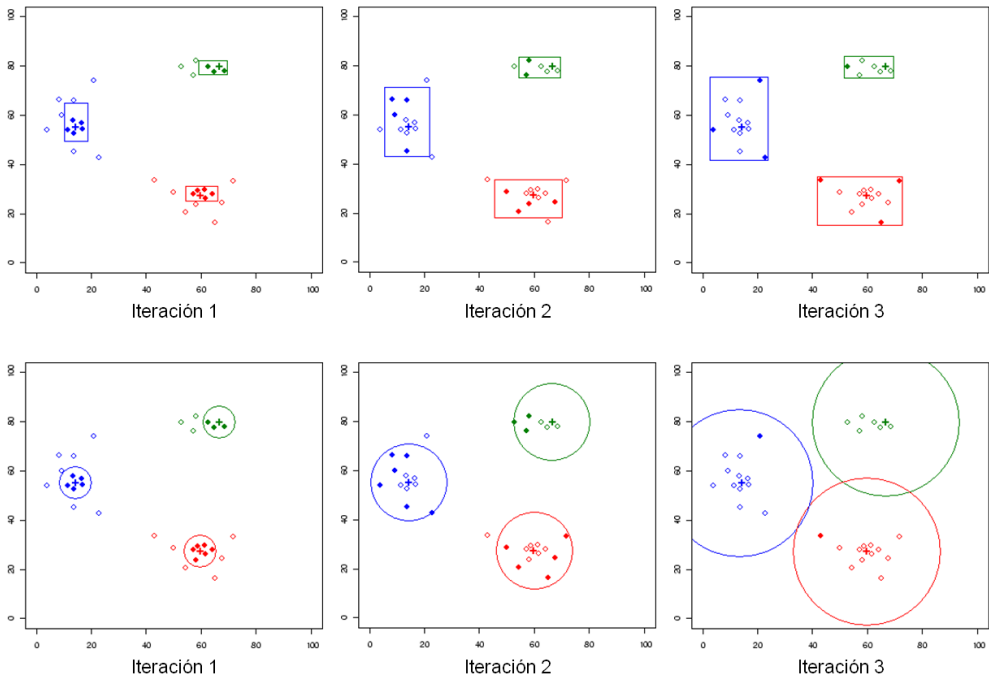


Figura 3.6: Evolución del algoritmo ACA mediante 3 iteraciones del autómatas celular: (Línea 1) *ACA paralelepípedos*. El *threshold* aumenta en proporción a la covarianza. (Línea 2) *ACA mínima distancia*. El *threshold* aumenta en igual proporción.

Como se puede observar, en cada iteración del autómatas celular va aumentando el *threshold*, y en consecuencia también aumenta el rango de píxeles que se clasifican. Los píxeles más próximos a sus clases correspondientes, desde un punto de vista espectral, son clasificados en la primera iteración. La mayoría de estos píxeles pertenecen incluso al conjunto de entrenamiento elegido por el analista experto. En las siguientes iteraciones se clasifican los píxeles más alejados de los centros espectrales de sus clases correspondientes, y por último se clasifican los píxeles inciertos mediante técnicas de clasificación contextual, con la seguridad de que dichas técnicas se basarán en píxeles vecinos bien clasificados en iteraciones anteriores, y por lo tanto con un nivel de calidad superior. Las primeras iteraciones, por lo tanto, tienen un nivel de calidad mayor que las siguientes.

### 3.3.6. Arquitectura del algoritmo ACA

Esta sección se centra en la arquitectura del algoritmo ACA, que está formada por una combinación de componentes relacionados con el ámbito de la teledetección y componentes relacionados con el modelo matemático de los autómatas celulares. La *Figura 3.7* muestra la arquitectura principal del algoritmo ACA.

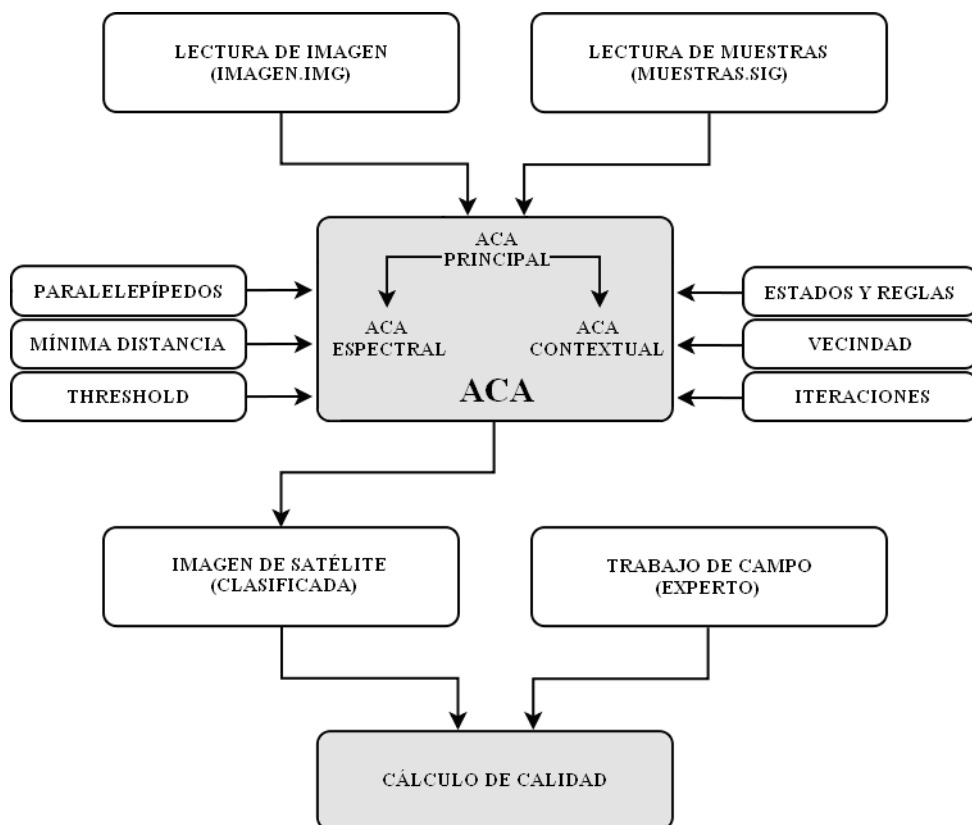


Figura 3.7: Arquitectura del algoritmo ACA.

Como se puede observar, la arquitectura del algoritmo ACA está compuesta por dos componentes principales: el **algoritmo ACA** propiamente dicho y la aplicación del **cálculo de calidad** de la clasificación obtenida. El algoritmo ACA, a su vez, está compuesto por los tres algoritmos descritos en secciones anteriores: *ACA principal*, *ACA espectral* y *ACA contextual*. No obstante, la arquitectura descrita en esta sección se trata en realidad de un modelo bastante simplificado, donde sólo se muestran los componentes relacionados con la obtención de los objetivos principales del algoritmo ACA. Los componentes relacionados con la obtención de la información complementaria de cada objetivo han sido obviados, para no aumentar la complejidad de la arquitectura. A continuación se explica el funcionamiento de los componentes principales:

- a) **Algoritmo ACA.** El algoritmo ACA tiene ocho componentes de entrada y un solo componente de salida: la imagen clasificada. Los ocho componentes de entrada se pueden agrupar en tres categorías distintas:
- **Componentes de entrada del algoritmo ACA relacionados con la imagen de satélite a clasificar.** El algoritmo ACA tiene dos entradas de este tipo: la aplicación de lectura de la imagen ( $imagen.img = ND$ ) y la aplicación de lectura del conjunto de muestras ( $muestras.sig = \{\overline{VC}, RD\}$ ) obtenido previamente a partir de píxeles que pertenecen a las clases elegidas para el proceso de clasificación. Estos dos componentes proporcionan al algoritmo ACA todos los elementos necesarios para realizar una clasificación de la imagen de tipo supervisada.
  - **Componentes de entrada del algoritmo ACA relacionados con los algoritmos clásicos de clasificación supervisada.** El algoritmo ACA se basa en los resultados obtenidos previamente por los clasificadores supervisados *paralelepípedos* o *mínima distancia*, modificados para que produzcan una clasificación jerárquica a través del *threshold* ( $thr$ ). El algoritmo ACA utiliza los componentes de entrada ( $imagen.img = ND$  y  $muestras.sig = \{\overline{VC}, RD\}$ ) para clasificar la imagen de satélite a través de los algoritmos clásicos modificados y mejorados mediante el uso de autómatas celulares. El analista experto debe elegir, antes de llevar a cabo el proceso de clasificación, el algoritmo supervisado clásico en el que quiere que se base el algoritmo ACA.
  - **Componentes de entrada del algoritmo ACA relacionados con los autómatas celulares.** Para mejorar los resultados obtenidos mediante los algoritmos clásicos supervisados, el algoritmo ACA modifica su comportamiento mediante el uso de autómatas celulares, añadiendo los elementos de estados ( $ACA.Q$ ), reglas ( $ACA.f$ ), vecindad ( $ACA.r$ ) e iteraciones ( $ACA.nIter$ ) al proceso de clasificación supervisado. Algunos de estos componentes relacionados con autómatas celulares pueden ser configurados por el analista experto antes de llevar a cabo el proceso de clasificación, como la vecindad (vecindad de *von Neumann*, vecindad de *Moore* o vecindad de *Moore* extendida), o el número de iteraciones, personalizando la configuración del algoritmo ACA para cada imagen de satélite concreta.
- b) **Cálculo de calidad.** Esta aplicación toma como entradas la imagen clasificada mediante el algoritmo ACA ( $imagenClasificada.img = ACA.Q$ ) y la imagen clasificada mediante el trabajo de campo realizado por parte de los expertos ecólogos, y obtiene la matriz de confusión entre ambas imágenes, ofreciendo como salida un índice que indica la tasa de acierto del proceso de clasificación, es decir, el porcentaje de píxeles bien clasificados. Además, ofrece una lista de los píxeles mal clasificados, relacionando la clase en la que han sido erróneamente clasificados por parte del algoritmo ACA con la clase a la que realmente pertenecen gracias al trabajo de campo, y de este modo se puede determinar si existe confusión entre clases por estar muy próximas entre sí desde el punto de vista espectral, como vimos al principio de este capítulo.

### 3.4. HERRAMIENTAS UTILIZADAS PARA LA INVESTIGACIÓN

En este apartado se describen las herramientas más importantes utilizadas para desarrollar el trabajo de investigación de esta tesis doctoral: por un lado el programa de tratamiento de imágenes de satélite *Erdas Imagine* y su módulo *Developers Toolkit*, y por otro lado las aplicaciones *Weka* y *MATLAB*.

#### 3.4.1. Erdas Imagine

**Erdas Imagine** [39] [40] es una herramienta software relacionada con el ámbito de la teledetección que proporciona las técnicas más avanzadas en el análisis de imágenes de satélite y modelado espacial para la generación de nueva información. Con *Erdas Imagine* se pueden visualizar los resultados en 2D y 3D, crear vídeos y composiciones de mapa, corregir geoméricamente las imágenes, así como analizar, clasificar y automatizar la realización de mapas generados de imágenes para una gran variedad de aplicaciones. Soporta múltiples formatos compatibles con otras herramientas disponibles en el mercado, permitiendo tanto importar como exportar los resultados obtenidos. Para desarrollar gran parte del trabajo de investigación de esta tesis doctoral se ha usado la *versión 9.2* de *Erdas Imagine*, cuya arquitectura principal es la siguiente:

- ***IMAGINE Essentials***: incorpora los elementos esenciales para la representación de imágenes de satélite en 2D y 3D con un amplio rango de formatos soportados, compatibilidad con otras herramientas como *ArcGIS*, georreferenciación de mapas, análisis simples y composición de productos cartográficos. Se trata de una herramienta de procesamiento de imágenes de satélite de bajo coste.
- ***IMAGINE Advantage***: incorpora sofisticadas herramientas de procesamiento de imágenes, interpolación de superficies, mosaico de imágenes, análisis *SIG* y ortorectificación. Esta herramienta ha sido desarrollada sobre la base de *IMAGINE Essentials* y constituye una solución muy robusta para el procesamiento avanzado de imágenes de satélite.
- ***IMAGINE Professional***: incorpora herramientas más profesionales, como el procesamiento de imágenes de radar, el modelado gráfico espacial de datos o la clasificación avanzada. Incluye todas las herramientas de *IMAGINE Essentials* e *IMAGINE Advantage*, y constituye uno de los sistemas de procesamiento de imágenes más completos para aquellos analistas expertos que requieran sofisticadas herramientas para complejos análisis de imágenes de satélite y fotografías aéreas.

No obstante, existen otros módulos de *Erdas Imagine* que complementan la funcionalidad descrita, como *IMAGINE InSAR*, *Leica Mosaic Pro* o *Terrain Prep Tool*, ampliando las funciones ya disponibles. Se ha elegido la herramienta *Erdas Imagine* para desarrollar el algoritmo ACA por varios motivos. En primer lugar, porque era necesario utilizar una herramienta que permitiera implementar nuevos módulos mediante su desarrollo en algún lenguaje de programación. En este sentido, *Erdas Imagine* dispone del paquete *Developers Toolkit*, que permite al usuario implementar nuevas aplicaciones.



En segundo lugar, porque la herramienta elegida debía ser capaz de manipular diferentes tipos de imágenes de satélite y fotografías aéreas, y además disponer de muchos módulos para poder analizar tanto las imágenes como los resultados. En este aspecto, *Erdas Imagine* ofrece una de las mejores propuestas del mercado, ya que dispone incluso de soporte *online* por si el analista experto o el programador necesita ayuda. En tercer lugar, porque era necesario que la herramienta elegida presentara una buena relación calidad-precio. En este sentido, *Erdas Imagine* proporciona una solución bastante equilibrada, y además el grupo de investigación de Informática Aplicada disponía de una licencia de dicha herramienta. El sistema utilizado para desarrollar el algoritmo ACA ha sido *IMAGINE Professional*, y los principales módulos usados han sido los siguientes:

- **Viewer**: se trata de un módulo que permite visualizar, combinar, analizar y presentar la información geográfica. Es la aplicación que se ha utilizado para mostrar las imágenes de satélite con las que se ha trabajado en esta tesis doctoral. Además, permite obtener una enorme cantidad de información a partir de las mismas.
- **Classifier**: se trata de un módulo que permite realizar tanto clasificaciones no supervisadas para generar el conjunto de entrenamiento, como clasificaciones supervisadas para obtener la imagen final clasificada. Permite utilizar una gran variedad de algoritmos de clasificación de imágenes de satélite. La *Figura 3.8* muestra la interfaz gráfica de *Erdas Imagine 9.2*, con los módulos *Viewer* y *Classifier* activos.

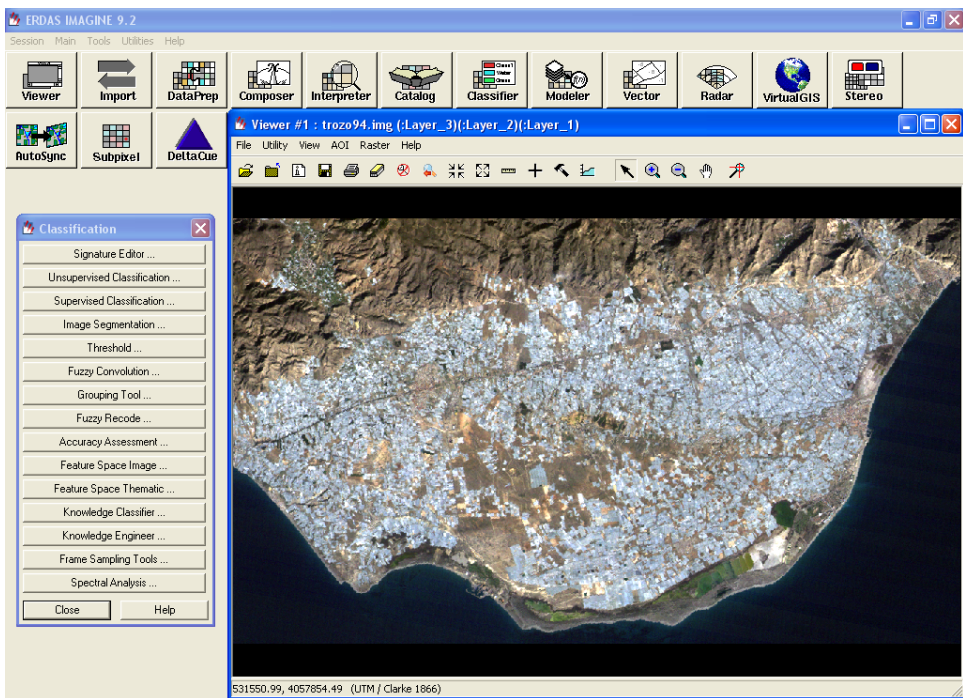


Figura 3.8: Interfaz gráfica de *Erdas Imagine 9.2*. Imagen de satélite: *El Ejido* (1994).

### 3.4.2. Developers Toolkit

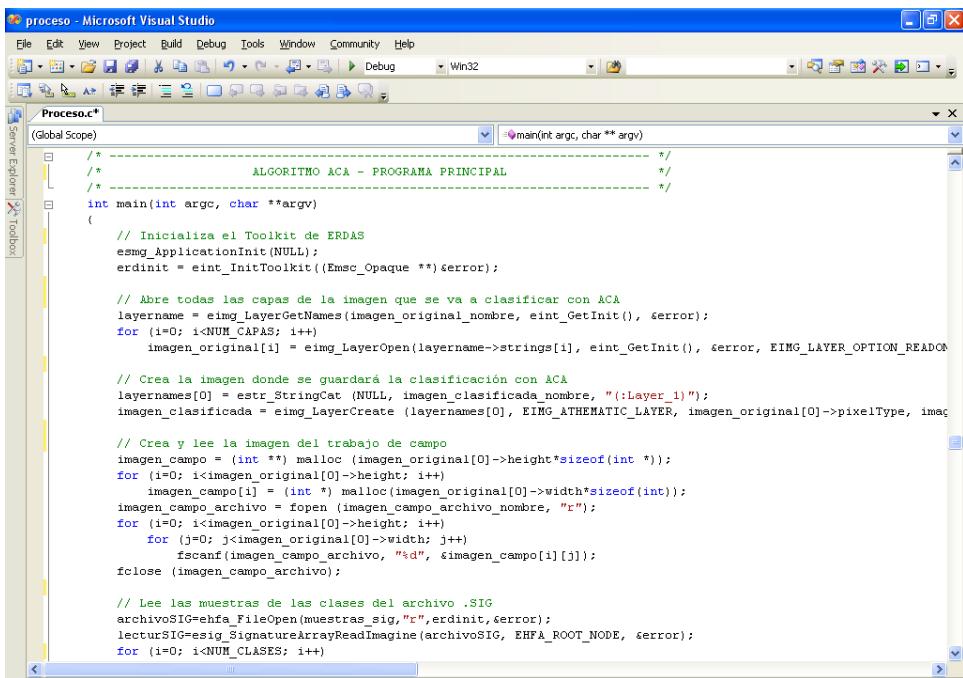
El módulo más importante de *Erdas Imagine* que se ha utilizado para desarrollar el algoritmo ACA ha sido **Developers Toolkit** [55]. Se trata de un conjunto de librerías que permite a los usuarios ampliar la funcionalidad de *Erdas Imagine*, implementando nueva funcionalidad a través de algoritmos informáticos programados en lenguaje C. De este modo, el analista experto puede modificar la versión comercial de *Erdas Imagine*, desarrollando nuevos módulos que se pueden ejecutar como extensión de los que incorpora esta potente herramienta. El *Toolkit* permite integrar las nuevas aplicaciones implementadas por el analista experto en el entorno de *Erdas Imagine*, utilizando las rutinas que ya estaban implementadas en la herramienta. El algoritmo ACA, por lo tanto, ha sido implementado utilizando los siguientes componentes:

- *Erdas Imagine 9.2*.
- *Developers Toolkit*.
- *Compilador Visual C++*.

Por otro lado, la librería de *Developers Toolkit* es muy extensa, y está organizada en distintos paquetes, cada uno compuesto a su vez por un conjunto de funciones. Los paquetes más importantes utilizados para la implementación del algoritmo ACA han sido los siguientes:

- **eint (Master Toolkit Initialization Package)**: es el paquete de inicialización del *Toolkit* de *Erdas*, y gestiona una estructura que tiene un indicador de inicialización para cada paquete de funciones del *Toolkit*. Dicha estructura es creada por la función *eint.InitToolkit*, que es la primera función que se debe llamar en un nuevo programa del *Toolkit* de *Erdas*.
- **eimg (Image File Access Package)**: es el paquete de acceso a los archivos de imágenes de satélite. Se trata de uno de los paquetes más utilizados del *Toolkit*, ya que proporciona las herramientas básicas para crear, leer y escribir los datos *raster* en archivos de disco. Para ello, cada banda de la imagen es definida como un *array* bidimensional, con su propio sistema de memoria virtual independiente. En la implementación del algoritmo ACA, este paquete ha sido utilizado tanto para leer la imagen de satélite a clasificar, como para crear y escribir datos en las distintas imágenes de satélite clasificadas en cada iteración del autómata celular.
- **esig (Signature Package)**: es el paquete de manipulación de los conjuntos de muestras. Permite al analista experto crear, manipular, evaluar, mostrar, mezclar y realizar numerosas tareas sobre uno o varios conjuntos de muestras. Este paquete también se ha utilizado en la implementación del algoritmo ACA, ya que se trata de un algoritmo de clasificación supervisado. En *Erdas Imagine*, los conjuntos de muestras se pueden crear como resultado de algún algoritmo no supervisado, o de manera interactiva seleccionando las muestras de cada clase en base al conocimiento experto humano. Los conjuntos de muestras que se han utilizado en los experimentos del algoritmo ACA han sido elegidos por ecólogos expertos, tomando muestras de cada clase de manera interactiva.

- **eerr (Error Logging and Reporting Package)**: es el paquete que se encarga de informar al usuario de los distintos errores que pueden ocurrir durante la ejecución de las funciones del *Toolkit*. Para ello, se utiliza una estructura que contiene el nombre de la función correspondiente y un código numérico donde se especifica el error cometido dentro de la función. Además, a cada código numérico se le suele asociar una cadena de texto, por lo que el usuario suele recibir el error en formato texto, no numérico. En la implementación del algoritmo ACA, este paquete se ha utilizado en las diferentes funciones del *Toolkit* para informar al analista experto de los distintos errores cometidos.
- **efio (Low Level File I/O Functions)**: es el paquete de manipulación de archivos y directorios, que permite leer, crear, escribir, localizar y encontrar información sobre los mismos. En la implementación del algoritmo ACA, este paquete se ha utilizado para almacenar de manera estructurada los archivos de resultados producidos en las distintas iteraciones del autómata celular.
- **esmg (Session Manager API Package)**: es el paquete que ofrece una interfaz con *Erdas Session Manager*, el administrador de la sesión de *Erdas*. En la implementación del algoritmo ACA, este paquete se ha utilizado para lanzar los procesos que realizan la clasificación supervisada. La *Figura 3.9* muestra una pequeña parte del código fuente del algoritmo ACA en *Visual Studio*.



```

/*
-----
ALGORITMO ACA - PROGRAMA PRINCIPAL
-----
*/
int main(int argc, char **argv)
(
    // Inicializa el Toolkit de ERDAS
    esmg_ApplicationInit(NULL);
    erdinit = eint_InitToolkit((Emsc_Opaque **) &error);

    // Abre todas las capas de la imagen que se va a clasificar con ACA
    layername = eimg_LayerGetNames(imagen_original_nombre, &error);
    for (i=0; i<NUM_CAPAS; i++)
        imagen_original[i] = eimg_LayerOpen(layername->strings[i], eint_GetInit(), &error, EIMG_LAYER_OPTION_READON

    // Crea la imagen donde se guardará la clasificación con ACA
    layernames[0] = estr_StringCat (NULL, imagen_clasificada_nombre, ":(Layer_1)");
    imagen_clasificada = eimg_LayerCreate (layernames[0], EIMG_ATHEMATIC_LAYER, imagen_original[0]->pixelType, imag

    // Crea y lee la imagen del trabajo de campo
    imagen_campo = (int **) malloc (imagen_original[0]->height*sizeof(int *));
    for (i=0; i<imagen_original[0]->height; i++)
        imagen_campo[i] = (int *) malloc (imagen_original[0]->width*sizeof(int));
    imagen_campo_archivo = fopen (imagen_campo_archivo_nombre, "r");
    for (i=0; i<imagen_original[0]->height; i++)
        for (j=0; j<imagen_original[0]->width; j++)
            fscanf (imagen_campo_archivo, "%d", &imagen_campo[i][j]);
    fclose (imagen_campo_archivo);

    // Lee las muestras de las clases del archivo .SIG
    archivoSIG=ehFa_FileOpen(muestras_sig,"r",erdinit,&error);
    lecturSIG=esig_SignatureArrayReadImagine(archivoSIG, EHF&_ROOT_NODE, &error);
    for (i=0; i<NUM_CLASES; i++)

```

Figura 3.9: Código fuente del algoritmo ACA en *Visual Studio*.

### 3.4.3. Weka

**Weka** [56] es una herramienta software que se utiliza para el aprendizaje automático y la minería de datos. Incorpora una gran cantidad de algoritmos de clasificación, pre-procesamiento y análisis de datos, *clustering*, regresión y modelado predictivo, así como aplicaciones de visualización de datos y resultados. La herramienta *Weka* está desarrollada en la *Universidad de Waikato* (Nueva Zelanda), está implementada en *Java*, por lo que es multiplataforma, y se distribuye como software libre bajo licencia *GNU-GPL*. Para desarrollar parte del trabajo de esta tesis doctoral se ha usado la *versión 3.6.10* de *Weka*, cuya interfaz gráfica se presenta al usuario con cuatro opciones principales: *Explorer*, *Experimenter*, *Knowledge Flow* y *Simple CLI*.

El módulo **Explorer** dispone de una gran cantidad de paneles para el preprocesamiento de datos, clasificación, *clustering*, asociación, selección de atributos y visualización de la información. El panel que más se ha utilizado durante el desarrollo de esta tesis doctoral ha sido el de clasificación, con el objetivo de comparar los resultados proporcionados por parte del algoritmo ACA con los de otros algoritmos de clasificación existentes en el mercado, a través del trabajo de campo realizado por expertos ecólogos. El panel de clasificación incorpora una gran cantidad de clasificadores, con la opción de poder configurar todos sus atributos. La *Figura 3.10* muestra el panel de clasificación de la herramienta *Weka* con los resultados obtenidos por el algoritmo *J48 (C4.5)*.

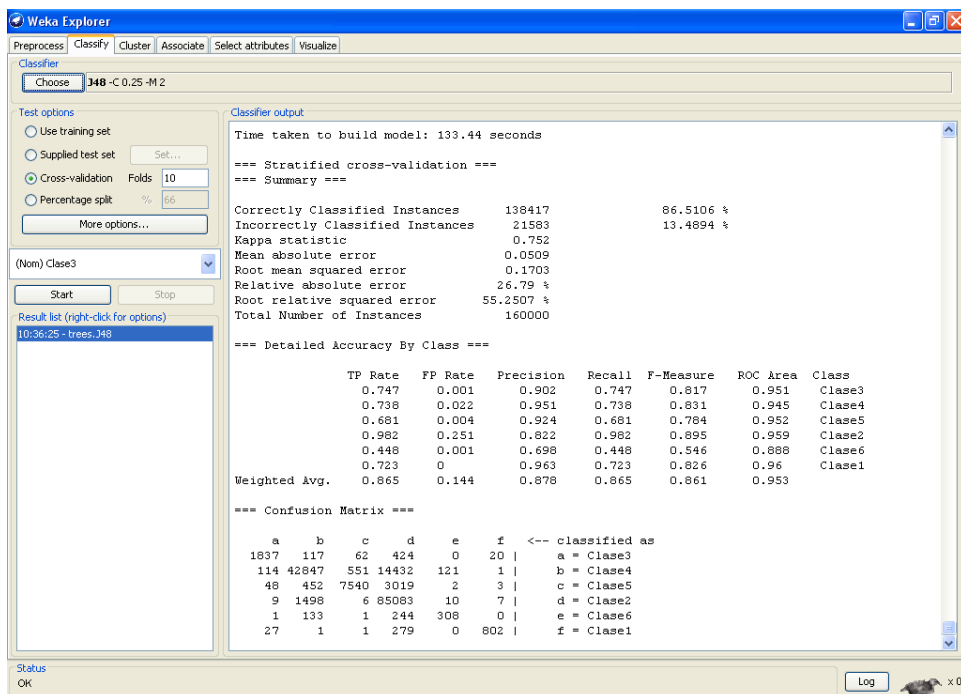


Figura 3.10: Panel de clasificación de *Weka*. Algoritmo *J48 (C4.5)*.

### 3.4.4. MATLAB

**MATLAB** [23] es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (*IDE*) con un lenguaje de programación propio, el *lenguaje M*, de tal modo que el usuario puede programar sus propios *scripts* de código fuente. Entre sus prestaciones básicas destacan la manipulación de matrices, la representación de funciones, la implementación de algoritmos, la creación de interfaces de usuario (*GUI*) y la comunicación con otros lenguajes de programación. Está disponible para varias plataformas, como *Windows*, *Unix* y *Mac OS X*.

Durante el desarrollo de esta tesis doctoral, *MATLAB* se ha utilizado sobre todo para desarrollar algunos trabajos complementarios relacionados con el ámbito de los autómatas celulares, como la implementación de numerosas simulaciones en el ámbito de la teledetección, la simulación de precipitaciones en imágenes de satélite *DEM* y la caracterización de texturas tanto de imágenes-AC como naturales, todo implementado mediante el lenguaje de programación M. Además, también se han utilizado las funciones de *MATLAB* que permiten representar de manera tridimensional las imágenes de satélite *DEM*, permitiendo de este modo elegir la perspectiva más adecuada para visualizar algunos detalles específicos de cada zona de estudio concreta. La *Figura 3.11* muestra la interfaz gráfica de la herramienta *MATLAB*, con el panel de visualización de figuras abierto representando la imagen de satélite *DEM* tridimensional utilizada para realizar la simulación de precipitaciones del algoritmo RACA.

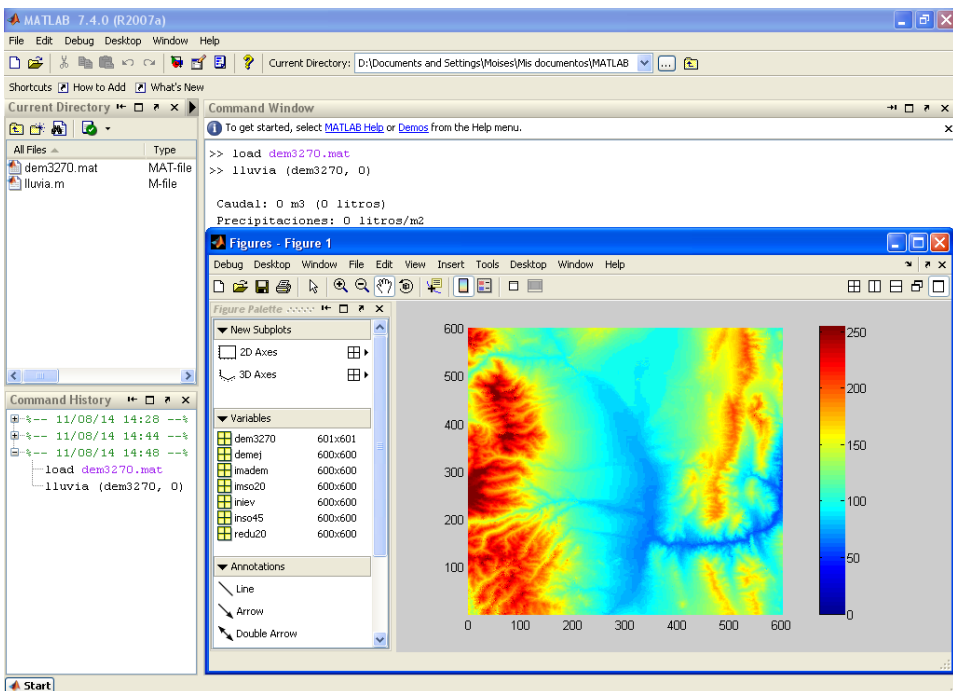


Figura 3.11: Interfaz gráfica de *MATLAB*. Imagen de satélite *DEM*: vista en planta.

### 3.5. RESUMEN Y CONCLUSIONES DEL CAPÍTULO

A pesar del gran número de algoritmos de clasificación de imágenes de satélite que existe en la actualidad, ninguno es 100% eficiente, por lo que aún es necesario seguir investigando para mejorar la tasa de acierto de los algoritmos clásicos. Por otro lado, si nos centramos en los algoritmos de clasificación *paralelepípedos* y *mínima distancia* clásicos, podemos observar que presentan una serie de limitaciones que pueden ser resueltas mediante el uso de autómatas celulares.

En la actualidad existen muy pocos trabajos de investigación relacionados con la clasificación de imágenes de satélite mediante autómatas celulares. Durante el desarrollo de esta tesis doctoral, y desde el marco de trabajo del proyecto *SOLERES*, se ha implementado un algoritmo de clasificación de imágenes de satélite basado en los autómatas celulares, llamado **algoritmo ACA**, que pretende alcanzar tres **objetivos principales**: mejorar la tasa de acierto de la clasificación, construir una clasificación jerárquica, y detectar bordes y píxeles inciertos/ruidosos.

El algoritmo ACA, por lo tanto, pretende utilizar los autómatas celulares no sólo para mejorar la tasa de acierto de la clasificación mediante el uso de técnicas contextuales, sino también para obtener una clasificación iterativa en la que los píxeles clasificados en una iteración determinada sean más fiables que los píxeles clasificados en la siguiente iteración, y así sucesivamente. Esto permitirá obtener una clasificación jerárquica dividida en capas o niveles de fiabilidad, donde cada capa corresponderá con una iteración determinada del autómata celular. Los resultados del proceso de clasificación, agrupados de este modo, podrán ser de mucha utilidad durante el posterior proceso de interpretación de los resultados obtenidos por parte de los analistas expertos.

El algoritmo ACA también pretende obtener, para cada uno de los tres objetivos principales, **información complementaria**: mostrar la evolución de las matrices de confusión intermedias, ofrecer la evolución de la precisión y el *recall*, y mostrar la evolución de las tablas intermedias de píxeles.

Para ello, el autómata celular del algoritmo ACA tiene *tres estados*, uno para cada objetivo principal, y *cuatro reglas*, una para cada tipo de píxel: ruidoso, foco, borde e incierto. Además de la descripción matemática, pseudocódigo y arquitectura, en este capítulo se han ofrecido varias fórmulas complementarias del algoritmo ACA, que permiten obtener no sólo la precisión y el *recall* de las clases para cada iteración del autómata celular, sino también el *grado de pertenencia* de los píxeles a sus clases correspondientes y el *grado de heterogeneidad* de cada clase de la imagen.

Por otro lado, para desarrollar el trabajo de investigación de esta tesis doctoral, se han utilizado principalmente cuatro **herramientas**: *Erdas Imagine*, *Developers Toolkit*, *Weka* y *MATLAB*. El algoritmo ACA ha sido implementado con el *Toolkit de Erdas*, utilizando el lenguaje de programación C. Los algoritmos RACA y TACA, descritos en el *Capítulo 2*, han sido implementados en *MATLAB*, utilizando el lenguaje de programación M. La comparativa del algoritmo ACA con otros algoritmos de clasificación existentes en el mercado, que se muestra en el siguiente capítulo, se ha llevado a cabo gracias a *Weka*. El *Capítulo 4* ofrece una amplia visión de los resultados obtenidos por parte del algoritmo ACA.

---

## CAPÍTULO 4

### RESULTADOS OBTENIDOS

---





# Capítulo 4

## RESULTADOS OBTENIDOS

### Contenidos

---

<b>4.1.</b>	<b>RESULTADOS PRINCIPALES DE ACA . . . . .</b>	<b>135</b>
4.1.1.	Características experimentales . . . . .	136
4.1.2.	Objetivo #1: mejorar la tasa de acierto de la clasificación . . .	142
4.1.3.	Objetivo #2: construir una clasificación jerárquica . . . . .	145
4.1.4.	Objetivo #3: detectar bordes y píxeles inciertos/ruidosos . . .	147
4.1.5.	Conclusiones . . . . .	151
<b>4.2.</b>	<b>RESULTADOS COMPLEMENTARIOS DE ACA . . . . .</b>	<b>152</b>
4.2.1.	Características experimentales . . . . .	153
4.2.2.	Información #1: evolución de las matrices de confusión . . . .	154
4.2.3.	Información #2: evolución de la precisión y el <i>recall</i> . . . . .	158
4.2.4.	Información #3: evolución de las tablas de píxeles . . . . .	166
4.2.5.	Conclusiones . . . . .	167
<b>4.3.</b>	<b>FORMATO DE SALIDA DEL ALGORITMO ACA . . . . .</b>	<b>168</b>
4.3.1.	Índice completo de los archivos de salida . . . . .	168
4.3.2.	Clasificación jerárquica: número de iteraciones y color . . . .	169
4.3.3.	Matrices de confusión intermedias por iteraciones . . . . .	173
4.3.4.	Datos de las gráficas evolutivas: acierto, precisión y <i>recall</i> . .	174
<b>4.4.</b>	<b>RESUMEN Y CONCLUSIONES DEL CAPÍTULO . . . . .</b>	<b>175</b>

---



**E**l algoritmo ACA, para la clasificación supervisada de imágenes de satélite mediante autómatas celulares, ha sido implementado, experimentado y validado en el marco de trabajo de los proyectos *SOLERES* [83] e *iSOLERES* [85]. Los resultados obtenidos por parte del algoritmo ACA han sido muy satisfactorios desde varios puntos de vista, alcanzando todos los objetivos principales propuestos. También se han obtenido unos resultados complementarios, que suponen una ampliación de los resultados principales. Este capítulo ofrece una amplia visión de los resultados principales y complementarios obtenidos por parte del algoritmo ACA, y está dividido en los siguientes apartados principales:

- **Resultados principales de ACA:** en este apartado se realiza una comparativa del algoritmo ACA con otros algoritmos de clasificación que existen en la literatura, empleando para ello trabajo de campo y tres imágenes de satélite de distintas regiones de la provincia de Almería: *Níjar*, *El Ejido Oeste* y *El Ejido Este*. Posteriormente se analizan los resultados obtenidos para cada uno de los tres objetivos propuestos, y para finalizar se ofrecen unas conclusiones.
- **Resultados complementarios de ACA:** en este apartado nos centramos en la obtención de la información complementaria que ofrece el algoritmo ACA para cada iteración del proceso de clasificación. Se trata de una ampliación de los resultados del apartado anterior, concretamente de la imagen de satélite de *El Ejido Oeste*. Posteriormente se analizan los resultados obtenidos por parte de dicha información complementaria, y se ofrecen unas conclusiones.
- **Formato de salida del algoritmo ACA:** en este apartado se muestra el índice completo de los archivos de salida del algoritmo ACA, la configuración de la clasificación jerárquica según el número de iteraciones y el color de las clases, y el formato de salida de las matrices de confusión intermedias y de los datos de las gráficas de evolución de la tasa de acierto, la precisión y el *recall*.
- **Resumen y conclusiones del capítulo:** en este apartado se destacan los aspectos más importantes de los resultados principales y complementarios obtenidos por parte del algoritmo ACA.

## 4.1. RESULTADOS PRINCIPALES DE ACA

En este apartado se realiza una comparativa del algoritmo ACA con otros algoritmos de clasificación que existen en la literatura, empleando para ello trabajo de campo y tres imágenes de satélite de distintas regiones de la provincia de Almería: *Níjar*, *El Ejido Oeste* y *El Ejido Este*. Posteriormente se analizan los resultados obtenidos para cada uno de los tres objetivos propuestos, y para finalizar se ofrecen unas conclusiones.

#### 4.1.1. Características experimentales

La comparativa del algoritmo ACA con otros algoritmos de clasificación se ha llevado a cabo sobre 3 imágenes de satélite *Landsat TM* multiespectrales de 7 bandas, con una resolución espacial de  $30 \times 30$  m. Las regiones de estudio elegidas han sido *Níjar*, *El Ejido Oeste* y *El Ejido Este*, pertenecientes a la provincia de Almería. El tamaño de todas las imágenes es de  $400 \times 400$  píxeles (un total de 160000 píxeles). La *Figura 4.1* muestra las imágenes de satélite de *Níjar*, *El Ejido Oeste* y *El Ejido Este* (bandas 3, 2, 1).

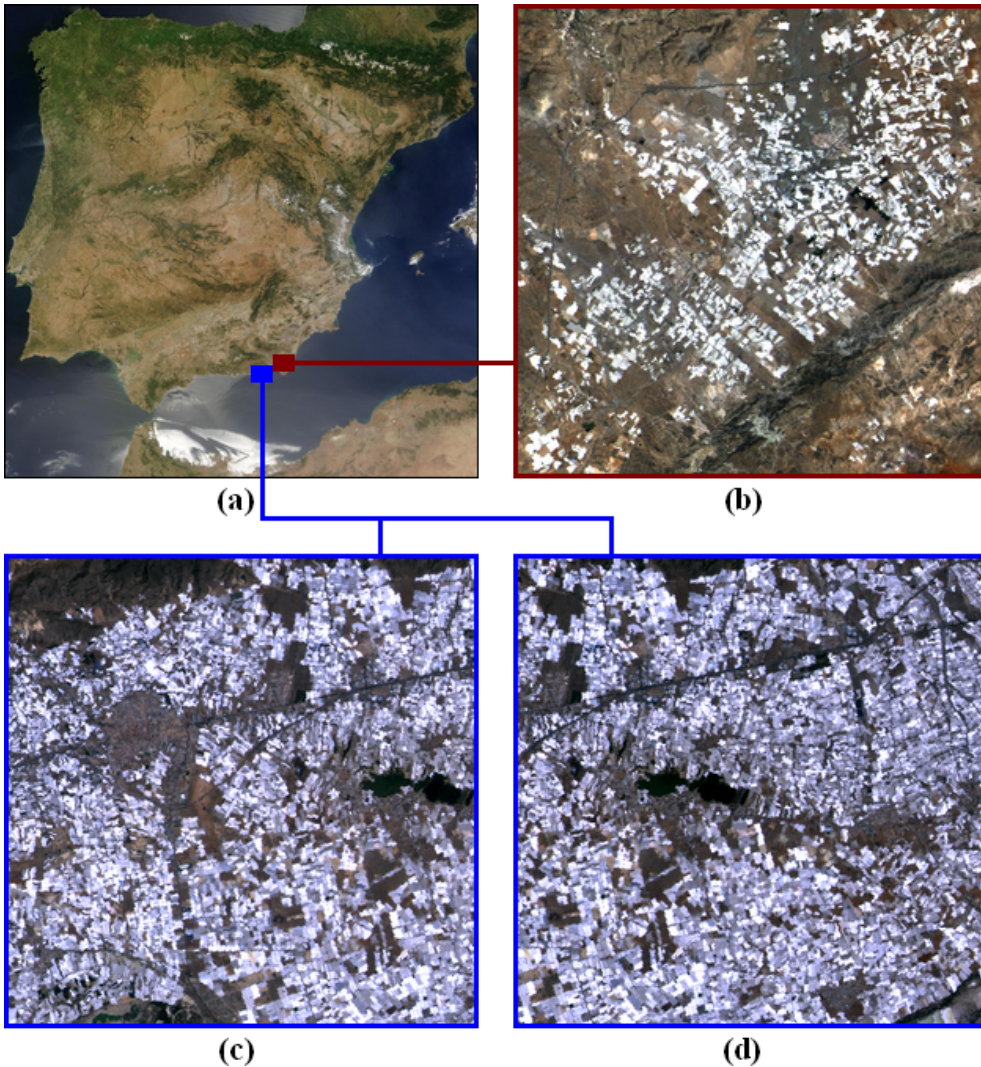


Figura 4.1: (a) Regiones de estudio de la provincia de Almería, España (abril de 2003). (b) *Níjar*. (c) *El Ejido Oeste*. (d) *El Ejido Este*.

La enorme diversidad de vegetación y tipos de suelo que existe en el sureste de España complica bastante la verificación de cualquier algoritmo de clasificación, como ya se mostró en anteriores experimentos del proyecto *SOLERES* [9] [10]. La *Figura 4.2* muestra los 6 **sectores principales de vegetación** del sureste de España [30] [31], donde se han marcado las regiones de estudio seleccionadas para este trabajo.

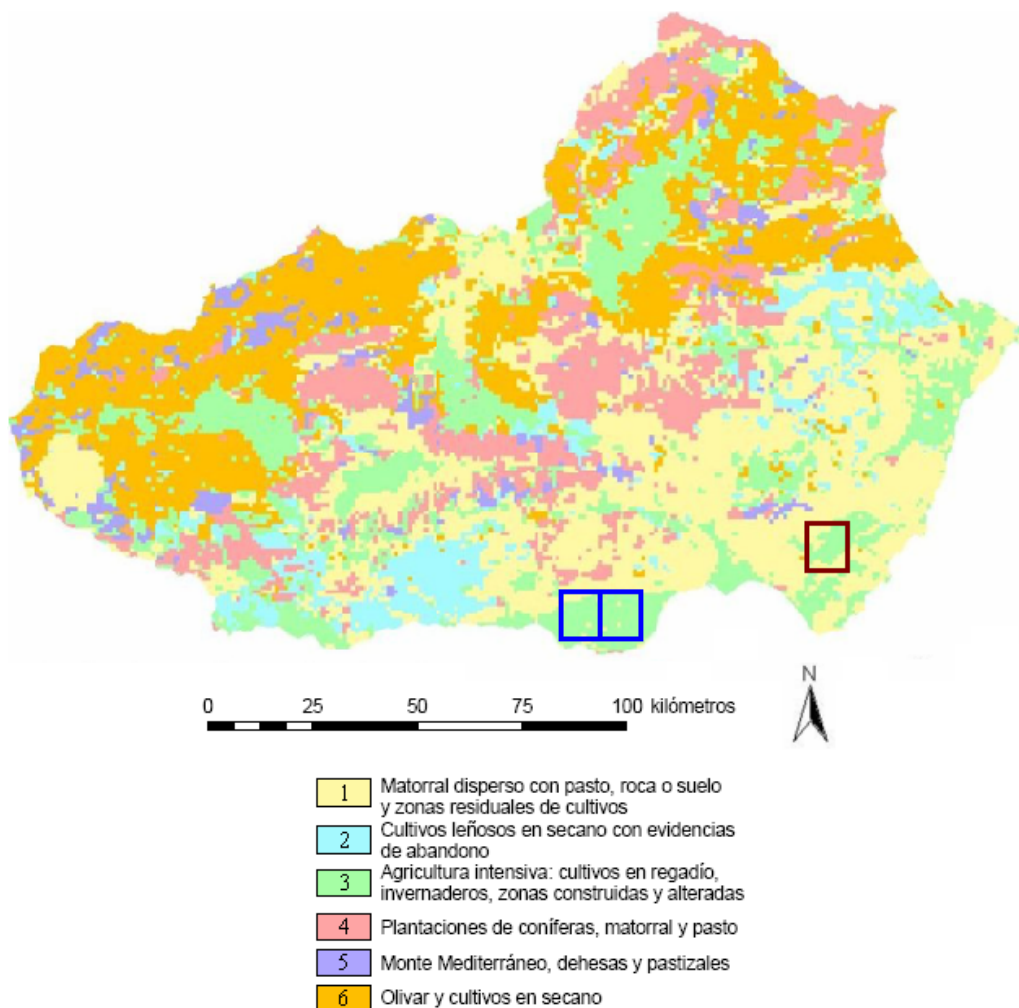


Figura 4.2: Sectores de vegetación del sureste de España. *Níjar* (cuadrado rojo) y *El Ejido* (cuadrados azules).

Esta gran diversidad ecológica provoca que muchas clases se encuentren muy próximas entre sí desde el punto de vista espectral. Además, para intensificar aún más la variabilidad, estos 6 sectores principales de vegetación están divididos, a su vez, en múltiples **variables ecológicas de vegetación**, como se muestra en la *Tabla 4.1*.

Sector	Variables ecológicas de vegetación de los sectores del sureste de España
1	Matorral denso Matorral disperso con pasto y roca o suelo Otros mosaicos de cultivos y vegetación natural
2	Cultivos leñosos abandonados Cultivos leñosos y vegetación natural leñosa Mosaico de cultivos leñosos en secano Mosaico de cultivos leñosos en regadío Otros cultivos leñosos en secano Matorral denso + coníferas dispersas
3	Carretera Invernadero Vegetación riparia Matorral denso Matorral disperso con pastizal Pastizal continuo Pastizal con claros (roca, suelo) Cultivos leñosos en regadío Cultivos leñosos y pastizales Mosaico de cultivos leñosos en regadío Cultivos herbáceos y pastizales Cultivos herbáceos y leñosos en regadío Cultivos herbáceos y vegetación natural leñosa Otros cultivos herbáceos en regadío Mosaico de cultivos en secano y en regadío Otras especies arbóreas y mezclas Playas, dunas y arenales Zonas húmedas y superficies de agua Suelo desnudo sin vegetación Superficies construidas y alteradas Zonas con fuertes procesos erosivos o incendiadas
4	Coníferas Pastizal + coníferas densas Pastizal + coníferas dispersas Pastizal + otras especies arbóreas o mezclas Matorral denso + coníferas densas Matorral denso + coníferas dispersas Matorral denso + otras especies arbóreas o mezclas Matorral disperso + coníferas densas Matorral disperso + coníferas dispersas Matorral disperso + otras especies arbóreas o mezclas
5	Quercíneas Pastizal continuo Pastizal + quercíneas densas Pastizal + quercíneas dispersas Matorral denso + quercíneas densas Matorral denso + quercíneas dispersas Matorral disperso + quercíneas densas Matorral disperso + quercíneas dispersas
6	Olivar Cultivos herbáceos en secano Cultivos herbáceos y leñosos en secano

Tabla 4.1: Variables ecológicas de vegetación de los 6 sectores principales.

En concreto, las regiones de *Níjar* y *El Ejido* están caracterizadas por la presencia de los *sectores de vegetación 1 y 3*, y se pueden etiquetar en **6 clases espectrales** distintas, algunas de ellas con un alto *grado de heterogeneidad*, es decir, con un alto porcentaje de píxeles inciertos. Sobre todo, *El Ejido* se caracteriza por la presencia de una enorme cantidad de invernaderos. Existen tantos invernaderos en *El Ejido*, que pueden ser vistos desde el espacio exterior desde la *ISS*, incluso en imágenes de satélite con baja resolución espacial. Por otro lado, *Níjar* también tiene invernaderos, aunque en menor cantidad.

El proceso de clasificación de los píxeles de la clase ‘*invernadero*’ es muy complicado, ya que se trata de una clase con un grado de heterogeneidad extremadamente alto, y por lo tanto sus píxeles pueden cubrir un amplio rango espectral en el espacio de características, debido a los diferentes materiales de construcción utilizados, tales como el plástico o el policarbonato, así como la reflectancia de cada tipo de material dependiendo de las características atmosféricas. Como consecuencia, los algoritmos de clasificación clásicos suelen fallar con este tipo de píxeles [1] [21]. Estas regiones se caracterizan además por la presencia de la clase ‘*superficies construidas y alteradas*’, es decir, zona urbana, que también se trata de una clase bastante heterogénea porque agrupa píxeles con diferentes características espectrales, como edificios y suelo desnudo. Los algoritmos de clasificación clásicos también suelen fallar con los píxeles de esta clase. Las imágenes de satélite de *Níjar* y *El Ejido* se caracterizan además por la presencia de otras clases de heterogeneidad media, como ‘*pastizal continuo*’, ‘*matorral disperso con pasto y roca o suelo*’ y ‘*zonas húmedas y superficies de agua*’. Finalmente, estas regiones también se caracterizan por la presencia de la clase ‘*carretera*’, con un bajo nivel de heterogeneidad. La *Tabla 4.2* muestra las características de las 6 clases que se pueden encontrar en las imágenes de satélite de *Níjar* y *El Ejido*.

Clase	Sector	Descripción: variable ecológica	Heterogeneidad
<i>C1</i>	3	Zonas húmedas y superficies de agua	Media
<b><i>C2</i></b>	<b>3</b>	<b>Invernaderos</b>	<b>Muy alta</b>
<i>C3</i>	3	Pastizal continuo	Media
<b><i>C4</i></b>	<b>3</b>	<b>Superficies construidas y alteradas</b>	<b>Alta</b>
<i>C5</i>	1	Matorral disperso con pasto y roca o suelo	Media
<i>C6</i>	3	Carretera	Baja

Tabla 4.2: Características principales de las 6 clases de *Níjar* y *El Ejido*.

En resumen, las regiones de *Níjar* y *El Ejido* se caracterizan por la presencia de 1 clase con una heterogeneidad muy alta, 1 clase con una heterogeneidad alta, 3 clases con una heterogeneidad media y 1 clase con una heterogeneidad baja. Es decir, existen 2 clases con un elevado número de píxeles inciertos, lo que complica el proceso de clasificación de cualquier algoritmo. El *Apéndice A* muestra una ampliación del trabajo de campo realizado a las tres regiones de estudio elegidas. Aparte de estos inconvenientes, muchas clases están muy próximas entre sí desde un punto de vista espectral, como se muestra en la *Figura 4.3* con la siguiente configuración de colores: *C1* (azul), *C2* (blanco), *C3* (verde), *C4* (marrón claro), *C5* (marrón oscuro) y *C6* (gris).

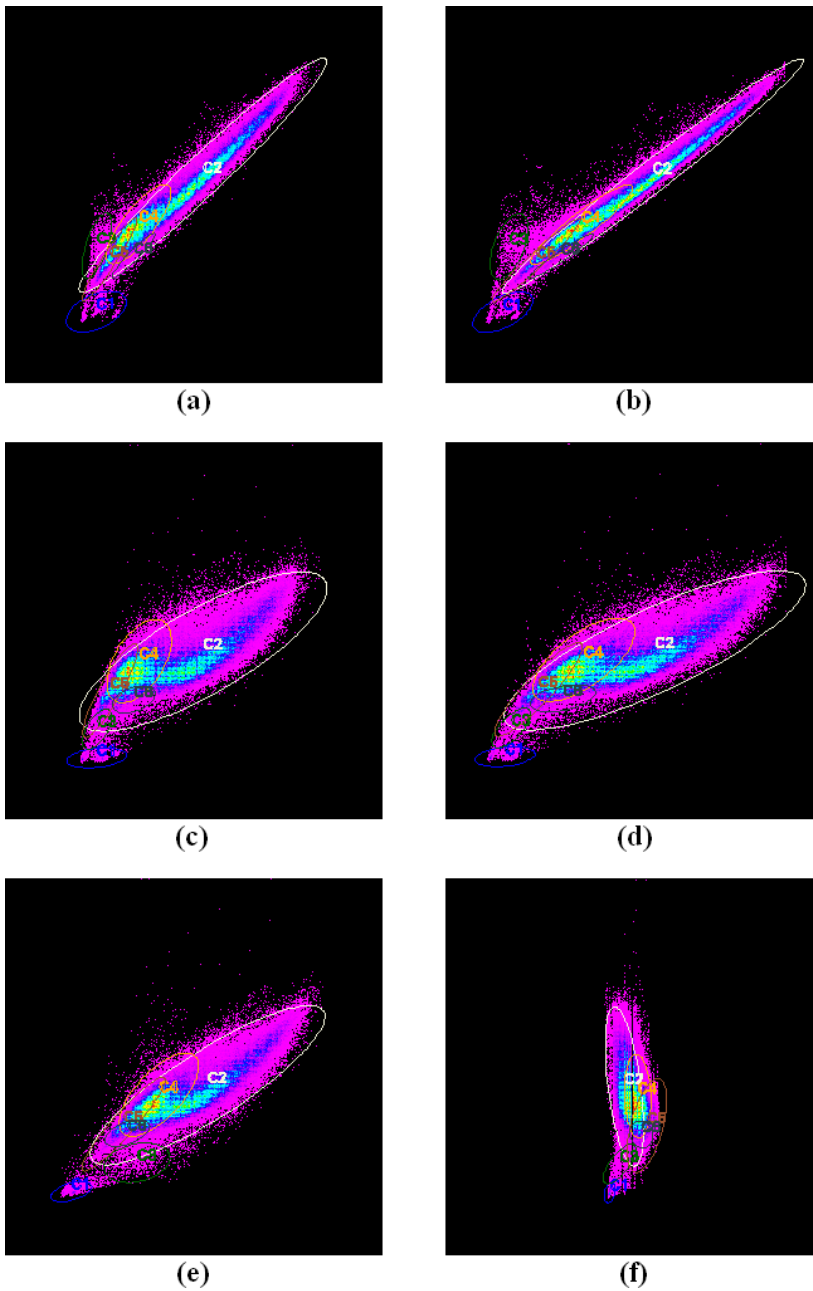


Figura 4.3: Gráficas espectrales de la imagen de satélite de *El Ejido Oeste*: (a) Bandas 2-4. (b) Bandas 3-4. (c) Bandas 2-7. (d) Bandas 3-7. (e) Bandas 4-7. (f) Bandas 6-7.



No obstante, las imágenes de satélite de *El Ejido* son más difíciles de clasificar que las de *Níjar*, ya que existen muchos más invernaderos en *El Ejido* que en *Níjar*, y los píxeles de dicha clase son los que tienen mayor grado de heterogeneidad. Además, la región de *El Ejido Este* tiene una mayor **concentración de invernaderos** que *El Ejido Oeste*. Concretamente, la imagen de *Níjar* tiene un 18 % de píxeles de invernadero, la imagen de *El Ejido Oeste* un 55 % y la imagen de *El Ejido Este* un 71 %. Como la clase invernadero es la que tiene mayor porcentaje de píxeles inciertos, que son los píxeles más complicados de clasificar correctamente, la imagen que ofrece mayor dificultad para los algoritmos de clasificación es la de *El Ejido Este*.

Para aumentar aún más el grado de complejidad del proceso de clasificación en la región de *El Ejido*, se ha añadido artificialmente un 1 % y 2 % de **concentración de ruido** a las imágenes del *Oeste* y *Este*, respectivamente. Dicho ruido es de tipo *impulsivo-gaussiano* aleatorio, y se ha aplicado a las imágenes de satélite antes de utilizar el algoritmo ACA. Se han elegido dichos porcentajes tan bajos porque, en el caso de que una imagen de satélite tenga ruido, el porcentaje de píxeles ruidosos no suele ser demasiado elevado. Sin embargo, el algoritmo ACA suele mejorar el proceso de clasificación de este tipo de píxeles, ya que utiliza *técnicas contextuales* a través de las reglas del autómata celular para tener en cuenta la media de la vecindad de los píxeles ruidosos, y por lo tanto casi siempre acierta en la asignación de la clase a un píxel ruidoso. Además, conforme aumenta el porcentaje de ruido, el porcentaje de mejora en la tasa de acierto del algoritmo ACA también aumenta respecto al resto de algoritmos clásicos, que suelen fallar en el etiquetado de este tipo de píxeles.

Con las características de estas tres imágenes de satélite se han establecido tres **niveles de complejidad** en el proceso de clasificación: *Níjar* (bajo), *El Ejido Oeste* (medio) y *El Ejido Este* (alto), con el objetivo de comparar los resultados de acierto del algoritmo ACA respecto a otros algoritmos de clasificación. La *Tabla 4.3* resume las características de las tres imágenes de satélite usadas para realizar dicho estudio comparativo.

Imagen de satélite	Ruido	Invernaderos	Complejidad de clasificación
<i>Níjar</i>	0 %	18 %	<b>Bajo</b>
<i>El Ejido Oeste</i>	1 %	55 %	<b>Medio</b>
<i>El Ejido Este</i>	2 %	71 %	<b>Alto</b>

Tabla 4.3: Características de *Níjar*, *El Ejido Oeste* y *El Ejido Este*.

La aplicación del algoritmo ACA a estas tres imágenes de satélite intenta alcanzar los objetivos descritos a lo largo del *Capítulo 3*: mejorar la tasa de acierto de la clasificación utilizando técnicas contextuales (*objetivo #1*), construir una clasificación jerárquica dividida en niveles de fiabilidad basados en el grado de pertenencia a las clases (*objetivo #2*) y detectar bordes espaciales así como píxeles inciertos y ruidosos (*objetivo #3*). Las siguientes secciones analizan los resultados obtenidos para cada uno de los tres objetivos principales propuestos.

### 4.1.2. Objetivo #1: mejorar la tasa de acierto de la clasificación

Como ya se ha comentado, el alto grado de heterogeneidad de ciertas clases de las regiones de estudio complican bastante la clasificación de sus píxeles y, en consecuencia, disminuye la tasa de acierto de los algoritmos de clasificación aplicados a estas imágenes de satélite. Sin embargo, la tasa de acierto del algoritmo ACA es mayor que la de otros algoritmos de clasificación supervisados, ya que utiliza *técnicas contextuales* en la función de transición  $f$  con el objetivo de mejorar la clasificación en los píxeles inciertos y ruidosos, mejorando de este modo la tasa de acierto final.

El algoritmo ACA **mejora el funcionamiento** de los algoritmos *paralelepípedos* y *mínima distancia* clásicos. Durante las primeras iteraciones del autómata celular sólo existen píxeles bien clasificados en la matriz de confusión, debido a que el *threshold* es muy bajo. La mayoría de los píxeles que se clasifican durante estas primeras iteraciones incluso pertenecen al conjunto de entrenamiento definido por los ecólogos expertos. Conforme el algoritmo ACA continúa su ejecución, aparece un punto en el que ya han sido clasificados todos los píxeles de las clases con heterogeneidad baja y media ( $C1$ ,  $C3$ ,  $C5$  y  $C6$ ), y a partir de este punto sólo faltan por clasificar algunos píxeles de las clases con alto grado de heterogeneidad ( $C2$  y  $C4$ ). Si establecemos un número total de 100 iteraciones para clasificar cada una de las tres imágenes de satélite con el algoritmo ACA, este evento ocurre aproximadamente en la iteración 40 en los tres procesos de clasificación. Durante las siguientes 60 iteraciones, el algoritmo ACA va refinando el proceso de clasificación para mejorar aún más los resultados, clasificando correctamente los píxeles que ofrecen más problemas: los inciertos (de las clases  $C2$  y  $C4$ ) y los ruidosos (repartidos por todas las clases). Cuando el algoritmo ACA llega a la iteración 100, todos los píxeles de la imagen de satélite han sido clasificados. Por lo tanto, el algoritmo ACA ha resuelto uno de los principales problemas del algoritmo *paralelepípedos* clásico, ya que, gracias al uso de autómatas celulares, no quedan píxeles sin clasificar al final del proceso, debido a que el *threshold* aumenta en cada iteración. Además, el algoritmo ACA ha agrupado los procesos de pre-clasificación (reducción de ruido), clasificación y post-clasificación (refinamiento de píxeles inciertos), en un solo proceso.

El uso de autómatas celulares no sólo mejora el funcionamiento de los algoritmos de clasificación clásicos, sino también **mejora la tasa de acierto**: el algoritmo ACA *paralelepípedos* mejora la tasa de acierto del algoritmo *paralelepípedos* clásico (un 4,82 % en *Níjar*, un 8,10 % en *El Ejido Oeste* y un 15,73 % en *El Ejido Este*), del mismo modo que el algoritmo ACA *mínima distancia* mejora la tasa de acierto del algoritmo *mínima distancia* clásico (un 3,31 % en *Níjar*, un 3,92 % en *El Ejido Oeste* y un 9,71 % en *El Ejido Este*).

En la **comparativa** realizada de la tasa de acierto del algoritmo ACA respecto a la de otros cinco algoritmos de clasificación ampliamente usados en la literatura ( $C4.5$  [87] [88], *multilayer perceptron* [92] [93], *naive Bayes* [36] [75], *k-NN* [2] [47] y *RBF network* [19] [20]), observamos que el algoritmo ACA proporciona mejores resultados en las imágenes más complejas de clasificar. Para realizar dicha comparativa se ha usado una imagen de campo de cada región de estudio, realizada por ecólogos expertos en investigaciones anteriores del proyecto *SOLERES*. Todos los algoritmos han sido evaluados usando *10-fold cross-validation*. La *Tabla 4.4* muestra los datos de la comparativa.

Algoritmo	<i>Níjar</i>	<i>El Ejido Oeste</i>	<i>El Ejido Este</i>
<i>ACA paralelepípedos</i>	89,15 %	83,12 %	<b>82,01 %</b>
<i>C4.5</i>	<b>94,67 %</b>	<b>86,51 %</b>	80,10 %
<i>ACA mínima distancia</i>	88,36 %	80,10 %	78,87 %
<i>Multilayer perceptron</i>	<b>90,62 %</b>	81,40 %	76,73 %
<i>Naive Bayes</i>	87,87 %	78,57 %	72,57 %
<i>k-NN (k=3)</i>	85,95 %	77,02 %	71,20 %
<i>RBF network</i>	85,14 %	76,31 %	69,48 %

Tabla 4.4: Tasa de acierto de los algoritmos de clasificación.

La comparativa realizada sobre la imagen de satélite de *Níjar*, con complejidad de clasificación baja, muestra que los algoritmos *C4.5* y *multilayer perceptron* mejoran la tasa de acierto del algoritmo *ACA paralelepípedos* en 5,52 % y 1,47 %, respectivamente. Sin embargo, aunque los algoritmos *C4.5* y *multilayer perceptron* tienen una mayor tasa de acierto que el algoritmo ACA en imágenes de satélite con complejidad de clasificación baja, no ofrecen la enorme cantidad de información que proporciona el algoritmo ACA: clasificación jerárquica dividida en niveles de proximidad espectral a las clases y detección de bordes, píxeles inciertos y ruidosos. De este modo, incluso con esta tasa de acierto, todavía es recomendable utilizar el algoritmo ACA, en lugar de los algoritmos *C4.5* y *multilayer perceptron*, si se desea obtener una mayor cantidad de información a partir del proceso de clasificación. El algoritmo *ACA paralelepípedos* mejora la tasa de acierto del algoritmo *ACA mínima distancia* en 0,79 %, y el algoritmo *ACA mínima distancia* mejora en 0,49 % la tasa de acierto del algoritmo *naive Bayes*, en 2,41 % la de *k-NN* con  $k=3$ , y en 3,22 % la de *RBF network*. De las tres imágenes de satélite elegidas para realizar esta comparativa, la imagen de *Níjar* es en la que el algoritmo ACA obtiene los peores resultados, ya que otros dos algoritmos obtienen mejor tasa de acierto.

La comparativa realizada sobre la imagen de satélite de *El Ejido Oeste*, con complejidad de clasificación media, muestra que sólo el algoritmo *C4.5* mejora la tasa de acierto del algoritmo *ACA paralelepípedos* en 3,39 %. El algoritmo *ACA paralelepípedos* mejora la tasa de acierto del algoritmo del algoritmo *multilayer perceptron* en 1,72 %, que a su vez mejora la tasa de acierto del algoritmo *ACA mínima distancia*. A continuación, el algoritmo *ACA mínima distancia* mejora en 1,53 % la tasa de acierto del algoritmo *naive Bayes*, en 3,08 % la de *k-NN* con  $k=3$ , y en 3,79 % la de *RBF network*. La imagen de satélite de *El Ejido Oeste* muestra que la tasa de acierto del algoritmo ACA es mejor que la del resto de algoritmos, a excepción del algoritmo *C4.5*.

La comparativa realizada sobre la imagen de satélite de *El Ejido Este*, con complejidad de clasificación alta, muestra que el algoritmo *ACA paralelepípedos* mejora la tasa de acierto del resto de algoritmos, sin excepciones. La tasa de acierto del algoritmo *ACA paralelepípedos* mejora en 1,91 % la del algoritmo *C4.5*, en 5,28 % la del algoritmo *multilayer perceptron*, en 9,44 % la del algoritmo *naive Bayes*, en 10,81 % la del algoritmo *k-NN* con  $k=3$ , y en 12,53 % la del algoritmo *RBF network*. La imagen de satélite de *El Ejido Este* muestra que el algoritmo ACA obtiene los mejores resultados.

La *Figura 4.4* muestra la **evolución de la tasa de acierto** de los algoritmos de clasificación en las tres imágenes de satélite. Se puede observar que la tasa de acierto de los dos algoritmos ACA ofrecen una mejor evolución que el resto, ya que decrecen de manera más lenta conforme aumenta la complejidad de clasificación.

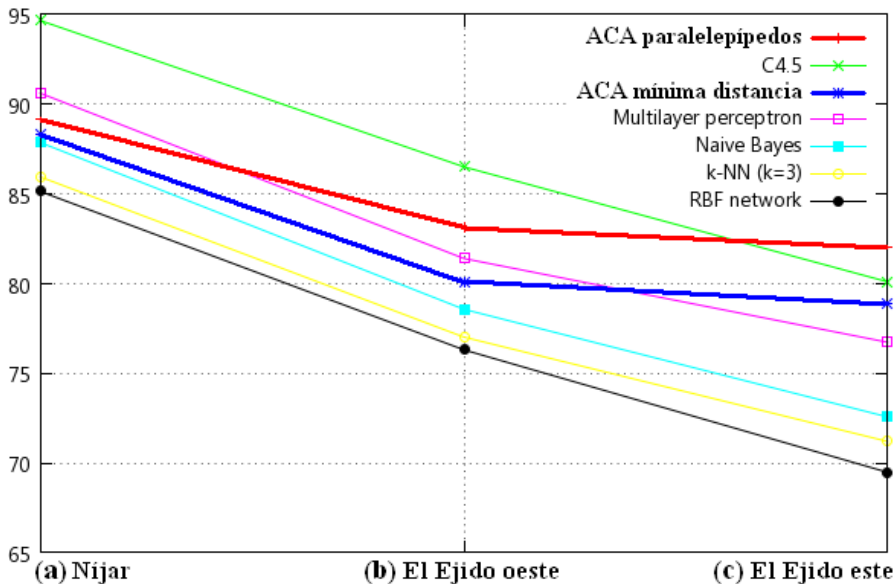


Figura 4.4: Evolución de la tasa de acierto de los algoritmos de clasificación: (a) *Níjar*. (b) *El Ejido Oeste*. (c) *El Ejido Este*.

En lo que respecta al **coste computacional**, el algoritmo ACA tiene una complejidad media, ya que no es tan rápido como los algoritmos *k-NN*, *naive Bayes* o *multilayer perceptron*, pero tampoco es tan lento como los algoritmos *C4.5* ó *RBF network*. La *Tabla 4.5* muestra el coste computacional de todos los algoritmos de clasificación.

Algoritmo	Coste computacional
<i>k-NN</i> (k=3)	$O(n \cdot \log n)$
<i>Naive Bayes</i>	$O(m \cdot n)$
<i>Multilayer perceptron</i>	$O(n^2)$
<b>ACA paralelepípedos</b>	$O(n \cdot i + \log n)$
<b>ACA mínima distancia</b>	$O(n \cdot i + \log n)$
<i>C4.5</i>	$O(m \cdot n^2)$
<i>RBF network</i>	$O(n^3)$

Tabla 4.5: Coste computacional de los algoritmos de clasificación (donde:  $n$  es el número de instancias de entrenamiento,  $m$  es el número de atributos e  $i$  es el número de iteraciones del autómata celular).

### 4.1.3. Objetivo #2: construir una clasificación jerárquica

El algoritmo ACA produce una clasificación jerárquica dividida en distintos niveles o capas de fiabilidad en base al grado de pertenencia a cada clase, teniendo en cuenta la proximidad espectral de los píxeles a sus clases correspondientes dentro del espacio de características. En cada iteración del autómata celular, el algoritmo ACA clasifica sólo aquellos píxeles de la imagen que no han sido clasificados aún y que están dentro del *threshold* correspondiente a la iteración actual. Además, el *threshold* va aumentando en cada iteración. De este modo, los píxeles que se encuentran más cerca de sus clases, desde un punto de vista espectral, son clasificados en las primeras iteraciones. En las siguientes iteraciones se van clasificando los píxeles que están más alejados. Finalmente, los píxeles inciertos y ruidosos son clasificados en las últimas iteraciones, utilizando técnicas contextuales, que se basan en píxeles vecinos correctamente clasificados en iteraciones previas. Por lo tanto, los primeros niveles de la clasificación jerárquica son más fiables, en términos de tasa de acierto, que los últimos, ya que sus píxeles están más próximos a sus clases, a nivel espectral. La clasificación jerárquica simula una clasificación *pseudodifusa*, ya que permite a los analistas expertos calcular el **grado de pertenencia** de cada píxel a su clase correspondiente, y además permite conocer el **grado de heterogeneidad** de cada clase, como se muestra en las siguientes fórmulas:

$$gp_{x,A} = \frac{\textit{iteración}_{A,fin} - \textit{iteración}_{x,clase(A)}}{\textit{iteración}_{A,fin}} \quad (4.1)$$

donde:

- $gp_{x,A}$ : grado de pertenencia del píxel  $x$  a la clase  $A$ .
- $\textit{iteración}_{A,fin}$ : iteración en la que todos los píxeles de la clase  $A$  se han clasificado.
- $\textit{iteración}_{x,clase(A)}$ : iteración en la que el píxel  $x$  ha sido clasificado en la clase  $A$ .

$$gh_A = \frac{\textit{iteración}_{A,fin}}{\textit{iteración}_{im,fin}} \quad (4.2)$$

donde:

- $gh_A$ : grado de heterogeneidad de la clase  $A$ .
- $\textit{iteración}_{A,fin}$ : iteración en la que todos los píxeles de la clase  $A$  se han clasificado.
- $\textit{iteración}_{im,fin}$ : iteración en la que todos los píxeles de la imagen se han clasificado.

Como se puede observar en la clasificación jerárquica obtenida, la mayoría de los píxeles son clasificados en las primeras 40 iteraciones, y durante las siguientes se va refinando el proceso de clasificación. En la iteración 40, las dos imágenes de satélite de *El Ejido* tienen un porcentaje de píxeles sin clasificar mayor que la imagen de satélite de *Níjar*, ya que el número de píxeles que pertenecen a la clase invernadero es mayor en *El Ejido*. En las siguientes 60 iteraciones, el algoritmo ACA va mejorando los resultados utilizando técnicas contextuales para clasificar los píxeles inciertos y ruidosos.

La *Figura 4.5* muestra la clasificación jerárquica de las tres imágenes de satélite después de 100 iteraciones del algoritmo *ACA mínima distancia* con la siguiente configuración de color: *C1* (azul), *C2* (blanco), *C3* (verde), *C4* (marrón claro), *C5* (marrón oscuro) y *C6* (gris). La clasificación jerárquica dividida en iteraciones del algoritmo *ACA paralelepípedos* es similar.

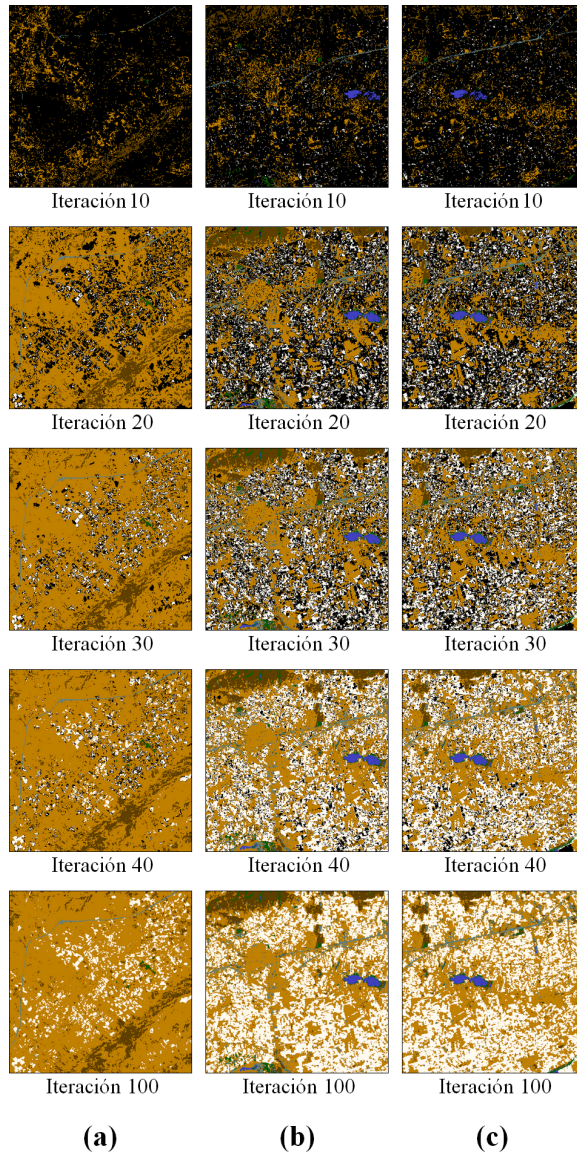


Figura 4.5: Iteraciones 10, 20, 30, 40 y 100 de la clasificación jerárquica del algoritmo *ACA mínima distancia*: (a) *Níjar*. (b) *El Ejido Oeste*. (c) *El Ejido Este*.

#### 4.1.4. Objetivo #3: detectar bordes y píxeles inciertos/ruidosos

El algoritmo ACA no sólo asigna la clase correspondiente (estado #1) y el número de iteración del autómata celular (estado #2) a cada píxel clasificado de la imagen, sino también el **tipo de píxel**: *incierto*, *ruidoso*, *foco* o *borde* (estado #3). Este tercer estado proporciona a los analistas expertos más información a partir del proceso de clasificación, información extra que permite obtener conclusiones específicas para cada una de las clases.

El análisis de los **píxeles ruidosos** es importante, ya que contribuyen a la mejora de la tasa de acierto del proceso de clasificación. El ruido introducido en las imágenes de *El Ejido Oeste* (1%) y *El Ejido Este* (2%) es de tipo *impulsivo-gaussiano* y, por lo tanto, los píxeles ruidosos se distribuyen de manera aleatoria a través de dichas imágenes de satélite. De este modo, el número de píxeles ruidosos de cada clase depende directamente del tamaño de las mismas y, en consecuencia, las clases que tienen un porcentaje más elevado de píxeles totales en la imagen, también tienen un mayor porcentaje de píxeles ruidosos. Por otro lado, el ruido introducido en cada clase en la imagen de satélite de *El Ejido Este* no siempre es el doble que el ruido introducido en cada clase en la imagen de *El Ejido Oeste*, ya que el número de píxeles totales de las clases de ambas imágenes no es el mismo. Dentro del total de píxeles ruidosos, es muy importante destacar el porcentaje de *píxeles ruidosos bien clasificados*, ya que aumentan la tasa de acierto del proceso de clasificación. La *Tabla 4.6* muestra el porcentaje de píxeles ruidosos bien clasificados de las dos imágenes de satélite de *El Ejido*, utilizando los algoritmos *ACA paralelepípedos* (*ACA P*) y *ACA mínima distancia* (*ACA MD*), divididos por clases.

Clase	<i>El Ejido Oeste</i>		<i>El Ejido Este</i>	
	<i>ACA P</i>	<i>ACA MD</i>	<i>ACA P</i>	<i>ACA MD</i>
#				
<i>C1</i>	0,007 %	0,007 %	0,013 %	0,013 %
<b><i>C2</i></b>	<b>0,454 %</b>	<b>0,418 %</b>	<b>1,036 %</b>	<b>0,961 %</b>
<i>C3</i>	0,008 %	0,008 %	0,006 %	0,006 %
<b><i>C4</i></b>	<b>0,310 %</b>	<b>0,314 %</b>	<b>0,535 %</b>	<b>0,584 %</b>
<i>C5</i>	0,048 %	0,052 %	0,036 %	0,039 %
<i>C6</i>	0,004 %	0,004 %	0,009 %	0,009 %
<b>Total</b>	<b>0,831 %</b>	<b>0,803 %</b>	<b>1,635 %</b>	<b>1,612 %</b>

Tabla 4.6: Píxeles ruidosos bien clasificados de *El Ejido Oeste* y *Este*.

Las clases con mayor porcentaje de píxeles ruidosos bien clasificados son *C2* y *C4*, precisamente porque son las clases que tienen un mayor número total de píxeles en las imágenes de satélite. En *El Ejido Oeste*, el algoritmo *ACA paralelepípedos* clasifica correctamente un 0,831 % y el algoritmo *ACA mínima distancia* un 0,803 % del total de los píxeles ruidosos, es decir, del 1 %. En *El Ejido Este*, el algoritmo *ACA paralelepípedos* clasifica correctamente un 1,635 % y el algoritmo *ACA mínima distancia* un 1,612 % del total de los píxeles ruidosos, es decir, del 2 %. En ambas imágenes de satélite, por tanto, el porcentaje de píxeles ruidosos bien clasificados es bastante elevado, y por lo tanto mejoran la tasa de acierto del proceso de clasificación.

El análisis de los **píxeles inciertos** permite a los analistas expertos especificar la contribución que ha realizado, cada clase concreta, a la mejora de la tasa de acierto del proceso de clasificación, ya que dicha mejora se basa principalmente en el refinamiento de la clasificación de este tipo de píxeles. Los píxeles inciertos, al igual que los píxeles ruidosos, pueden estar bien o mal clasificados. Los píxeles inciertos mal clasificados de cada clase corresponde a la suma de todos los valores que se encuentran fuera de la diagonal principal de la fila correspondiente de la matriz de confusión, y dicho valor se resta del número total de píxeles inciertos de dicha clase para encontrar el número de *píxeles inciertos bien clasificados*. Estos píxeles muestran el porcentaje de mejora del proceso de clasificación del algoritmo ACA sobre los algoritmos de clasificación clásicos. Las *Tablas 4.7, 4.8 y 4.9* muestran los píxeles inciertos bien clasificados en las imágenes de satélite de *Níjar, El Ejido Oeste* y *El Ejido Este*, obtenidos por parte de los algoritmos *ACA paralelepípedos* y *ACA mínima distancia*, divididos por clases.

Clase	<i>Níjar</i>	
	#	
<i>C1</i>	0,000 %	0,000 %
<i>C2</i>	<b>31,034 %</b>	<b>17,612 %</b>
<i>C3</i>	0,397 %	0,373 %
<i>C4</i>	<b>68,043 %</b>	<b>23,474 %</b>
<i>C5</i>	0,046 %	0,224 %
<i>C6</i>	0,002 %	0,002 %

Tabla 4.7: Píxeles inciertos bien clasificados de *Níjar*.

Clase	<i>El Ejido Oeste</i>	
	#	
<i>C1</i>	17,903 %	2,848 %
<i>C2</i>	<b>44,191 %</b>	<b>23,773 %</b>
<i>C3</i>	0,510 %	0,400 %
<i>C4</i>	<b>31,802 %</b>	<b>11,913 %</b>
<i>C5</i>	0,071 %	0,309 %
<i>C6</i>	0,000 %	0,000 %

Tabla 4.8: Píxeles inciertos bien clasificados de *El Ejido Oeste*.

Clase	<i>El Ejido Este</i>	
	#	
<i>C1</i>	17,841 %	2,794 %
<i>C2</i>	<b>43,561 %</b>	<b>22,393 %</b>
<i>C3</i>	0,488 %	0,376 %
<i>C4</i>	<b>31,551 %</b>	<b>11,676 %</b>
<i>C5</i>	0,055 %	0,249 %
<i>C6</i>	0,001 %	0,001 %

Tabla 4.9: Píxeles inciertos bien clasificados de *El Ejido Este*.



Las clases  $C2$  y  $C4$  son de nuevo las que tienen un mayor porcentaje de píxeles inciertos bien clasificados, pero a diferencia de lo que ocurre con los píxeles ruidosos, esto no sucede porque haya más píxeles de estas clases en las imágenes de satélite, sino porque estas dos clases son las más heterogéneas. En las clases con un grado de heterogeneidad bajo y medio, los píxeles inciertos suelen aparecer en los bordes espaciales de las clases, y por lo tanto los píxeles vecinos suelen ser de distintas clases y, en consecuencia, el algoritmo ACA no ofrece resultados muy satisfactorios. Sin embargo, en las clases con mayor grado de heterogeneidad, los píxeles inciertos pueden aparecer en lugares de la imagen en los que están rodeados por otros píxeles de la misma clase y, por lo tanto, el algoritmo ACA los clasifica correctamente. Las clases  $C2$  y  $C4$  coinciden precisamente con las clases que habíamos establecido previamente como altamente heterogéneas y que necesariamente requieren un mayor número de iteraciones para clasificar todos sus píxeles. De hecho, los resultados del *objetivo #2* han mostrado que las últimas 60 iteraciones del algoritmo ACA se utilizaron sólo para mejorar el acierto de estas dos clases, cuyos píxeles están más lejos de los centros espectrales de sus clases. Las otras clases tienen un porcentaje muy bajo de píxeles inciertos bien clasificados (menos del 1%), excepto la clase  $C1$  en las dos imágenes de *El Ejido*, así que el algoritmo ACA apenas ha mejorado la tasa de acierto de la clasificación en estas clases. Por lo tanto, el algoritmo ACA ha mejorado la tasa de acierto principalmente gracias al refinamiento de la clasificación de los píxeles de las clases  $C2$  y  $C4$ , altamente heterogéneas, donde los algoritmos clásicos suelen fallar. Aunque algunos porcentajes en la imagen de satélite de *Níjar* son menores que en las otras imágenes, la tasa de acierto obtenida en dicha imagen es mayor, ya que dicha imagen tiene un menor número total de píxeles inciertos.

El análisis de los **píxeles borde y foco** permite a los analistas expertos establecer la distribución espacial de las clases a través de las tres imágenes de satélite. Esta información es bastante importante, ya que nos permite conocer cómo están distribuidos los píxeles de las clases a lo largo de las distintas imágenes de satélite y, de este modo, obtener características específicas que diferencien una misma clase en distintas regiones geográficas. La clase  $C1$  tiene el mayor porcentaje de píxeles borde en *Níjar* y el menor porcentaje en las dos imágenes de *El Ejido*. Estos valores se deben a que, en *Níjar*, la clase  $C1$  sólo aparece en forma de pequeñas balsas de agua que ocupan un solo píxel, rodeado por píxeles de otras clases, mientras que en las imágenes de *El Ejido* las regiones de agua son más grandes y tienen una *distribución espacial elíptica*. La clase  $C3$ , por su lado, tiene un porcentaje elevado de píxeles borde en las tres imágenes de satélite, ya que la vegetación de las tres regiones geográficas se distribuye en muchos pequeños grupos de píxeles y, por lo tanto, la mayoría de los píxeles son frontera de la clase vegetación. Por otro lado, la clase  $C6$  tiene el mayor porcentaje de píxeles borde en la mayoría de las imágenes de satélite, lo que indica que los píxeles de esta clase tienen una *distribución espacial lineal*, que es la distribución espacial de las carreteras. Como se puede observar, a partir del porcentaje de píxeles borde de una clase se puede obtener fácilmente la distribución espacial de los píxeles de la misma. Las *Tablas 4.10, 4.11 y 4.12* muestran los píxeles borde y foco en las imágenes de satélite de *Níjar, El Ejido Oeste y El Ejido Este* obtenidos por parte de los algoritmos *ACA paralelepípedos y ACA mínima distancia*, divididos por clase.

<i>Níjar</i>				
Clase #	<i>ACA paralelepípedos</i>		<i>ACA mínima distancia</i>	
	Borde	Foco	Borde	Foco
<b>C1</b>	<b>100,0 %</b>	<b>0,00 %</b>	<b>100,0 %</b>	<b>0,00 %</b>
<i>C2</i>	79,96 %	20,04 %	79,78 %	20,22 %
<b>C3</b>	<b>93,16 %</b>	<b>6,84 %</b>	<b>93,21 %</b>	<b>6,79 %</b>
<i>C4</i>	50,92 %	49,08 %	89,97 %	10,03 %
<i>C5</i>	82,06 %	17,94 %	82,12 %	17,88 %
<b>C6</b>	<b>98,94 %</b>	<b>1,06 %</b>	<b>98,99 %</b>	<b>1,01 %</b>

Tabla 4.10: Píxeles borde y foco de *Níjar*.

<i>El Ejido Oeste</i>				
Clase #	<i>ACA paralelepípedos</i>		<i>ACA mínima distancia</i>	
	Borde	Foco	Borde	Foco
<b>C1</b>	<b>45,02 %</b>	<b>54,98 %</b>	<b>44,97 %</b>	<b>55,03 %</b>
<i>C2</i>	59,99 %	40,01 %	59,76 %	40,24 %
<b>C3</b>	<b>88,03 %</b>	<b>11,97 %</b>	<b>88,05 %</b>	<b>11,95 %</b>
<i>C4</i>	65,78 %	34,22 %	66,01 %	33,99 %
<i>C5</i>	61,00 %	39,00 %	61,01 %	38,99 %
<b>C6</b>	<b>96,01 %</b>	<b>3,99 %</b>	<b>96,04 %</b>	<b>3,96 %</b>

Tabla 4.11: Píxeles borde y foco de *El Ejido Oeste*.

<i>El Ejido Este</i>				
Clase #	<i>ACA paralelepípedos</i>		<i>ACA mínima distancia</i>	
	Borde	Foco	Borde	Foco
<b>C1</b>	<b>45,86 %</b>	<b>54,14 %</b>	<b>45,80 %</b>	<b>54,20 %</b>
<i>C2</i>	60,21 %	39,79 %	60,07 %	39,93 %
<b>C3</b>	<b>97,19 %</b>	<b>2,81 %</b>	<b>97,23 %</b>	<b>2,77 %</b>
<i>C4</i>	73,67 %	26,33 %	73,75 %	26,25 %
<i>C5</i>	63,47 %	36,53 %	63,49 %	36,51 %
<b>C6</b>	<b>96,73 %</b>	<b>3,27 %</b>	<b>96,75 %</b>	<b>3,25 %</b>

Tabla 4.12: Píxeles borde y foco de *El Ejido Este*.

### 4.1.5. Conclusiones

Como conclusión, se puede afirmar que los resultados obtenidos por parte del algoritmo ACA son muy satisfactorios desde varios puntos de vista, ya que se han conseguido alcanzar de manera exitosa los tres objetivos establecidos de antemano.

En primer lugar, respecto al **objetivo #1**, la *tasa de acierto* obtenida por parte del algoritmo ACA en imágenes de satélite con complejidad de clasificación baja, media y alta es bastante elevada. En la *comparativa* realizada con los otros algoritmos de clasificación (*C4.5*, *multilayer perceptron*, *naive Bayes*, *k-NN* y *RBF network*), la tasa de acierto del algoritmo ACA es mayor que la obtenida por la mayoría de clasificadores en imágenes con complejidad de clasificación baja y media, y la más elevada en imágenes con complejidad alta. Es decir, conforme aumenta el grado de complejidad de clasificación de las imágenes de satélite, aumenta también el porcentaje de mejora en la tasa de acierto del algoritmo ACA respecto al resto de clasificadores. El algoritmo ACA *mejora el proceso de clasificación* precisamente en las imágenes de satélite que presentan grandes dificultades y clases muy heterogéneas, con muchos píxeles inciertos y ruidosos. Por lo tanto, aunque el algoritmo ACA se puede usar para clasificar cualquier tipo de imagen de satélite, su aplicación está especialmente recomendada para las más complejas de clasificar. El algoritmo ACA alcanza este objetivo gracias al uso de la información contextual proporcionada por la vecindad del autómata celular, permitiendo realizar una *clasificación espectral-contextual mixta*. Este proceso es mejorado gracias a la clasificación jerárquica, ya que el algoritmo ACA utiliza como vecindad de los píxeles inciertos y ruidosos otros píxeles clasificados en iteraciones previas y, por lo tanto, píxeles más fiables por ser más cercanos a sus clases desde un punto de vista espectral. Este proceso permite clasificar los píxeles inciertos y ruidosos con un grado de fiabilidad mucho mayor.

El algoritmo ACA no sólo mejora la tasa de acierto de los algoritmos *paralelepípedos* y *mínima distancia* clásicos, sino también su *funcionamiento*, resolviendo todas las limitaciones descritas en el *Capítulo 3*. Por un lado, en los resultados del algoritmo ACA *paralelepípedos* no existen píxeles sin clasificar, ya que el *threshold* aumenta en cada iteración del autómata celular hasta que todos los píxeles de la imagen han sido clasificados, y además no existen píxeles clasificados en varias clases, ya que los píxeles inciertos son refinados con información contextual. Por otro lado, en los resultados del algoritmo ACA *mínima distancia* se reducen bastante los errores de comisión utilizando información contextual, y aunque la varianza no se tiene en cuenta en el proceso de clasificación del algoritmo ACA, realmente es simulada para los píxeles inciertos y ruidosos mediante las funciones de vecindad del autómata celular.

Como consecuencia de los buenos resultados obtenidos, el algoritmo ACA optimiza la funcionalidad general de cualquier *SIG* que lo utilice, y cuyo funcionamiento se base en la identificación adecuada de los píxeles de las imágenes de satélite en sus clases correspondientes, debido a que mejora la tasa de acierto de la clasificación de las imágenes de satélite. Por lo tanto, respecto a la mejora de la tasa de acierto durante el proceso de clasificación, los resultados obtenidos por parte del algoritmo ACA con imágenes de satélite con complejidad baja y media son muy buenos, y los obtenidos con imágenes de satélite con complejidad alta son excelentes.

En segundo lugar, respecto al **objetivo #2**, el algoritmo ACA obtiene una *clasificación jerárquica* dividida en niveles o capas de fiabilidad basadas en el grado de pertenencia de los píxeles a las clases correspondientes, donde cada nivel corresponde a una iteración del autómata celular. A lo largo de la ejecución del algoritmo ACA, el *threshold* va aumentando progresivamente en cada iteración. De este modo, las primeras iteraciones proporcionan las capas más fiables, ya que están formadas por los píxeles que se encuentran más cercanos a los centros de sus clases desde un punto de vista espectral, y muchos de estos píxeles incluso pertenecen al conjunto de entrenamiento elegido por los ecólogos expertos. Conforme vamos avanzando en el número de iteraciones del autómata celular, el *threshold* aumenta y, en consecuencia, la distancia espectral de los píxeles respecto a los centros de sus clases también lo hace y, como resultado, los píxeles clasificados son menos fiables en términos de tasa de acierto. Finalmente, los píxeles inciertos y ruidosos son clasificados en las últimas iteraciones con ayuda de la *información contextual* de sus píxeles vecinos, la mayoría de ellos ya clasificados en iteraciones anteriores, y por lo tanto con un mayor grado de fiabilidad, lo que ayuda a mejorar la tasa de acierto final. Utilizando este método, el algoritmo ACA simula una *clasificación pseudodifusa* en la que el *grado de pertenencia* de cada píxel a su clase correspondiente está indicado por el número de iteración en que ha sido clasificado, de tal modo que dicho grado de pertenencia es muy alto en las primeras iteraciones, más fiables, pero es considerablemente bajo en las últimas iteraciones, donde se clasifican los píxeles más problemáticos. La presentación de los resultados del proceso de clasificación del algoritmo ACA divididos en niveles jerárquicos proporciona al analista experto información extra de cada píxel, en este caso, el grado de pertenencia a su clase correspondiente y en consecuencia su fiabilidad de clasificación. También ofrece información extra de las clases, permitiendo determinar el *grado de heterogeneidad* de las mismas, ya que en las primeras 40 iteraciones se clasifican todos los píxeles de las clases con menor grado de heterogeneidad, y los píxeles inciertos que pertenecen a las clases altamente heterogéneas son clasificados en las siguientes 60 iteraciones. Por lo tanto, cuantas más iteraciones use una clase para clasificar todos sus píxeles, mayor será su grado de heterogeneidad.

En tercer lugar, respecto al **objetivo #3**, el algoritmo ACA también ofrece la detección de los bordes espaciales de las clases, información extra que puede ser útil para el posterior análisis e interpretación de los resultados, así como una lista de píxeles inciertos y ruidosos, para que los analistas expertos siempre puedan identificarlos con facilidad. El análisis de los *píxeles ruidosos e inciertos* muestran la contribución de cada clase concreta a la *mejora en la tasa de acierto*, mientras que el análisis de los *píxeles borde y foco* determinan la *distribución espacial de los píxeles* de cada clase.

## 4.2. RESULTADOS COMPLEMENTARIOS DE ACA

En este apartado nos centramos en la obtención de la información complementaria que ofrece el algoritmo ACA para cada iteración del proceso de clasificación. Se trata de una ampliación de los resultados del apartado anterior, concretamente de la imagen de satélite de *El Ejido Oeste*. Posteriormente se analizan los resultados obtenidos por parte de dicha información complementaria, y se ofrecen unas conclusiones.

### 4.2.1. Características experimentales

La obtención de la información complementaria por iteraciones se ha llevado a cabo sobre la imagen de satélite *Landsat TM* de *El Ejido Oeste* del apartado anterior. Se trata de una imagen de satélite multispectral de 7 bandas, con una resolución espacial de  $30 \times 30$  m y de tamaño  $400 \times 400$  píxeles, es decir, un total de 160000 píxeles. La *Figura 4.6* muestra la imagen de satélite de *El Ejido Oeste* (bandas 3, 2, 1).

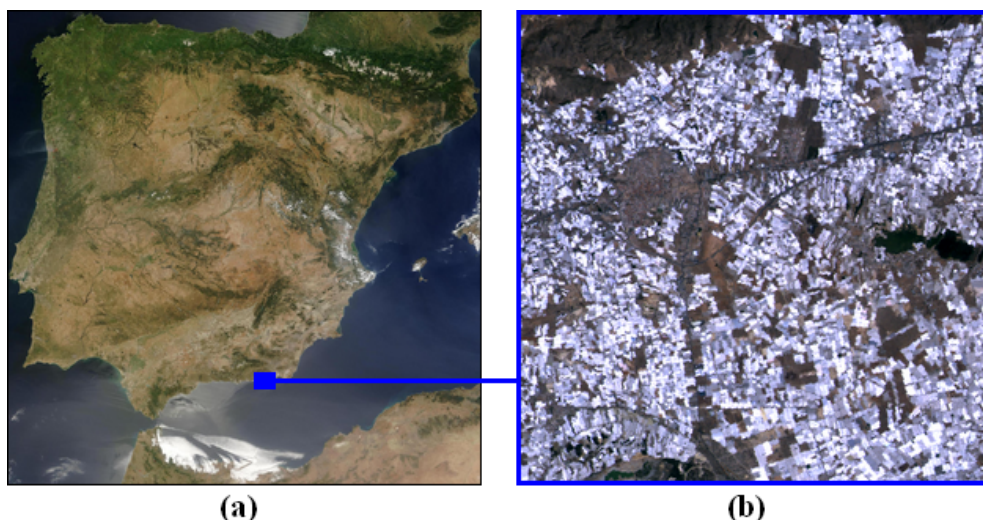


Figura 4.6: (a) Área de estudio de Almería, España (abril de 2003). (b) *El Ejido Oeste*.

La *Tabla 4.13* muestra las características de las seis clases que se pueden encontrar en la imagen de satélite de *El Ejido Oeste*, donde se puede observar el sector al que corresponde cada clase, la variable ecológica de vegetación con la que se identifica y el grado de heterogeneidad de la misma.

Clase	Sector	Descripción: variable ecológica	Heterogeneidad
<i>C1</i>	3	Zonas húmedas y superficies de agua	Media
<i>C2</i>	3	<b>Invernaderos</b>	<b>Muy alta</b>
<i>C3</i>	3	Pastizal continuo	Media
<i>C4</i>	3	<b>Superficies construidas y alteradas</b>	<b>Alta</b>
<i>C5</i>	1	Matorral disperso con pasto y roca o suelo	Media
<i>C6</i>	3	Carretera	Baja

Tabla 4.13: Características principales de las 6 clases de *El Ejido Oeste*.

Por otro lado, la imagen de satélite de *El Ejido Oeste* tiene un 55% de píxeles de invernadero y un 1% de ruido artificial añadido, lo que la convierte en una imagen con complejidad de clasificación media, según el criterio adoptado en el apartado anterior.

### 4.2.2. Información #1: evolución de las matrices de confusión

El algoritmo ACA no sólo mejora la tasa de acierto final de la clasificación obtenida por otros algoritmos de clasificación supervisados gracias al tratamiento especial que realiza sobre los píxeles problemáticos, sino que además ofrece una enorme cantidad de información complementaria a partir del proceso de clasificación para cada una de las iteraciones del autómata celular. En esta sección vamos a utilizar varias *matrices de confusión intermedias* correspondientes a distintas iteraciones, con el objetivo de comprender mejor el funcionamiento del algoritmo ACA, mostrando detalles concretos de los momentos más importantes del proceso de clasificación.

Gracias a dichas matrices de confusión intermedias, en esta sección se ha comparado cada algoritmo de clasificación supervisado clásico, *paralelepípedos* y *mínima distancia*, con el correspondiente algoritmo de clasificación ACA modificado, *ACA paralelepípedos* y *ACA mínima distancia*, para comprobar el porcentaje de mejora en la tasa de acierto. Con el objetivo de comparar los algoritmos clásicos y los algoritmos ACA, se han utilizado las dos matrices de confusión finales obtenidas entre ambas imágenes clasificadas y la imagen de trabajo de campo realizado por los ecólogos expertos en anteriores experimentos del proyecto *SOLERES*, y posteriormente se ha calculado el número de píxeles correctamente clasificados en cada algoritmo, mediante la suma de los valores presentes en la diagonal principal de la matriz de confusión correspondiente. Este proceso también se ha realizado para las matrices de confusión intermedias de los algoritmos *ACA paralelepípedos* y *ACA mínima distancia*.

El algoritmo *paralelepípedos clásico* clasifica correctamente 120032 píxeles y, por lo tanto, alcanza un **75,02%** de acierto. Por otro lado, si contamos el número total de píxeles de la matriz de confusión, verificamos que no hay 160000 píxeles, sino que sólo hay 148130 píxeles, es decir, faltan 11870 píxeles, un 7,42% del total de píxeles de la imagen. De este porcentaje, un 1% corresponde a los píxeles ruidosos, y el resto corresponde a los píxeles que no han sido clasificados debido a las limitaciones del algoritmo clásico comentadas en el *Capítulo 3*. La *Tabla 4.14* muestra la matriz de confusión del algoritmo *paralelepípedos* clásico.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>1072</b>	0	15	2	2	18
<i>C2</i>	35	<b>62835</b>	27	15098	35	514
<i>C3</i>	54	0	<b>1245</b>	130	819	181
<i>C4</i>	1	498	22	<b>46474</b>	3057	4567
<i>C5</i>	52	0	579	2290	<b>7756</b>	74
<i>C6</i>	1	0	3	24	0	<b>650</b>

Tabla 4.14: Matriz de confusión del algoritmo *paralelepípedos* clásico.

El algoritmo *ACA paralelepípedos* mejora los resultados del algoritmo *paralelepípedos* clásico. Sin embargo, antes de considerar la matriz de confusión final del algoritmo *ACA paralelepípedos*, correspondiente a la última iteración del proceso de clasificación, vamos a explorar otras matrices de confusión intermedias, para entender mejor el funcionamiento de este algoritmo.

Por ejemplo, podemos observar que en las primeras iteraciones del autómata celular sólo nos encontramos con píxeles bien clasificados en la matriz de confusión, ya que el *threshold* es muy pequeño, de modo que la mayor parte de los píxeles clasificados en estas primeras iteraciones corresponden al conjunto de entrenamiento elegido por los ecólogos expertos. Debido al comportamiento basado en iteraciones, el algoritmo proporciona unos resultados de clasificación jerárquica divididos en diferentes niveles de fiabilidad, donde los mejores píxeles son clasificados en las primeras iteraciones y los peores (inciertos y ruidosos) son clasificados en las últimas. La primera iteración en la que nos encontramos con píxeles clasificados erróneamente en la matriz de confusión es la número 2, en la que hay 137 píxeles correctamente clasificados y 2 mal clasificados (0,09% de acierto), tal como se muestra en la *Tabla 4.15*.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>9</b>	0	0	0	0	0
<i>C2</i>	0	<b>3</b>	0	0	0	0
<i>C3</i>	0	0	<b>0</b>	0	0	0
<i>C4</i>	0	0	0	<b>27</b>	0	<b>2</b>
<i>C5</i>	0	0	0	0	<b>78</b>	0
<i>C6</i>	0	0	0	0	0	<b>20</b>

Tabla 4.15: Matriz de confusión del algoritmo *ACA paralelepípedos* (iteración 2).

Si el algoritmo continúa su ejecución, podemos observar que existe un punto en el que ya han sido clasificados todos los píxeles de las clases con un grado de heterogeneidad bajo y medio (*C1*, *C3*, *C5* y *C6*), y desde ese punto sólo permanecen sin clasificar algunos píxeles de las clases con un grado de heterogeneidad alto (*C2* y *C4*). Este suceso se produce en la iteración 40, en la que 153341 píxeles ya han sido clasificados. De estos píxeles, 128441 están correctamente clasificados y 24900 están mal clasificadas, es decir, el algoritmo alcanza un **80,28%** de acierto, de modo que en este punto el algoritmo *ACA paralelepípedos* ya ha mejorado los resultados del paralelepípedo clásico, como se muestra en la *Tabla 4.16*.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>1077</b>	0	13	2	2	15
<i>C2</i>	31	<b>68293</b>	25	12134	34	508
<i>C3</i>	54	0	<b>1250</b>	130	815	180
<i>C4</i>	1	390	22	<b>49413</b>	2995	4528
<i>C5</i>	52	0	579	2288	<b>7758</b>	74
<i>C6</i>	1	0	3	24	0	<b>650</b>

Tabla 4.16: Matriz de confusión del algoritmo *ACA paralelepípedos* (iteración 40).

Durante las siguientes 60 iteraciones, el algoritmo *ACA paralelepípedos* refina el proceso de clasificación, con el objetivo de seguir mejorando los resultados, clasificando correctamente los píxeles más problemáticos: los píxeles inciertos de las clases *C2* y *C4*, y los píxeles ruidosos de todas las clases.

En la iteración 100, todos y cada uno de los 160000 píxeles de la imagen de satélite han sido clasificados. De estos píxeles, 132988 están correctamente clasificados y 27012 están mal clasificados, es decir, un **83,12%** de acierto, por lo que el algoritmo *ACA paralelepípedos* ha mejorado un **8,10%** los resultados con respecto el algoritmo *paralelepípedos* clásico (aproximadamente un 1% de los píxeles ruidosos y un 7% de los píxeles inciertos), como se muestra en la *Tabla 4.17*.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>1077</b>	0	13	2	2	15
<i>C2</i>	31	<b>72661</b>	25	14169	34	513
<i>C3</i>	54	0	<b>1250</b>	130	815	180
<i>C4</i>	1	461	22	<b>49592</b>	2996	4528
<i>C5</i>	52	0	579	2288	<b>7758</b>	74
<i>C6</i>	1	0	3	24	0	<b>650</b>

Tabla 4.17: Matriz de confusión del algoritmo *ACA paralelepípedos* (iteración 100).

Además, el algoritmo *ACA paralelepípedos* ha resuelto las limitaciones del algoritmo *paralelepípedos* clásico, ya que, gracias al uso de los autómatas celulares, ya no existen píxeles sin clasificar al final del proceso de clasificación, gracias al incremento del *threshold* en cada iteración. Por lo tanto, los autómatas celulares no sólo mejoran la tasa de acierto del proceso de clasificación, sino que también mejoran el funcionamiento del algoritmo clásico.

El algoritmo *mínima distancia clásico* clasifica correctamente 121888 píxeles, por lo que alcanza un **76,18%** de acierto. Como se puede observar, el algoritmo mínima distancia clásico obtiene mejores resultados que el algoritmo *paralelepípedos* clásico. La *Tabla 4.18* muestra la matriz de confusión del algoritmo mínima distancia clásico.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>1074</b>	0	19	0	0	16
<i>C2</i>	32	<b>61348</b>	12	25467	28	545
<i>C3</i>	44	0	<b>1154</b>	70	780	381
<i>C4</i>	0	412	7	<b>49339</b>	2962	4881
<i>C5</i>	47	0	496	1804	<b>8318</b>	86
<i>C6</i>	0	0	2	21	0	<b>655</b>

Tabla 4.18: Matriz de confusión del algoritmo mínima distancia clásico.

Al igual que ocurre con el algoritmo anterior, vamos a explorar otras matrices de confusión intermedias antes de considerar la matriz de confusión final del algoritmo *ACA mínima distancia*. Tal como sucede en el algoritmo anterior, podemos observar que en las primeras iteraciones del autómata celular sólo existen píxeles bien clasificados en la matriz de confusión. La primera iteración en la que aparecen píxeles erróneamente clasificados en la matriz de confusión es la número 3, en la que hay 757 píxeles correctamente clasificados (0,47% de acierto) y 3 píxeles mal clasificados, como se puede observar en la *Tabla 4.19*.



	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>41</b>	0	0	0	0	0
<i>C2</i>	0	<b>11</b>	0	0	0	0
<i>C3</i>	0	0	<b>5</b>	0	0	0
<i>C4</i>	0	0	0	<b>264</b>	0	<b>3</b>
<i>C5</i>	0	0	0	0	<b>342</b>	0
<i>C6</i>	0	0	0	0	0	<b>94</b>

Tabla 4.19: Matriz de confusión del algoritmo *ACA mínima distancia* (iteración 3).

En el algoritmo *ACA mínima distancia*, el punto en el que todos los píxeles de las clases con grado de heterogeneidad bajo y medio ya han sido clasificados ocurre en la iteración 44, en la que 149720 píxeles han sido clasificados. De estos píxeles, 120877 están correctamente clasificados y 28843 están erróneamente clasificados, es decir, un **75,55%** de acierto, por lo que en este punto el algoritmo *ACA mínima distancia* está a punto de mejorar los resultados del algoritmo mínima distancia clásico. Durante las siguientes 56 iteraciones, el algoritmo *ACA mínima distancia* refina el proceso de clasificación para seguir mejorando los resultados, como se muestra en la *Tabla 4.20*.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>1075</b>	0	18	0	0	16
<i>C2</i>	28	<b>59684</b>	11	17295	28	455
<i>C3</i>	44	0	<b>1158</b>	69	778	380
<i>C4</i>	0	237	5	<b>49982</b>	2697	4331
<i>C5</i>	47	0	492	1804	<b>8323</b>	85
<i>C6</i>	0	0	2	21	0	<b>655</b>

Tabla 4.20: Matriz de confusión del algoritmo *ACA mínima distancia* (iteración 44).

En la iteración 100, cada uno de los 160000 píxeles de la imagen de satélite han sido clasificados. De estos píxeles, 128162 están correctamente clasificados y 31838 están mal clasificados, es decir, un **80,10%** de precisión, así que el algoritmo *ACA mínima distancia* ha mejorado un **3,92%** los resultados del algoritmo mínima distancia clásico (aproximadamente un 1% de píxeles ruidosos y un 3% de píxeles inciertos), como se muestra en la *Tabla 4.21*.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>
<i>C1</i>	<b>1075</b>	0	18	0	0	16
<i>C2</i>	28	<b>66763</b>	11	20137	28	465
<i>C3</i>	44	0	<b>1158</b>	69	778	380
<i>C4</i>	0	376	5	<b>50188</b>	2697	4335
<i>C5</i>	47	0	492	1804	<b>8323</b>	85
<i>C6</i>	0	0	2	21	0	<b>655</b>

Tabla 4.21: Matriz de confusión del algoritmo *ACA mínima distancia* (iteración 100).

En resumen, utilizando la imagen de satélite de *El Ejido Oeste*, el algoritmo *ACA paralelepípedos* mejora la tasa de acierto del algoritmo *paralelepípedos* clásico un 8,10 % y el algoritmo *ACA mínima distancia* mejora la tasa de acierto del algoritmo *mínima distancia* clásico un 3,92 %. Además, en ambos algoritmos *ACA* se han agrupado los procesos de pre-clasificación (reducción de ruido), clasificación y post-clasificación (refinamiento de píxeles inciertos). La *Figura 4.7* muestra la evolución de la tasa de acierto de los algoritmos *ACA paralelepípedos* y *ACA mínima distancia* a lo largo de las 100 iteraciones del proceso de clasificación.

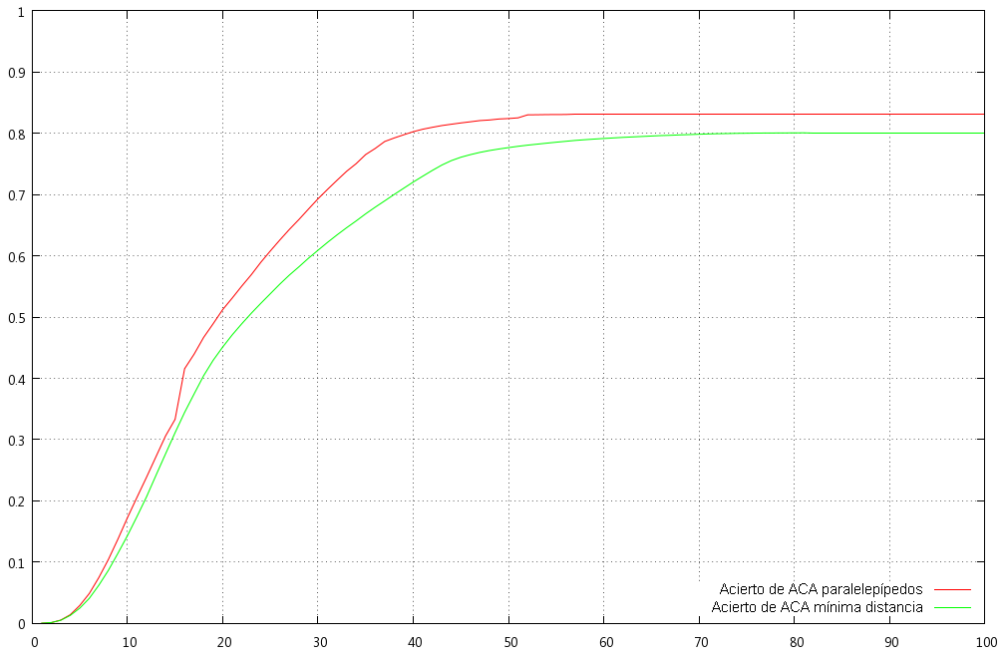


Figura 4.7: Evolución de la tasa de acierto de los algoritmos *ACA paralelepípedos* y *ACA mínima distancia* a lo largo de las 100 iteraciones de la clasificación.

### 4.2.3. Información #2: evolución de la precisión y el *recall*

Esta sección se centra en la información complementaria que ofrece el algoritmo *ACA* como consecuencia de la clasificación jerárquica. En primer lugar, se muestra una clasificación jerárquica más detallada de la imagen de satélite de *El Ejido Oeste*, y posteriormente se muestra la evolución de la precisión y el *recall* de las clases a lo largo de las iteraciones del autómata celular. Las *Figuras 4.8*, *4.9* y *4.10* muestran el proceso de clasificación de la imagen de satélite de *El Ejido Oeste* durante las 100 iteraciones del algoritmo *ACA mínima distancia* con la siguiente configuración de colores: *C1* (azul), *C2* (blanco), *C3* (verde), *C4* (marrón claro), *C5* (marrón oscuro) y *C6* (gris).

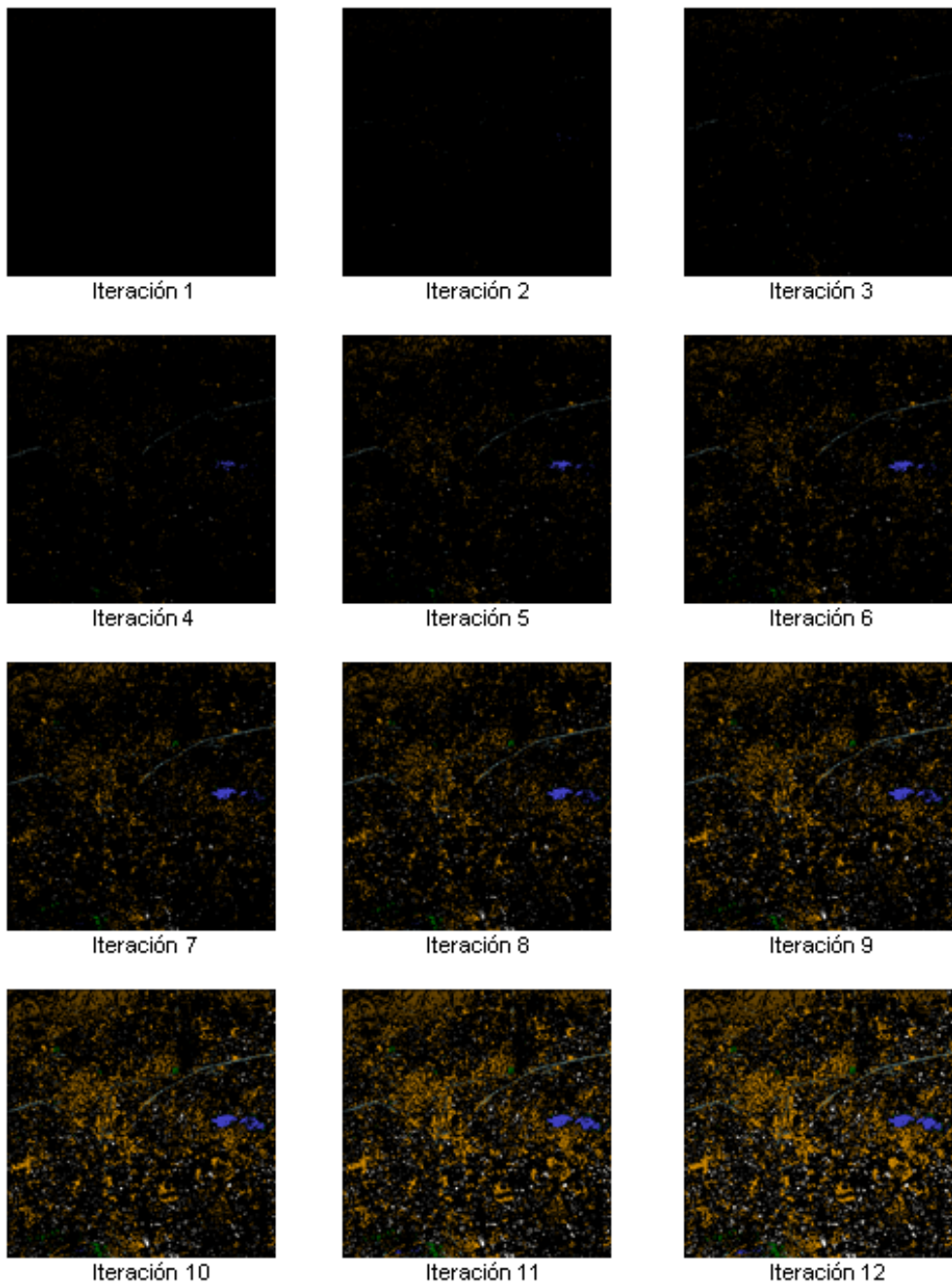


Figura 4.8: Iteraciones 1-12 de la clasificación *ACA mínima distancia*, *El Ejido Oeste*.

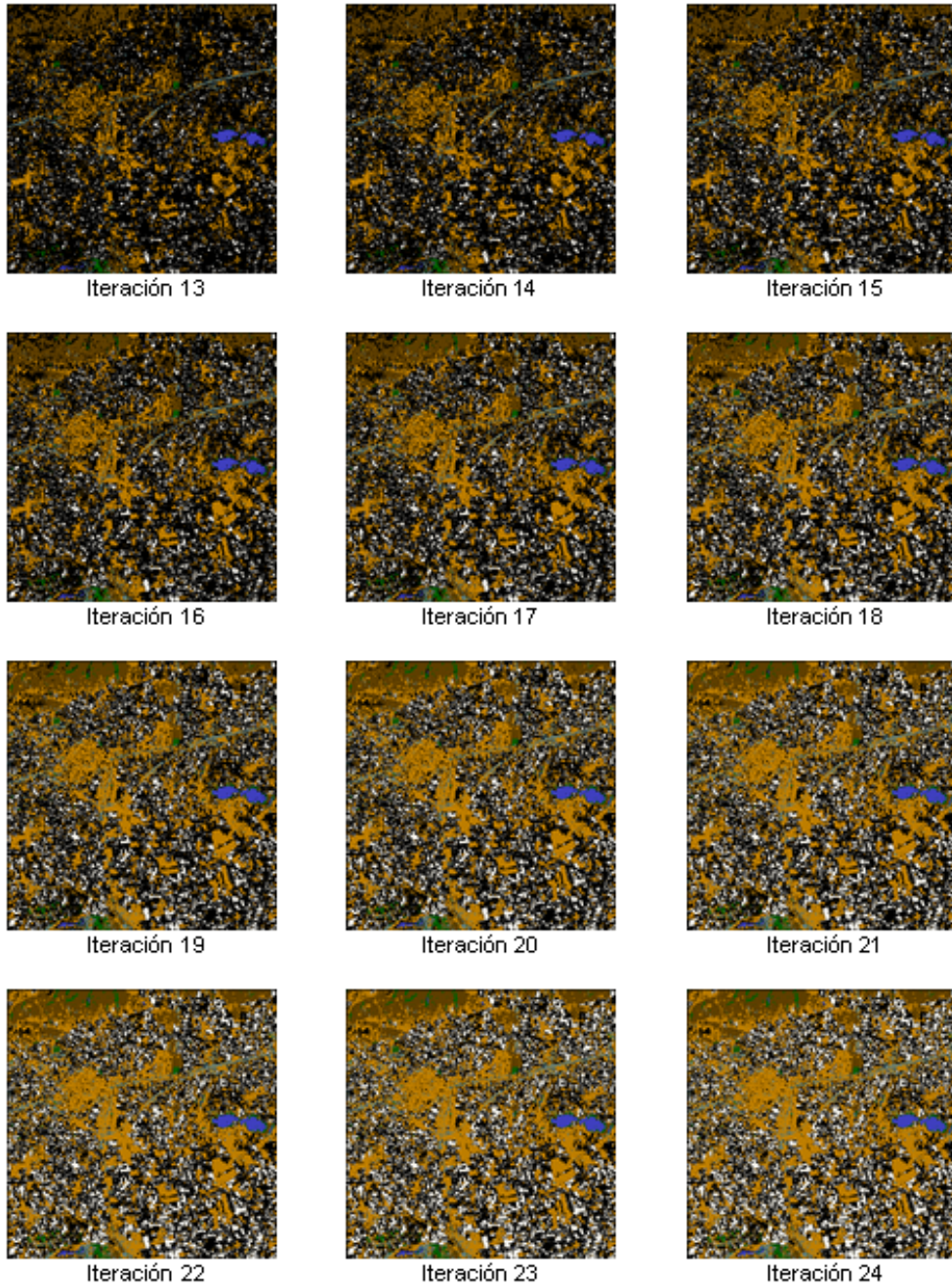


Figura 4.9: Iteraciones 13-24 de la clasificación *ACA mínima distancia*, *El Ejido Oeste*.



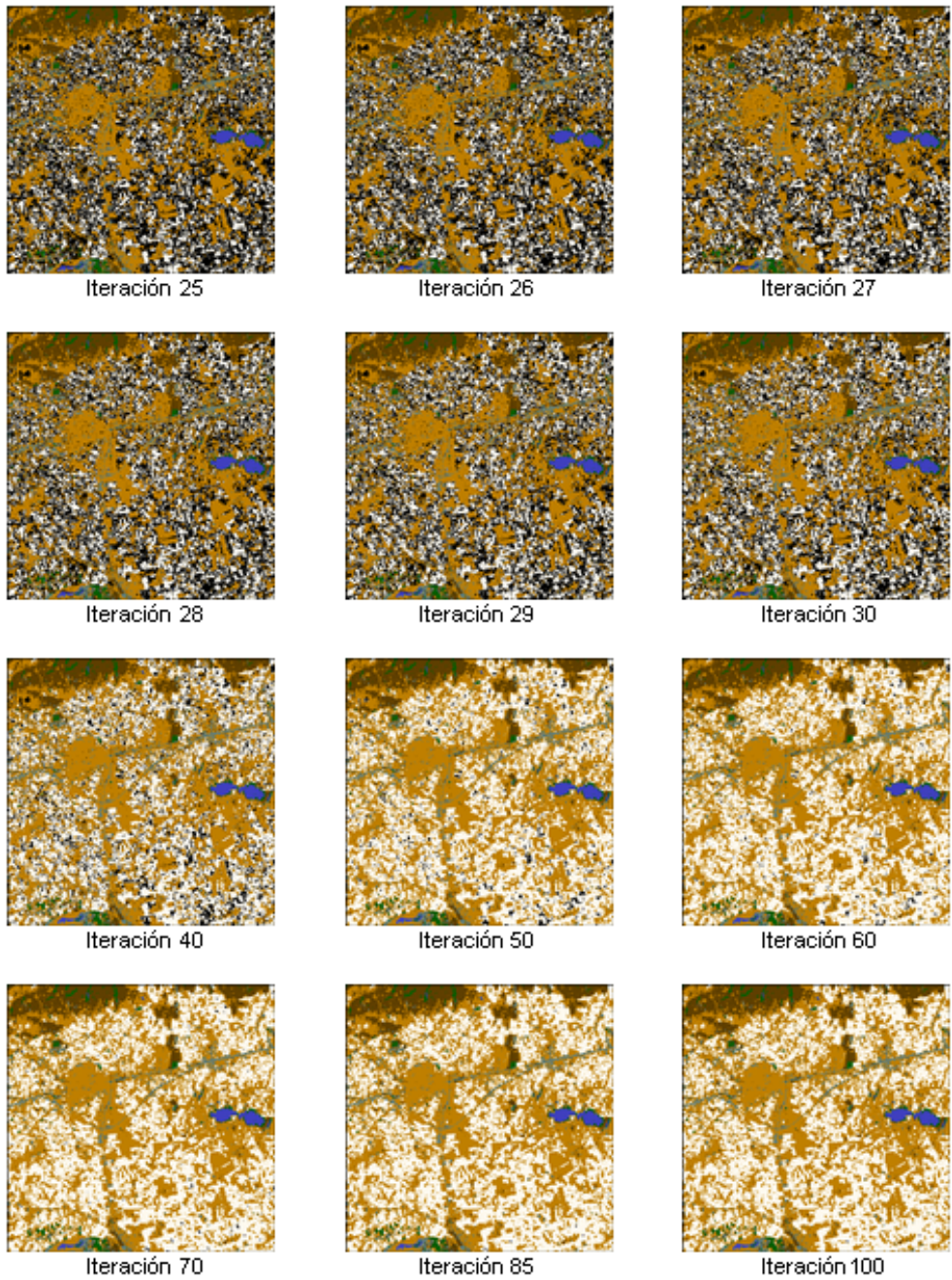


Figura 4.10: Iteraciones 25-100 de la clasificación *ACA mínima distancia*, *El Ejido Oeste*.

Con el fin de analizar la evolución de las diferentes clases a lo largo del proceso de clasificación, el algoritmo ACA calcula la **precisión** y el **recall** de cada clase en todas las iteraciones. En las primeras iteraciones, los valores de precisión y *recall* son muy altos porque existen pocos píxeles mal clasificados en las matrices de confusión. Sin embargo estos valores van disminuyendo progresivamente en las siguientes iteraciones, conforme van apareciendo píxeles mal clasificados en las matrices de confusión.

Con toda esta información podemos ampliar la comparativa realizada en el apartado anterior con los algoritmos de clasificación *C4.5*, *multilayer perceptron*, *naive Bayes*, *k-NN* y *RBF network*, a través de la información de precisión y *recall*. Para ello, tomamos los valores de precisión y *recall* de la última iteración de los algoritmos *ACA paralelepípedos* y *ACA mínima distancia*, y los valores del resto de algoritmos. Las Figuras 4.11 y 4.12 muestran la comparativa de la precisión y el *recall* de todos los algoritmos usados en el apartado anterior, para la imagen de satélite de *El Ejido Oeste*.

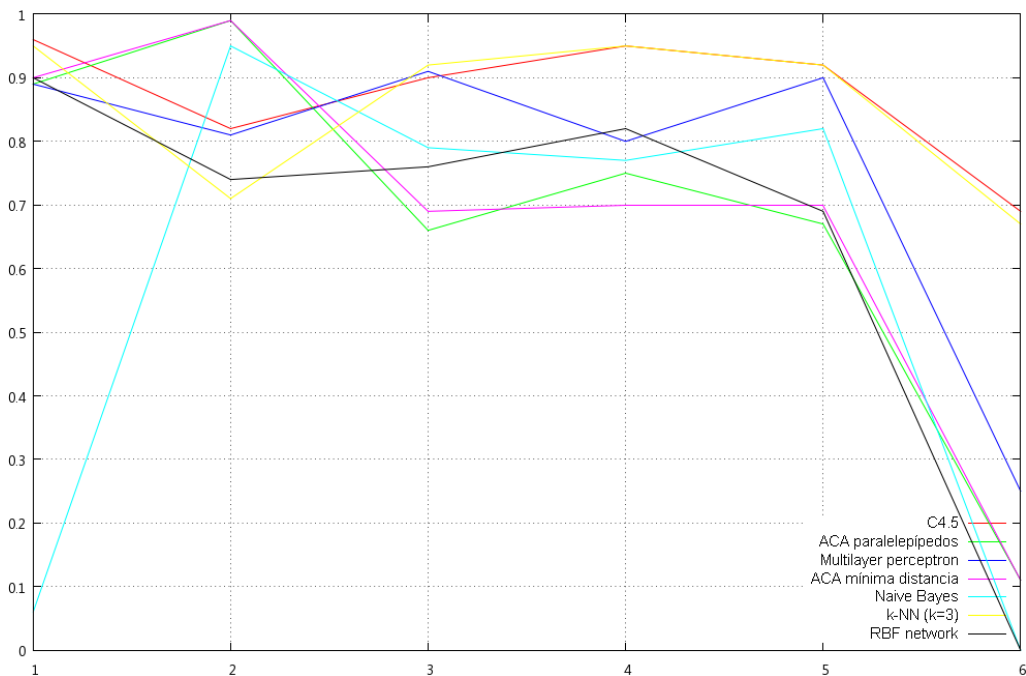


Figura 4.11: Precisión de los algoritmos de la comparativa, *El Ejido Oeste*.

La Figura 4.13 muestra la evolución de la precisión de las clases de *ACA paralelepípedos* a través de las 100 iteraciones del proceso de clasificación. Si nos centramos en la evolución de la precisión de las clases del algoritmo *ACA paralelepípedos*, podemos observar que la clase con menor precisión es *C6* y la clase con mayor precisión es *C2*. Además, podemos observar que, desde la iteración 40 a la 100, el nivel de precisión de todas las clases se mantiene estable, excepto el nivel de precisión de las clases *C2* y *C4*, las clases definidas con un grado de heterogeneidad elevado.

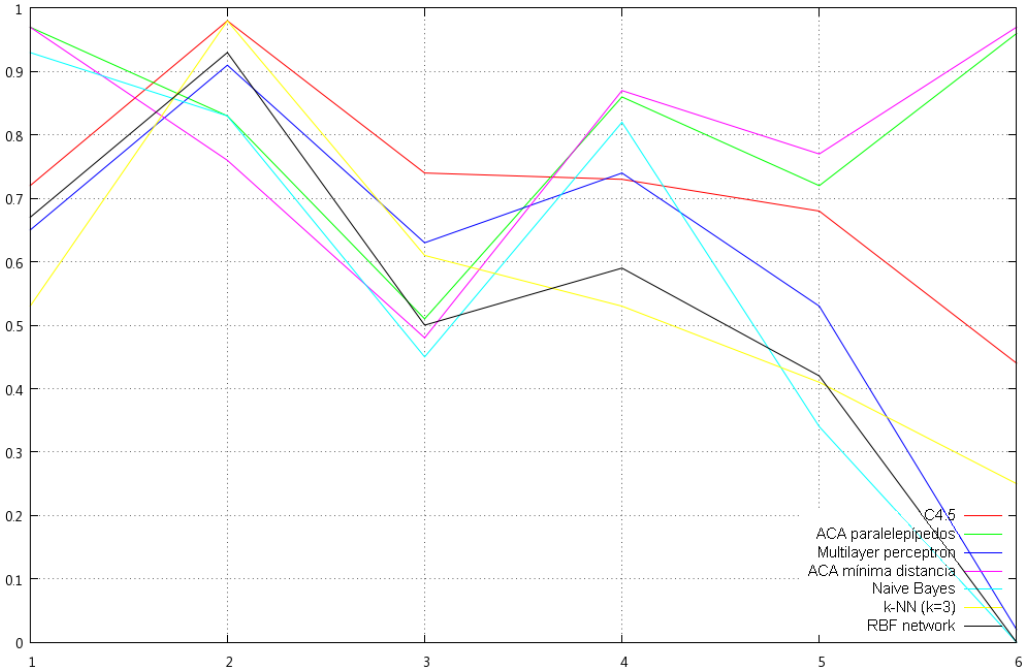


Figura 4.12: *Recall* de los algoritmos de la comparativa, *El Ejido Oeste*.

La Figura 4.14 muestra la evolución del *recall* de algoritmo *ACA paralelepípedos* a través de las 100 iteraciones del proceso de clasificación. Si observamos la evolución del *recall* de las clases del algoritmo *ACA paralelepípedos*, nos encontramos que la clase con el menor nivel de *recall* es *C3* y la clase con el mayor nivel de *recall* es *C1*. Además, se puede observar que, al igual que ocurre en la evolución de la precisión de las clases, desde la iteración 40 a la 100, el nivel de *recall* de todas las clases se mantiene estable, excepto el nivel de *recall* de las clases *C2* y *C4*.

La Figura 4.15 muestra la evolución de la precisión de las clases de *ACA mínima distancia* a través de las 100 iteraciones del proceso de clasificación. De la misma manera, si nos centramos en la evolución de la precisión de las clases del algoritmo *ACA mínima distancia*, podemos observar que las clases con el menor y el mayor nivel de precisión son las mismas que en el algoritmo *ACA paralelepípedos* (*C6* y *C2*), y desde la iteración 44 a la 100 el nivel de precisión de todas las clases permanece estable, excepto los nivel de precisión de las clases *C2* y *C4*.

La Figura 4.16 muestra la evolución del *recall* de las clases de *ACA mínima distancia* a través de las 100 iteraciones del proceso de clasificación. Si observamos la evolución del *recall* de las clases del algoritmo *ACA mínima distancia*, nos encontramos que la clase con el menor nivel de *recall* es *C3*, y las clases con mayor nivel de *recall* son *C1* y *C6*.

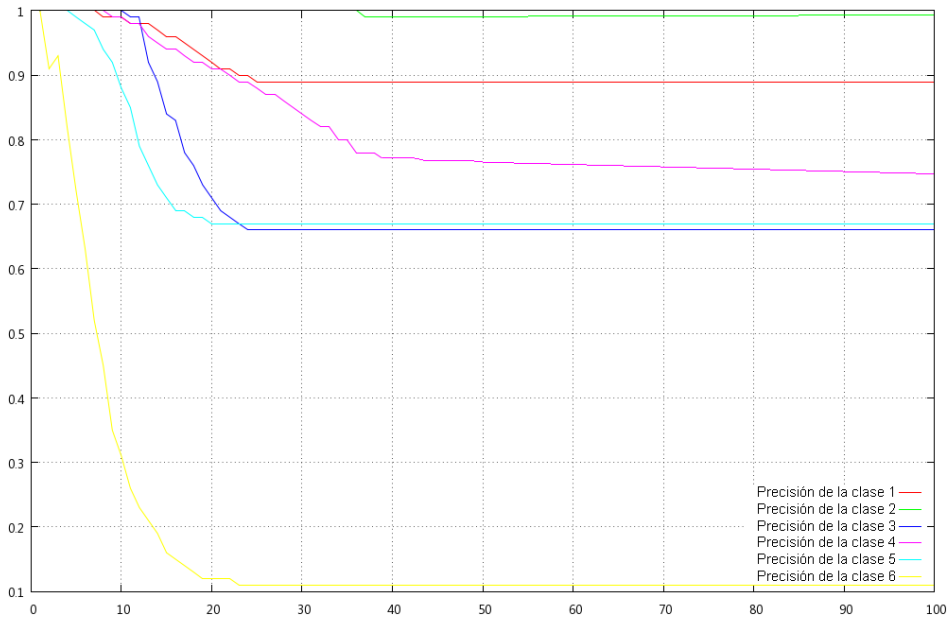


Figura 4.13: Evolución de la precisión de las clases con *ACA paralelepípedos*.

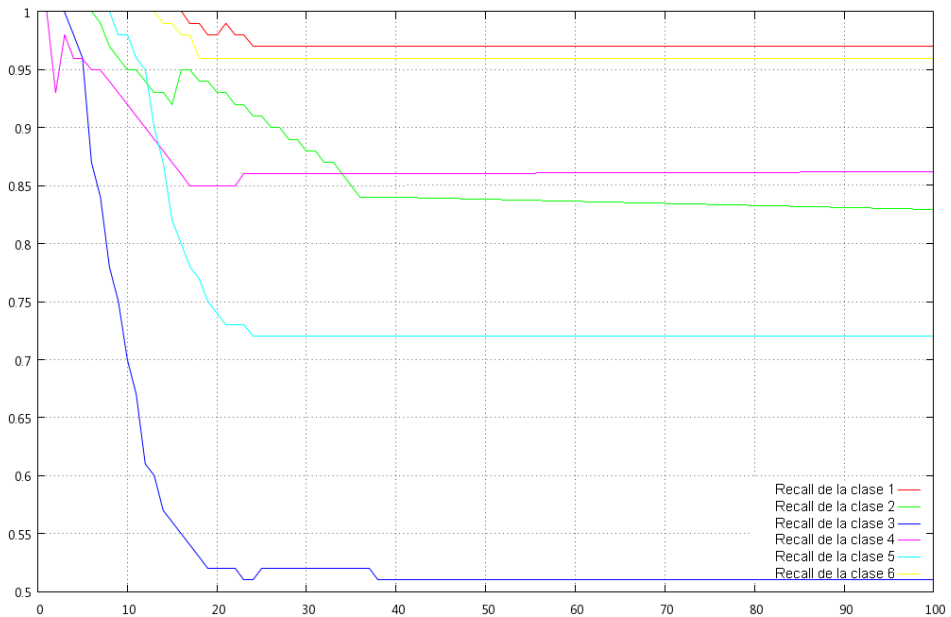


Figura 4.14: Evolución del *recall* de las clases con *ACA paralelepípedos*.



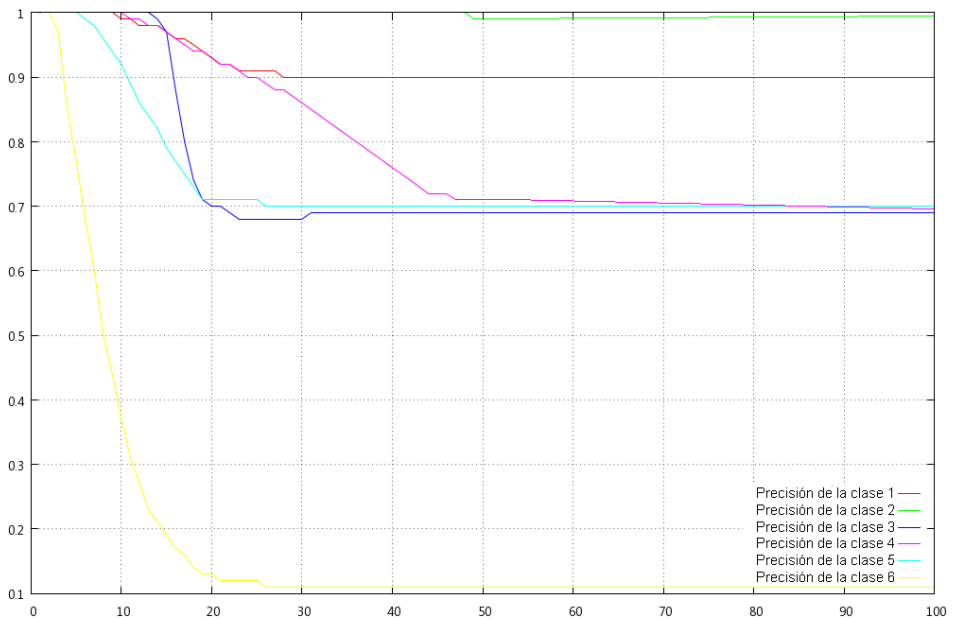


Figura 4.15: Evolución de la precisión de las clases con *ACA mínima distancia*.

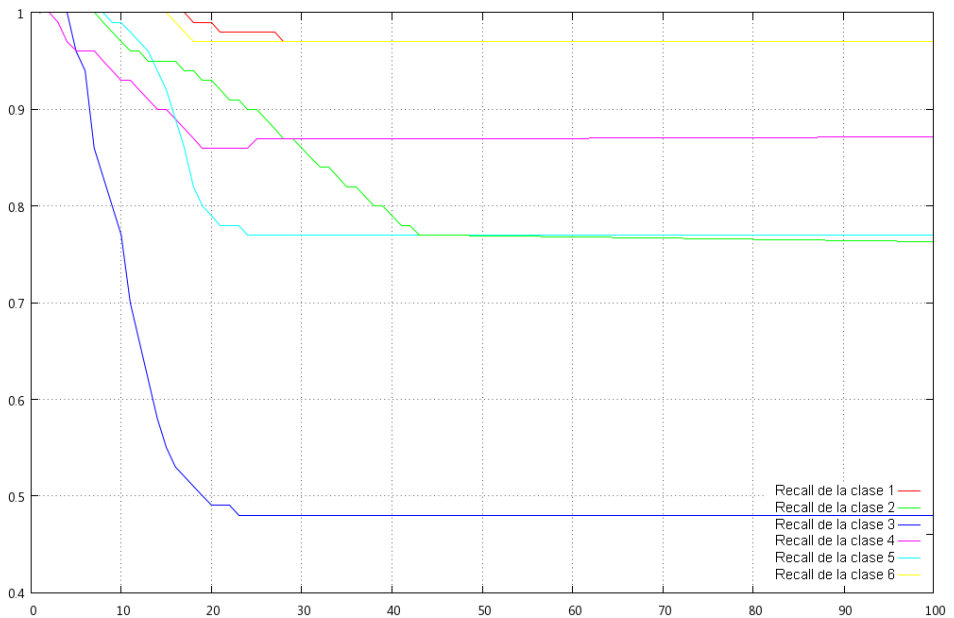


Figura 4.16: Evolución del *recall* de las clases con *ACA mínima distancia*.

#### 4.2.4. Información #3: evolución de las tablas de píxeles

El algoritmo ACA no sólo ofrece la posibilidad de observar los resultados del *objetivo #3* mediante las tablas finales con los porcentajes de los distintos tipos de píxeles, sino también mediante las *tablas intermedias por iteraciones*, que desglosan el número específico de píxeles de cada tipo. Las *Tablas 4.22* y *4.23* corresponden a la iteración 100 de *ACA paralelepípedos* y *ACA mínima distancia*, y tienen las siguientes características:

- La suma de todos los *píxeles bien clasificados* de cada clase (ruidosos, inciertos, bordes y focos) corresponde al valor de la *diagonal principal* de esa misma clase en la matriz de confusión correspondiente.
- El valor de los *píxeles erróneos* de cada clase, por otro lado, corresponde a la suma de los valores que se encuentran *fuera de la diagonal principal* de esa misma clase en la matriz de confusión correspondiente. Además, los píxeles mal clasificados pueden ser de los cuatro tipos de píxeles establecidos (ruidosos, inciertos, bordes y focos), es decir, cualquier tipo puede tener píxeles bien y mal clasificados.
- La suma de *todos los valores* de cada clase (los cuatro valores de los píxeles bien clasificados y el valor de los píxeles erróneos), corresponde al número total de píxeles de dicha clase, es decir, los que se han clasificado *correctamente* en la clase correspondiente y los que se han clasificado *erróneamente* en otra clase.

<i>El Ejido Oeste</i>						
Clase	Píxeles bien clasificados				Píxeles erróneos	
#	Ruidoso	Incierto	Borde	Foco	Ruidoso/incierto/borde/foco	Total
<i>C1</i>	11	7	477	582	32	1109
<i>C2</i>	727	11847	36052	24035	14772	87433
<i>C3</i>	12	19	1073	146	1179	2429
<i>C4</i>	496	3811	29888	15397	8008	57600
<i>C5</i>	77	32	4666	2983	2993	10751
<i>C6</i>	6	1	617	26	28	678

Tabla 4.22: Píxeles ruidosos, inciertos, bordes, focos y erróneos. *ACA paralelepípedos*.

<i>El Ejido Oeste</i>						
Clase	Píxeles bien clasificados				Píxeles erróneos	
#	Ruidoso	Incierto	Borde	Foco	Ruidoso/incierto/borde/foco	Total
<i>C1</i>	11	1	478	585	34	1109
<i>C2</i>	668	6653	35665	23777	20669	87432
<i>C3</i>	12	18	993	135	1271	2429
<i>C4</i>	502	1077	32082	16527	7413	57601
<i>C5</i>	83	32	5007	3201	2428	10751
<i>C6</i>	6	1	622	26	23	678

Tabla 4.23: Píxeles ruidosos, inciertos, bordes, focos y erróneos. *ACA mínima distancia*.

### 4.2.5. Conclusiones

Como vimos en el apartado anterior, el algoritmo ACA mejora la tasa de acierto en las imágenes de satélite de baja, media y elevada complejidad, proporciona una clasificación jerárquica dividida en niveles de fiabilidad basándose en el grado de pertenencia de los píxeles a sus clases, y obtiene una lista de píxeles ruidosos, inciertos, bordes y focos. Además, como hemos visto en este apartado, el algoritmo ACA también proporciona una gran cantidad de información complementaria dividida por iteraciones, que puede servir para analizar más en profundidad los resultados obtenidos por el proceso de clasificación de la imagen de satélite.

En primer lugar, el algoritmo ACA proporciona una **matriz de confusión intermedia** para cada iteración del autómata celular, de tal modo que el analista experto puede estudiar la evolución de dichas matrices de confusión a lo largo del proceso de clasificación de la imagen de satélite. Gracias a estas matrices de confusión intermedias se puede analizar en profundidad no sólo el comportamiento del algoritmo ACA en cada imagen de satélite concreta, sino además estudiar la evolución de la tasa de acierto a lo largo del proceso de clasificación. Existen una serie de matrices de confusión que son claves para estudiar dicho proceso de clasificación, ya que corresponden con eventos importantes, y por lo tanto nos ofrecen información complementaria para poder analizar más en profundidad los resultados obtenidos.

En segundo lugar, podemos obtener los valores de **precisión** y **recall** de cada una de las clases, para cada iteración del autómata celular. Esta información permite al analista experto estudiar en profundidad la evolución de dichos valores en las distintas clases a lo largo del proceso de clasificación de la imagen de satélite, y por lo tanto también sirve de información complementaria para poder analizar los resultados obtenidos con mayor detenimiento y obtener nuevas conclusiones.

En tercer lugar, el algoritmo ACA obtiene una **tabla intermedia de píxeles** con información desglosada de cada tipo de píxel, para cada iteración del autómata celular. Esta información puede servir al analista experto para complementar los resultados obtenidos por las matrices de confusión intermedias, así como los resultados obtenidos por las tablas de píxeles basadas en porcentajes, vistas en el apartado anterior.

Es importante indicar que el algoritmo ACA dispone de una opción que permite calcular o no todos estos resultados complementarios por iteraciones, de tal modo que el analista experto puede decidir si desea optar entre aumentar la velocidad de la clasificación, desactivando el cálculo de los resultados complementarios, o por el contrario desea obtener la mayor cantidad de información posible a partir del proceso de clasificación, activando el cálculo de dichos resultados.

Como ventaja final del uso de autómatas celulares para la clasificación de imágenes de satélite, podemos concluir que permiten configurar una clasificación personalizada de cada imagen de satélite en particular, para cada área de estudio específico, modificando sólo algunas de las propiedades de los autómatas celulares: vecindad y número de iteraciones. Además, también se pueden implementar, mediante programación, nuevas reglas y estados para obtener nuevos resultados.

### 4.3. FORMATO DE SALIDA DEL ALGORITMO ACA

En este apartado se muestra el índice completo de los archivos de salida del algoritmo ACA, la configuración de la clasificación jerárquica según el número de iteraciones y el color de las clases, y el formato de salida de las matrices de confusión intermedias y de los datos de las gráficas de evolución del acierto, la precisión y el *recall*.

#### 4.3.1. Índice completo de los archivos de salida

El algoritmo ACA ofrece una serie de archivos de salida como resultado del proceso de clasificación. La mayoría de estos archivos de salida son opcionales, de tal modo que se pueden omitir para aumentar la velocidad de ejecución de la clasificación. De este modo, el analista experto puede elegir entre incrementar la velocidad de ejecución o aumentar la cantidad de resultados obtenidos. El índice completo de los archivos de salida proporcionados por parte del algoritmo ACA es el siguiente:

- **Clasificación jerárquica:** sucesión de imágenes de satélite en formato *IMG* que almacenan las distintas capas de fiabilidad en las que se divide el proceso de clasificación, donde cada capa corresponde a una iteración del autómata celular, y puede estar representada en escalas de grises o colores (*objetivo #2*).
- **Información de las clases:** archivo de texto que almacena, para cada clase de la imagen, su grado de heterogeneidad (*objetivo #2*), así como el porcentaje de píxeles ruidosos, inciertos, bordes y focos correspondiente, tanto bien como mal clasificados (*objetivo #3*).
- **Información de los píxeles:** archivo de texto que almacena, para cada píxel de la imagen, los tres estados del autómata celular, es decir, la clase correspondiente, la iteración en la que se clasificó y el tipo de píxel (*objetivos #1, #2 y #3*), así como su grado de pertenencia a la clase (*objetivo #2*).
- **Matrices de confusión intermedias:** archivo de texto que almacena la información de las distintas matrices de confusión intermedias, para cada una de las iteraciones del autómata celular (*información complementaria del objetivo #1*).
- **Gráficas de evolución del acierto, precisión y *recall*:** archivo de texto que almacena la información de los resultados numéricos de las gráficas de evolución por iteraciones para la tasa de acierto (*objetivo #1 e información complementaria del objetivo #1*), así como la precisión y el *recall* de cada clase (*información complementaria del objetivo #2*).
- **Tablas intermedias de píxeles:** archivo de texto que almacena la información relacionada con la evolución de las tablas intermedias de píxeles para cada iteración (*información complementaria del objetivo #3*).

Las siguientes secciones muestran los aspectos básicos de la configuración de la clasificación jerárquica del algoritmo ACA, así como el formato del archivo de las matrices de confusión intermedias, y el formato del archivo de las gráficas de evolución del acierto, la precisión y el *recall*.

### 4.3.2. Clasificación jerárquica: número de iteraciones y color

El algoritmo ACA se puede configurar para obtener diferentes resultados relacionados con el número de capas de fiabilidad de la *clasificación jerárquica*. Para ello, el analista experto sólo tiene que especificar el número de iteraciones en las que desea dividir el proceso de clasificación de la imagen de satélite, y el *threshold* es recalculado automáticamente por parte del algoritmo ACA. De esta forma, si el analista experto selecciona un número de iteraciones bajo, el *threshold* será elevado, y si selecciona un número de iteraciones elevado, el *threshold* será bajo. El *threshold* se ajusta automáticamente para que se cumplan los siguientes objetivos:

- Todos los *píxeles* de la imagen de satélite son *clasificados*, es decir, no quedan píxeles sin etiquetar al final del proceso de clasificación.
- Todas las *iteraciones* son *útiles*, es decir, no existen iteraciones en las que no se clasifiquen píxeles.

La *Figura 4.17* muestra la imagen de satélite original de las provincias de Almería y Granada, así como la zona de estudio tomada para experimentar y mostrar las diferentes configuraciones de la clasificación jerárquica del algoritmo ACA.

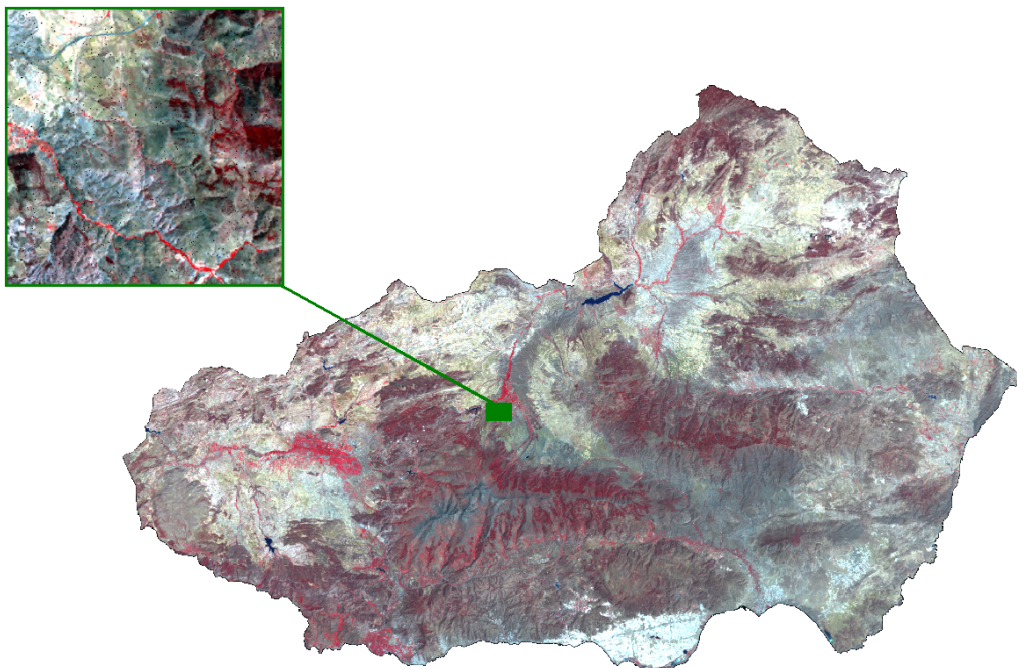


Figura 4.17: Zona de estudio tomada de la imagen de satélite original de las provincias de Almería y Granada (España).

La imagen de satélite de la zona de estudio es de tipo *Landsat TM* multispectral de 7 bandas, con una resolución espacial de  $30 \times 30$  m y un tamaño de  $300 \times 300$  píxeles, es decir, un total de 90000 píxeles. Se trata de una región geográfica con un porcentaje de píxeles inciertos extremadamente elevado debido a la enorme heterogeneidad del terreno y un porcentaje de ruido artificial significativo. Dicha región geográfica es tan heterogénea que está caracterizada por la presencia de variables ecológicas de los 6 sectores de vegetación, previamente establecidos en la *Tabla 4.1*. El número de clases ha sido establecido a un total de 8, representando a los 6 sectores de vegetación. Se han elegido muestras de las 8 clases distintas, y a continuación se ha aplicado el algoritmo ACA, utilizando dicho conjunto de entrenamiento. El algoritmo ACA permite mostrar los resultados de la clasificación jerárquica en escala de grises o en colores, para representar las distintas clases, y además permite elegir el número de iteraciones del autómata celular, en las que se divide el proceso de clasificación. La *Figura 4.18* muestra la clasificación jerárquica obtenida por el algoritmo *ACA paralelepípedos* de la zona de estudio, dividida en 6 capas de fiabilidad con las 8 clases representadas en escalas de grises, y las *Figuras 4.19* y *4.20* muestran la misma clasificación, dividida en 30 capas con las 8 clases representadas por diferentes colores.

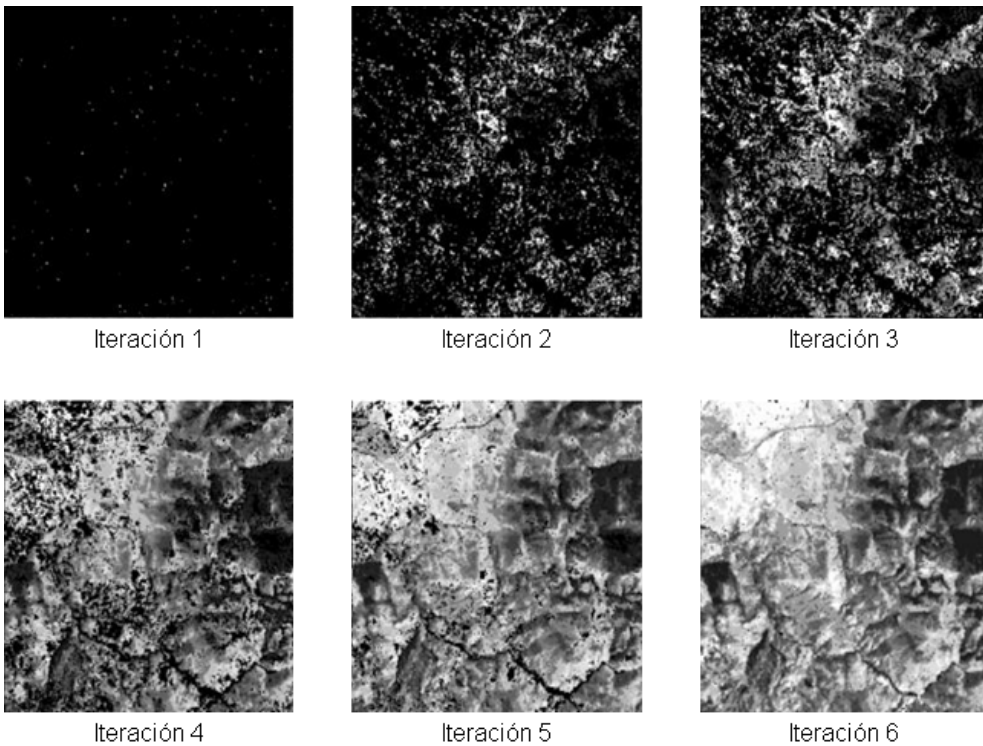


Figura 4.18: Clasificación jerárquica dividida en 6 capas de fiabilidad.

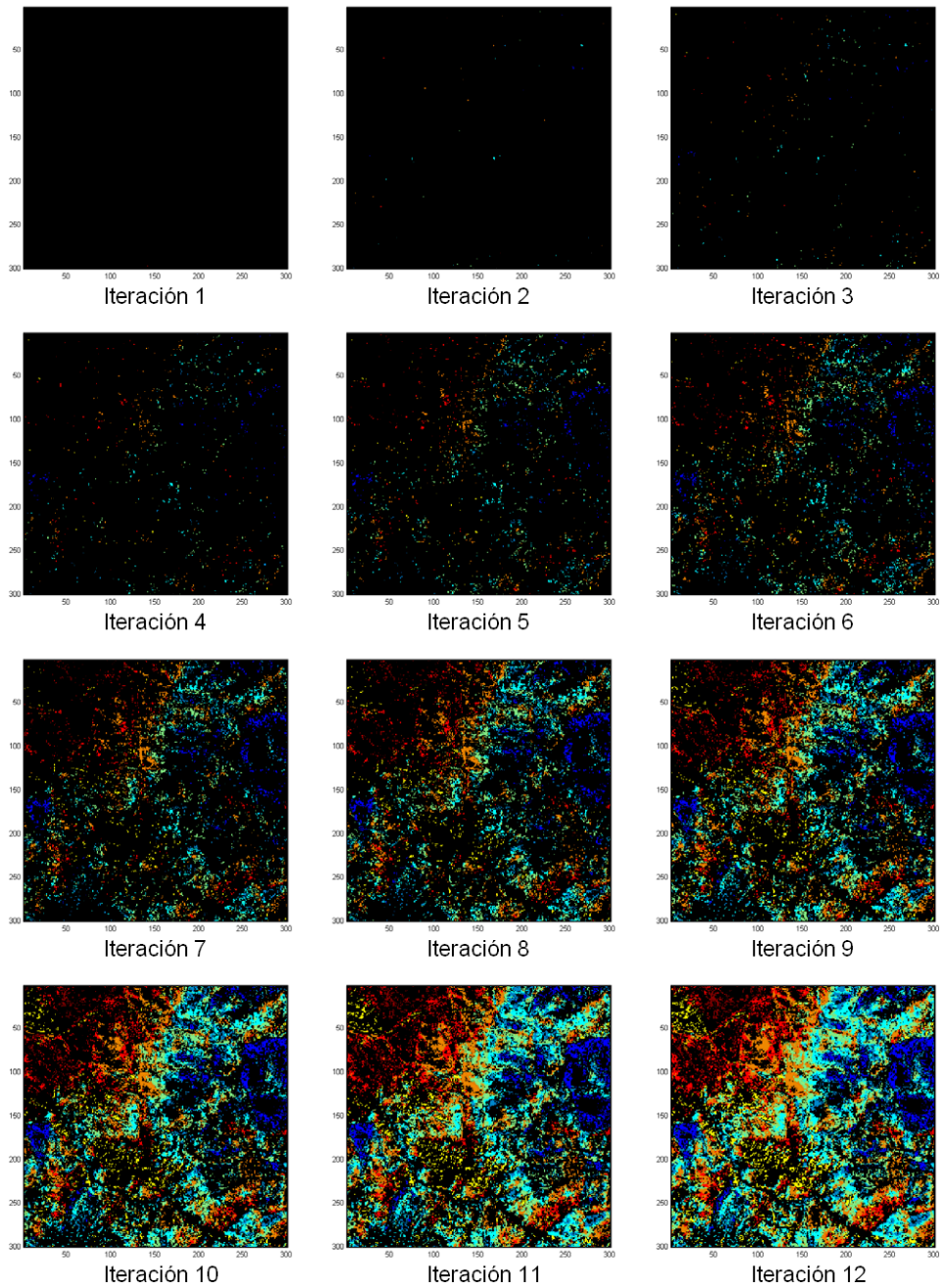


Figura 4.19: Iteraciones 1-12 de la clasificación *ACA paralelepípedos*.



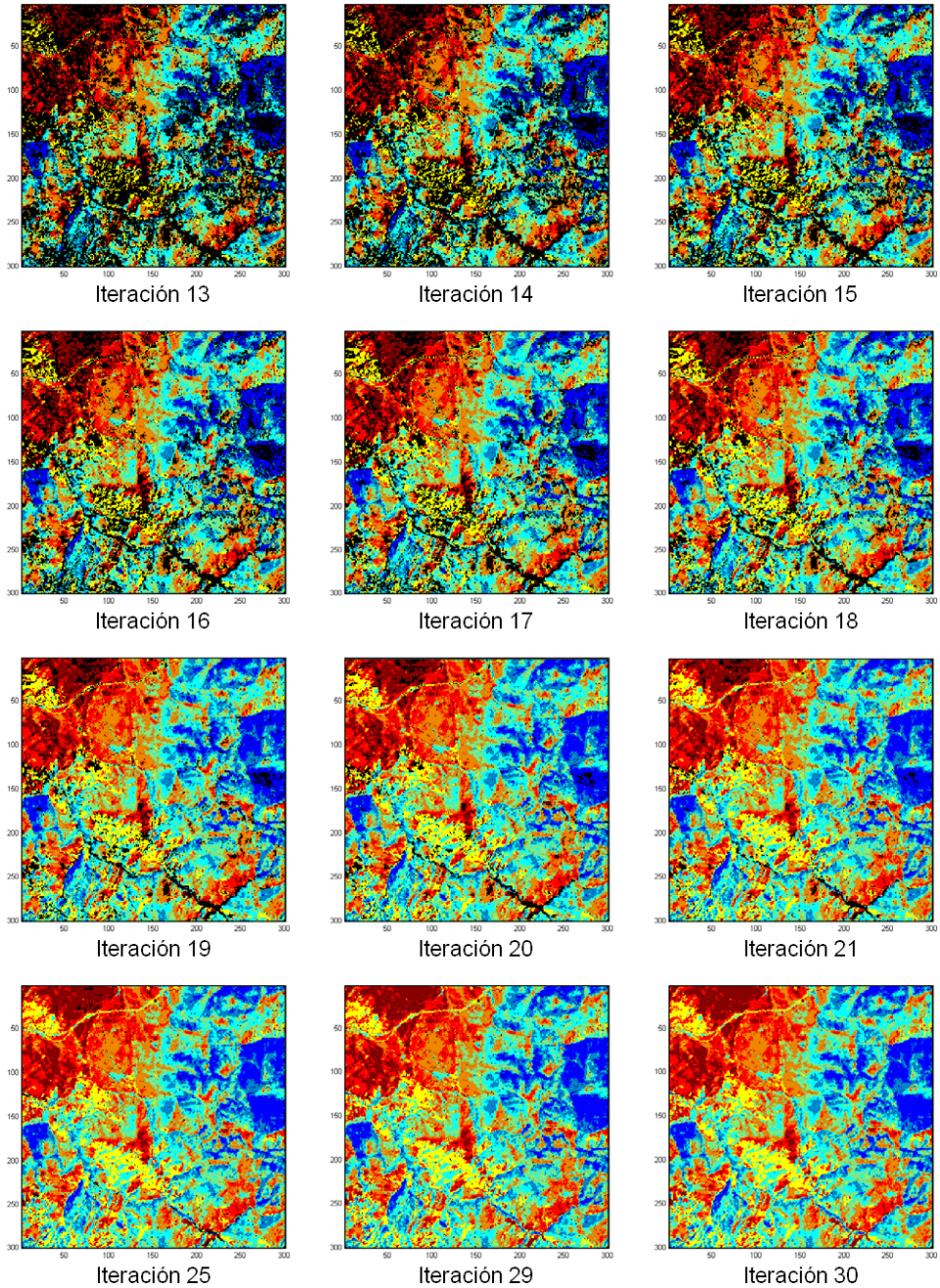


Figura 4.20: Iteraciones 13-30 de la clasificación *ACA paralelepípedos*.



### 4.3.3. Matrices de confusión intermedias por iteraciones

El algoritmo ACA ofrece, como información de salida, un archivo de datos de las *matrices de confusión intermedias*, es decir, una matriz para cada iteración del autómata celular. La *Figura 4.21* muestra el formato de salida del archivo de las matrices de confusión intermedias por iteraciones para la imagen de satélite de *El Ejido Oeste* utilizada en los dos apartados anteriores, hasta la iteración número 6, con dos capturas de pantalla del archivo de texto en la misma imagen.

```

Matrices - Paralelepipedos.dat
Archivo Edición Formato Ver Ayuda

Iteración: 1
Píxeles bien clasificados: 2
Píxeles mal clasificados: 159998
Accuracy: 0.000012
Balanced error rate: 0.000000
Precision (por clase): 0.00 0.00 0.00 0.00 1.00 0.00
Recall (por clase): 0.00 0.00 0.00 0.00 1.00 0.00
Matriz de confusión:
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

-----
Iteración: 2
Píxeles bien clasificados: 137
Píxeles mal clasificados: 159863
Accuracy: 0.000856
Balanced error rate: 0.068966
Precision (por clase): 1.00 1.00 0.00 1.00 1.00 0.91
Recall (por clase): 1.00 1.00 0.00 0.93 1.00 1.00
Matriz de confusión:
9 0 0 0 0
0 3 0 0 0
0 0 0 0 0
0 0 0 27 0
0 0 0 0 78
0 0 0 0 20

-----
Iteración: 3
Píxeles bien clasificados: 803
Píxeles mal clasificados: 159197
Accuracy: 0.005019
Balanced error rate: 0.023529
Precision (por clase): 1.00 1.00 1.00 1.00 1.00 0.93
Recall (por clase): 1.00 1.00 1.00 0.98 1.00 1.00
Matriz de confusión:
35 0 0 0 0
0 12 0 0 0
0 0 7 0 0
0 0 0 332 0
0 0 0 0 317
0 0 0 0 100

-----
Iteración: 4
Píxeles bien clasificados: 2209
Píxeles mal clasificados: 157791
Accuracy: 0.013806
Balanced error rate: 0.061397
Precision (por clase): 1.00 1.00 1.00 1.00 1.00 0.82
Recall (por clase): 1.00 1.00 0.98 0.96 1.00 1.00
Matriz de confusión:
127 0 0 0 0
0 56 0 0 0
0 0 44 0 0
0 0 0 932 1
0 0 0 0 880
0 0 0 0 170

-----
Iteración: 5
Píxeles bien clasificados: 4652
Píxeles mal clasificados: 155348
Accuracy: 0.029075
Balanced error rate: 0.085165
Precision (por clase): 1.00 1.00 1.00 1.00 0.99 0.72
Recall (por clase): 1.00 1.00 0.96 0.96 1.00 1.00
Matriz de confusión:
244 0 0 0 0
0 159 0 0 0
0 0 90 0 4
0 0 0 2400 4
0 0 0 0 1490
0 0 0 0 269

-----
Iteración: 6
Píxeles bien clasificados: 7771
Píxeles mal clasificados: 152229
Accuracy: 0.048569
Balanced error rate: 0.180639
Precision (por clase): 1.00 1.00 1.00 1.00 0.98 0.83
Recall (por clase): 1.00 1.00 0.87 0.95 1.00 1.00
Matriz de confusión:
383 0 0 0 0
0 475 0 1 0
0 0 141 0 21
0 0 0 4181 21
0 0 0 0 2255
0 0 0 0 336
  
```

Figura 4.21: Archivo de datos de los resultados de las matrices de confusión intermedias, proporcionadas por el algoritmo *ACA paralelepípedos* para la imagen de *El Ejido Oeste*.

Como se puede observar, cada sección comienza con el número de iteración del autómata celular. En segundo lugar se muestra el número de píxeles bien clasificados de la matriz de confusión, es decir, la suma del número de píxeles presentes en la diagonal principal. En tercer lugar se muestra el número de píxeles mal clasificados, que en realidad es un valor que representa la suma de dos valores: los píxeles mal clasificados, es decir, los que se encuentran fuera de la diagonal principal de la matriz de confusión correspondiente, y los píxeles que todavía no se han clasificado. A continuación se muestra la tasa de acierto y el error balanceado para la iteración correspondiente. Seguidamente se muestran los valores de precisión y *recall* para cada una de las clases que intervienen en el proceso de clasificación. Y para finalizar, y lo más importante, se muestra la matriz de confusión intermedia de la iteración correspondiente.

#### 4.3.4. Datos de las gráficas evolutivas: acierto, precisión y *recall*

El algoritmo ACA también ofrece, como información de salida, un archivo de datos de los resultados numéricos de la *gráficas de evolución iterativas* para el *acierto*, la *precisión* y el *recall*. La *Figura 4.22* muestra el formato de salida de los resultados numéricos de las gráficas de evolución para la imagen de satélite de *El Ejido Oeste* utilizada en los dos apartados anteriores, hasta la iteración número 40.

#	tpi	ejk	acierto	pre1	pre2	pre3	pre4	pre5	pre6	rec1	rec2	rec3	rec4	rec5	rec6
1	2	159998	0.000012	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
2	137	159863	0.000856	1.00	1.00	0.00	1.00	1.00	0.91	1.00	1.00	0.93	1.00	1.00	1.00
3	803	159197	0.005019	1.00	1.00	1.00	1.00	1.00	0.93	1.00	1.00	1.00	0.98	1.00	1.00
4	2209	157791	0.013806	1.00	1.00	1.00	1.00	1.00	0.82	1.00	1.00	0.98	0.96	1.00	1.00
5	4652	155348	0.029075	1.00	1.00	1.00	1.00	0.99	0.72	1.00	1.00	0.96	0.96	1.00	1.00
6	7771	152229	0.048569	1.00	1.00	1.00	1.00	0.98	0.63	1.00	1.00	0.87	0.95	1.00	1.00
7	11944	148036	0.074650	1.00	1.00	1.00	1.00	0.97	0.52	1.00	0.99	0.84	0.95	1.00	1.00
8	16658	144342	0.104112	0.99	1.00	1.00	1.00	0.94	0.45	1.00	0.97	0.78	0.94	1.00	1.00
9	21968	138032	0.137300	0.99	1.00	1.00	0.99	0.92	0.35	1.00	0.96	0.75	0.93	0.98	1.00
10	27567	132433	0.172294	0.99	1.00	1.00	0.99	0.88	0.31	1.00	0.95	0.70	0.92	0.98	1.00
11	32881	127119	0.205506	0.98	1.00	0.99	0.98	0.85	0.26	1.00	0.95	0.67	0.91	0.96	1.00
12	38120	121880	0.238250	0.98	1.00	0.99	0.98	0.79	0.23	1.00	0.94	0.61	0.90	0.95	1.00
13	43520	116480	0.272000	0.98	1.00	0.92	0.96	0.76	0.21	1.00	0.93	0.60	0.89	0.90	1.00
14	48942	111058	0.305887	0.97	1.00	0.89	0.95	0.73	0.19	1.00	0.93	0.57	0.88	0.87	0.99
15	53278	106722	0.332987	0.96	1.00	0.84	0.94	0.71	0.16	1.00	0.92	0.56	0.87	0.82	0.99
16	66498	93502	0.415612	0.96	1.00	0.83	0.94	0.69	0.15	1.00	0.95	0.55	0.86	0.80	0.98
17	70310	89690	0.439438	0.95	1.00	0.78	0.93	0.69	0.14	0.99	0.95	0.54	0.85	0.78	0.98
18	74687	85313	0.466794	0.94	1.00	0.76	0.92	0.68	0.13	0.99	0.94	0.53	0.85	0.77	0.98
19	78259	81741	0.489119	0.93	1.00	0.73	0.92	0.68	0.12	0.98	0.94	0.52	0.85	0.75	0.96
20	81944	78056	0.512150	0.92	1.00	0.71	0.91	0.67	0.12	0.98	0.93	0.52	0.85	0.74	0.96
21	84957	75043	0.530981	0.91	1.00	0.69	0.91	0.67	0.12	0.99	0.93	0.52	0.85	0.73	0.96
22	88083	71917	0.550519	0.91	1.00	0.68	0.90	0.67	0.12	0.98	0.92	0.52	0.85	0.73	0.96
23	91068	68932	0.569175	0.90	1.00	0.67	0.89	0.67	0.11	0.98	0.92	0.51	0.86	0.73	0.96
24	94307	65993	0.589419	0.90	1.00	0.66	0.89	0.67	0.11	0.97	0.91	0.51	0.86	0.72	0.96
25	97283	62717	0.608019	0.89	1.00	0.66	0.88	0.67	0.11	0.97	0.91	0.52	0.86	0.72	0.96
26	100148	59852	0.625925	0.89	1.00	0.66	0.87	0.67	0.11	0.97	0.90	0.52	0.86	0.72	0.96
27	102912	57088	0.643200	0.89	1.00	0.66	0.87	0.67	0.11	0.97	0.90	0.52	0.86	0.72	0.96
28	105491	54509	0.659319	0.89	1.00	0.66	0.86	0.67	0.11	0.97	0.89	0.52	0.86	0.72	0.96
29	108202	51798	0.676262	0.89	1.00	0.66	0.85	0.67	0.11	0.97	0.89	0.52	0.86	0.72	0.96
30	110893	49107	0.693081	0.89	1.00	0.66	0.84	0.67	0.11	0.97	0.88	0.52	0.86	0.72	0.96
31	113355	46645	0.708469	0.89	1.00	0.66	0.83	0.67	0.11	0.97	0.88	0.52	0.86	0.72	0.96
32	115747	44253	0.723419	0.89	1.00	0.66	0.82	0.67	0.11	0.97	0.87	0.52	0.86	0.72	0.96
33	118079	41921	0.737994	0.89	1.00	0.66	0.82	0.67	0.11	0.97	0.87	0.52	0.86	0.72	0.96
34	120075	39925	0.750469	0.89	1.00	0.66	0.80	0.67	0.11	0.97	0.86	0.52	0.86	0.72	0.96
35	122460	37540	0.765375	0.89	1.00	0.66	0.80	0.67	0.11	0.97	0.85	0.52	0.86	0.72	0.96
36	124050	35950	0.775312	0.89	1.00	0.66	0.78	0.67	0.11	0.97	0.84	0.52	0.86	0.72	0.96
37	125868	34132	0.786675	0.89	0.99	0.66	0.78	0.67	0.11	0.97	0.84	0.52	0.86	0.72	0.96
38	126795	33205	0.792469	0.89	0.99	0.66	0.78	0.67	0.11	0.97	0.84	0.51	0.86	0.72	0.96
39	127636	32364	0.797725	0.89	0.99	0.66	0.77	0.67	0.11	0.97	0.84	0.51	0.86	0.72	0.96
40	128441	31559	0.802756	0.89	0.99	0.66	0.77	0.67	0.11	0.97	0.84	0.51	0.86	0.72	0.96

Figura 4.22: Archivo de datos de los resultados numéricos de las gráficas de evolución, proporcionadas por el algoritmo *ACA paralelepípedos* para la imagen de *El Ejido Oeste*.

Aunque el formato de este archivo de salida es completamente distinto al de la sección anterior, realmente su contenido es una repetición de la información proporcionada por el archivo de salida de las matrices de confusión. Esto se debe a que el formato del archivo de salida del algoritmo ACA con estos resultados numéricos es compatible con *Gnuplot*, por lo que posteriormente se pueden representar las distintas gráficas de evolución iterativas a través del *script* correspondiente. Como se puede observar, la primera columna indica el número de iteración del autómata celular, por lo que cada fila del archivo de datos representa la información de una iteración determinada. La segunda columna, llamada *tpi*, indica el número de píxeles bien clasificados en la matriz de confusión para la iteración correspondiente. La tercera columna, llamada *ejk*, representa la suma de dos valores: los píxeles mal clasificados y los píxeles que todavía no se han clasificado. La cuarta columna indica el acierto de cada iteración, valor que se calcula dividiendo los dos valores anteriores. Las siguientes columnas indican los valores de precisión y *recall* para cada una de las 6 clases en los que se han etiquetado los píxeles de la imagen de satélite de *El Ejido Oeste*.

## 4.4. RESUMEN Y CONCLUSIONES DEL CAPÍTULO

Los resultados obtenidos por parte del algoritmo ACA han sido muy satisfactorios desde varios puntos de vista, por lo que los autómatas celulares han sido validados como una técnica que mejora los resultados del proceso de clasificación de imágenes de satélite. El algoritmo ACA ha conseguido alcanzar los tres objetivos establecidos en el *Capítulo 3*.

En primer lugar, en relación con el **objetivo #1**, el algoritmo ACA ha conseguido una *tasa de acierto más elevada* que la mayoría de los algoritmos de clasificación en imágenes de satélite con complejidad baja y media, y más elevada que todos los clasificadores en imágenes de satélite con complejidad elevada. Además, conforme vamos incrementando el número de píxeles inciertos y ruidosos, más va aumentando la ventaja de la tasa de acierto del algoritmo ACA respecto al resto de clasificadores. En lo que respecta al *coste computacional*, el algoritmo ACA tiene una complejidad media. Además, no sólo ha mejorado la tasa de acierto de los algoritmos *paralelepípedos* y *mínima distancia* clásicos, sino que además ha *optimizado su funcionamiento*, solventando las limitaciones descritas en el *Capítulo 3*. Por otro lado, el algoritmo ACA también ofrece la posibilidad de obtener las *matrices de confusión intermedias* de cada iteración del autómata celular, que sirven para comprender mejor el funcionamiento del proceso de clasificación, así como la *evolución de la tasa de acierto*, por iteraciones (**información complementaria del objetivo #1**).

En segundo lugar, en relación con el **objetivo #2**, el algoritmo ACA obtiene una *clasificación jerárquica* dividida en niveles o capas de fiabilidad basadas en el grado de pertenencia de los píxeles a las clases correspondientes, donde cada capa corresponde a una iteración del autómata celular. De este modo, el algoritmo ACA simula una clasificación *pseudodifusa*, en la que se puede calcular el *grado de pertenencia* de cada píxel a su clase correspondiente, teniendo en cuenta la iteración en la que fue clasificado. También podemos obtener el *grado de heterogeneidad* de las distintas clases de la imagen de satélite. Por otro lado, el algoritmo ACA también ofrece la posibilidad de obtener las *gráficas de evolución* de la *precisión* y el *recall* de las distintas clases que intervienen en el proceso de clasificación de la imagen de satélite, por iteraciones (**información complementaria del objetivo #2**).

En tercer lugar, en relación con el **objetivo #3**, el algoritmo ACA obtiene el porcentaje de píxeles ruidosos, inciertos, focos y bordes de cada una de las clases que intervienen en el proceso de clasificación, tanto bien como mal clasificados. El análisis de los píxeles *inciertos y ruidosos* muestran la contribución de cada clase concreta a la *mejora de la tasa de acierto*, mientras que el análisis de los píxeles *borde y foco* determinan la *distribución espacial* de los píxeles de cada clase en la imagen de satélite. Por otro lado, el algoritmo ACA también ofrece la posibilidad de obtener las *tablas intermedias de píxeles*, con información detallada de los tipos de píxel de cada clase concreta, por iteraciones (**información complementaria del objetivo #3**).

El *Capítulo 5* muestra las distintas aportaciones realizadas a la comunidad científica por parte de esta tesis doctoral, no sólo por parte del algoritmo ACA, sino también por parte de los algoritmos RACA y TACA vistos en el *Capítulo 2*, así como de otros trabajos complementarios relacionados con el ámbito de la teledetección. También muestra las publicaciones derivadas y las líneas de investigación abiertas.



---

## CAPÍTULO 5

# APORTACIONES Y LÍNEAS DE INVESTIGACIÓN ABIERTAS

---



# Capítulo 5

## APORTACIONES Y LÍNEAS DE INVESTIGACIÓN ABIERTAS

### Contenidos

---

<b>5.1.</b>	<b>APORTACIONES A LA COMUNIDAD CIENTÍFICA . . . . .</b>	<b>181</b>
5.1.1.	Clasificación de imágenes de satélite con ACA: objetivo#1 . . . . .	182
5.1.2.	Clasificación de imágenes de satélite con ACA: objetivo#2 . . . . .	183
5.1.3.	Clasificación de imágenes de satélite con ACA: objetivo#3 . . . . .	183
5.1.4.	Simulación de precipitaciones con el algoritmo RACA . . . . .	184
5.1.5.	Caracterización de texturas con el algoritmo TACA . . . . .	185
5.1.6.	Correlación de variables ecológicas con <i>RBFs</i> . . . . .	185
<b>5.2.</b>	<b>PUBLICACIONES DERIVADAS . . . . .</b>	<b>186</b>
5.2.1.	Clasificación de imágenes de satélite: algoritmo ACA . . . . .	186
5.2.2.	Autómatas celulares: algoritmos RACA y TACA . . . . .	187
5.2.3.	Teledetección: correlación de variables ecológicas con <i>RBFs</i> . . . . .	187
<b>5.3.</b>	<b>LÍNEAS DE INVESTIGACIÓN ABIERTAS . . . . .</b>	<b>188</b>
5.3.1.	Clasificación de imágenes de satélite con ACA . . . . .	188
5.3.2.	Simulación de precipitaciones con el algoritmo RACA . . . . .	189
5.3.3.	Caracterización de texturas con el algoritmo TACA . . . . .	189

---





**L**a clasificación de imágenes de satélite mediante el algoritmo ACA ha obtenido unos resultados muy satisfactorios desde varios puntos de vista, como hemos visto en el *Capítulo 4*. Esta tesis doctoral también ha abarcado otras líneas de investigación distintas a las del algoritmo ACA, como por ejemplo la simulación de precipitaciones sobre imágenes *DEM* mediante el algoritmo RACA, y la caracterización de texturas de imágenes-AC y naturales mediante el algoritmo TACA, relacionadas con los autómatas celulares, y la correlación de variables ecológicas mediante *RBFs*, dentro del ámbito de la teledetección. Este capítulo ofrece las aportaciones, publicaciones y líneas de investigación abiertas de esta tesis doctoral, y está dividido en los siguientes apartados principales:

- **Aportaciones a la comunidad científica:** en este apartado se ofrece un resumen de las aportaciones realizadas, por parte de esta tesis doctoral, relacionadas con la clasificación de imágenes de satélite mediante el algoritmo ACA, divididas en los tres objetivos establecidos, la simulación de precipitaciones mediante el algoritmo RACA, la caracterización de texturas mediante el algoritmo TACA, y la correlación de variables ecológicas mediante *RBFs*.
- **Publicaciones derivadas:** en este apartado se muestran los artículos y capítulos de libro que se han publicado, relacionados con las distintas líneas de investigación abiertas durante el desarrollo de esta tesis doctoral, divididas en tres categorías diferentes: clasificación supervisada de imágenes de satélite mediante autómatas celulares a través del algoritmo ACA, publicaciones específicas sobre autómatas celulares, relacionadas con los algoritmos RACA y TACA, y publicaciones específicas dentro del ámbito de la teledetección, relacionadas con la correlación de variables ecológicas mediante *RBFs*.
- **Líneas de investigación abiertas:** en este apartado se muestran las distintas líneas de investigación abiertas en las que se puede profundizar en un futuro, para mejorar el trabajo de investigación que se ha realizado durante el desarrollo de esta tesis doctoral. Al igual que los apartados anteriores, dichas líneas de investigación abiertas se han dividido en los tres algoritmos principales implementados: clasificación de imágenes de satélite mediante el algoritmo ACA, simulación de precipitaciones en imágenes *DEM* mediante el algoritmo RACA, y caracterización de texturas de imágenes-AC y naturales mediante el algoritmo TACA.

## 5.1. APORTACIONES A LA COMUNIDAD CIENTÍFICA

En este apartado se ofrece un resumen de las aportaciones realizadas, por parte de esta tesis doctoral, relacionadas con la clasificación de imágenes de satélite mediante el algoritmo ACA, divididas en los tres objetivos establecidos, la simulación de precipitaciones mediante el algoritmo RACA, la caracterización de texturas mediante el algoritmo TACA, y la correlación de variables ecológicas mediante *RBFs*.

### 5.1.1. Clasificación de imágenes de satélite con ACA: objetivo #1

El *objetivo #1* del algoritmo ACA es el más importante de los tres, ya que la principal finalidad de cualquier algoritmo de clasificación consiste en ofrecer la mejor tasa de acierto posible. El algoritmo ACA ha conseguido realizar numerosas aportaciones relacionadas con este objetivo, y las más importantes son las siguientes:

- *Aportación 1.* El algoritmo ACA está especialmente recomendado para las imágenes de satélite que presentan mayores dificultades durante el proceso de clasificación, aunque realmente se puede utilizar con cualquier tipo de imagen. Obtiene una tasa de acierto muy buena en imágenes de satélite de complejidad de clasificación baja y media, y excelente en imágenes de complejidad elevada.
- *Aportación 2.* En la comparativa realizada con otros algoritmos de clasificación ampliamente utilizados (*C4.5*, *multilayer perceptron*, *naive Bayes*, *k-NN* y *RBF network*), la tasa de acierto del algoritmo ACA es más elevada que la de la mayoría de dichos algoritmos en imágenes de complejidad baja y media, y la más elevada en imágenes de alta complejidad.
- *Aportación 3.* La evolución de la tasa de acierto del algoritmo ACA es mejor que el resto de algoritmos de clasificación utilizados conforme vamos aumentando la complejidad de clasificación de las imágenes de satélite. Por lo tanto, conforme aumenta el número de píxeles inciertos y ruidosos en la imagen, aumenta también la ventaja de la tasa de acierto del algoritmo ACA.
- *Aportación 4.* El algoritmo ACA presenta un clasificación espectral-contextual mixta basada en un autómata celular multiestado, que mejora la tasa de acierto de los algoritmos clásicos gracias a las reglas establecidas para el autómata celular.
- *Aportación 5.* El coste computacional del algoritmo ACA tiene una complejidad media, ya que no es tan rápido en ejecución como los algoritmos *k-NN*, *naive Bayes* o *multilayer perceptron*, pero tampoco es tan lento como los algoritmos *C4.5* y *RBF network*.
- *Aportación 6.* El algoritmo ACA mejora el funcionamiento de los algoritmos *paralelepípedos* y *mínima distancia* clásicos, eliminando por completo las limitaciones de dichos algoritmos. En el algoritmo *ACA paralelepípedos* no quedan píxeles sin clasificar, y no existen píxeles clasificados en varias clases. En el algoritmo *ACA mínima distancia*, se reducen los errores de comisión y se simula la varianza durante el proceso de clasificación.
- *Aportación 7.* El funcionamiento de los *SIG* puede ser mejorado si utilizan el algoritmo ACA como algoritmo de clasificación de imágenes de satélite, ya que mejora la tasa de acierto en imágenes complejas de clasificar.
- *Aportación 8.* El algoritmo ACA proporciona información complementaria por iteraciones relacionada con el *objetivo #1*, concretamente las matrices de confusión intermedias, que nos permiten estudiar el funcionamiento del algoritmo y analizar la evolución de la tasa de acierto.

### 5.1.2. Clasificación de imágenes de satélite con ACA: objetivo#2

El *objetivo #2* del algoritmo ACA, por un lado, ayuda a que se consigan los buenos resultados del objetivo anterior, y por otro lado, ofrece información adicional relacionada con el proceso de clasificación. El algoritmo ACA ha conseguido realizar numerosas aportaciones relacionadas con este objetivo, y las más importantes son las siguientes:

- *Aportación 9.* El algoritmo ACA proporciona una clasificación jerárquica dividida en niveles o capas de fiabilidad basadas en la proximidad espectral de los píxeles a sus clases correspondientes, de tal manera que las primeras capas son más fiables que las últimas, en términos de tasa de acierto.
- *Aportación 10.* La clasificación jerárquica del algoritmo ACA ofrece unos resultados pseudodifusos, en los que podemos calcular el grado de pertenencia de los píxeles a sus clases correspondientes, a través de la fórmula descrita en el *Capítulo 3*.
- *Aportación 11.* La clasificación jerárquica del algoritmo ACA también nos permite calcular el grado de heterogeneidad de las clases.
- *Aportación 12.* Se puede realizar una clasificación personalizada para cada imagen de satélite concreta, modificando algunas de las propiedades del autómata celular del algoritmo ACA, como la vecindad y el número de iteraciones.
- *Aportación 13.* El algoritmo ACA proporciona información complementaria por iteraciones relacionados con el *objetivo #2*, concretamente la evolución de la precisión y el *recall* de cada clase, que nos permiten estudiar con mayor detenimiento el comportamiento de las mismas.

### 5.1.3. Clasificación de imágenes de satélite con ACA: objetivo#3

El *objetivo #3* del algoritmo ACA ofrece información adicional relacionada con los distintos tipos de píxeles que nos podemos encontrar en la imagen de satélite: ruidosos, inciertos, bordes y focos. El algoritmo ACA ha conseguido realizar numerosas aportaciones relacionadas con este objetivo, y las más importantes son las siguientes:

- *Aportación 14.* El análisis de los píxeles ruidosos e inciertos bien clasificados muestran la contribución de cada clase concreta a la mejora en la tasa de acierto.
- *Aportación 15.* El análisis de los píxeles borde y foco proporcionados por el algoritmo ACA determinan la distribución espacial de los píxeles de cada clase.
- *Aportación 16.* El algoritmo ACA ha agrupado la pre-clasificación (reducción de ruido), clasificación y post-clasificación (refinamiento de píxeles inciertos).
- *Aportación 17.* El algoritmo ACA utiliza todos los componentes de los autómatas celulares: estados, reglas, vecindad, iteraciones y función de transición.
- *Aportación 18.* El algoritmo ACA proporciona información complementaria por iteraciones relacionada con el *objetivo #3*, concretamente las tablas intermedias de píxeles, que permiten estudiar la evolución de los píxeles.

#### 5.1.4. Simulación de precipitaciones con el algoritmo RACA

El algoritmo RACA utiliza los autómatas celulares para implementar el modelado y la simulación de precipitaciones sobre imágenes de satélite tridimensionales *DEM*. El algoritmo RACA ha conseguido realizar varias aportaciones, y las más importantes son las siguientes:

- *Aportación 19.* El proceso de modelado y simulación de precipitaciones del algoritmo RACA simplifica la metodología de otros algoritmos existentes en la literatura, ya que ha sido implementado para que el autómata celular utilice sólo dos estados y dos reglas básicas. Por lo tanto, a diferencia de otros algoritmos que utilizan una gran cantidad de parámetros de entrada, el algoritmo RACA apenas necesita parámetros, y por lo tanto no es necesario realizar un estudio previo de la región donde se va a realizar la simulación.
- *Aportación 20.* El algoritmo RACA realiza la simulación de las precipitaciones sobre una imagen de satélite 3D *DEM*, donde cada píxel almacena la altura a la que se encuentra esa zona concreta de la región geográfica. Existen muy pocos trabajos de investigación relacionados con la aplicación de autómatas celulares sobre imágenes *DEM*, por lo que se trata de un campo que aún ofrece un amplio abanico para seguir investigando.
- *Aportación 21.* Se puede personalizar la simulación del algoritmo RACA, especificando tanto la vecindad del autómata celular como el máximo número de iteraciones utilizados para el proceso. De este modo, se pueden obtener distintos resultados, dependiendo de la configuración que el analista experto realice del proceso de simulación.
- *Aportación 22.* Se pueden introducir condiciones iniciales a la simulación realizada por el algoritmo RACA, relacionadas con el nivel de agua inicial almacenado en varias zonas de la región geográfica. Esta opción resulta de verdadera utilidad, sobre todo cuando se desea aplicar varias gráficas de precipitaciones a la misma imagen de satélite *DEM*, y se desea averiguar el caudal de agua remanente tras la simulación de todas las gráficas concatenadas.
- *Aportación 23.* Los resultados tridimensionales del algoritmo RACA permiten realizar una posterior navegación virtual 3D para analizar con mayor detenimiento los resultados obtenidos, eligiendo la perspectiva más adecuada para poder observar con más detalle una región específica de la imagen *DEM* sobre la que se ha realizado la simulación.
- *Aportación 24.* Los resultados completos proporcionados por el algoritmo RACA, para cada gráfica de precipitaciones, son tanto numéricos como tridimensionales, y nos permiten tomar decisiones sobre asuntos tan importantes para una población como la prevención de desastres naturales en regiones urbanas con probabilidad de inundaciones, la proyección de futuros proyectos de construcción y urbanismo, y la estimación del futuro suministro de agua potable de una población.

### 5.1.5. Caracterización de texturas con el algoritmo TACA

El algoritmo TACA utiliza los autómatas celulares para caracterizar texturas de imágenes-AC y naturales. El algoritmo TACA ha conseguido realizar varias aportaciones, y las más importantes son las siguientes:

- *Aportación 25.* El algoritmo TACA permite caracterizar texturas basándose en las propiedades de las imágenes-AC producidas por un modelo de autómata celular unidimensional lineal de 3 celdas, utilizando las reglas de *Stephen Wolfram* a través de un proceso de ingeniería inversa.
- *Aportación 26.* El proceso de caracterización de texturas del algoritmo TACA utiliza como novedad el *HPH* y el *HPV* de la imagen binaria original, para obtener las primitivas más importantes de dichos histogramas, y obtener a partir de dicha información las reglas de *Stephen Wolfram* correspondientes.
- *Aportación 27.* El algoritmo TACA se puede usar tanto para imágenes-AC como para imágenes compuestas por varias imágenes-AC, así como para texturas naturales, concretamente las derivadas del *Álbum Brodatz*, y ofrece como resultado una imagen que caracteriza la textura, formada por la composición de las imágenes-AC, obtenidas por las primitivas más importantes del *HPH* y el *HPV*.

### 5.1.6. Correlación de variables ecológicas con *RBFs*

Durante el tiempo de desarrollo de esta tesis doctoral también se ha participado en otros trabajos paralelos relacionados con el ámbito de la teledetección, como por ejemplo una investigación sobre la correlación de variables ecológicas con *RBFs* (*Radial Basis Function nets*). Aunque no se ha desarrollado la arquitectura principal de esta metodología, sí se ha colaborado en ciertas tareas relacionadas con el tratamiento de las imágenes de satélite que posteriormente han sido utilizadas por las *RBFs* para correlacionar las variables ecológicas. Además, una parte de los resultados obtenidos de esta investigación, realizada en colaboración con un grupo de ecólogos expertos de Almería y Madrid, ha sido usada posteriormente en esta tesis doctoral para realizar el trabajo de campo de las zonas de estudio, y validar así la clasificación obtenida por los distintos algoritmos de clasificación, incluido el algoritmo ACA.

- *Aportación 28.* “Este trabajo de investigación presenta una novedosa aplicación de las redes neuronales con arquitectura de red de función de base radial, como herramienta para la simplificación y reducción de costes en la construcción de mapas de sectorización ecológica. Este proceso agiliza y sustituye al proceso clásico de obtención de las variables ecológicas mediante estudios de campo, demasiado costosos en tiempo. Las redes de función de base radial han sido utilizadas como herramientas que permiten aproximar los datos de campo por información recogida de los satélites *Landsat*, correlacionando esta con variables ecológicas de vegetación, para sustituirlas en el proceso de confección de los mapas. Este trabajo mejora el tiempo y reduce el coste de los estudios de sectorización ecológica, limitando los estudios de campo y automatizando la generación de estas variables” [30] [31].

## 5.2. PUBLICACIONES DERIVADAS

En este apartado se muestran los artículos y capítulos de libro que se han publicado relacionados con las distintas líneas de investigación abiertas durante el desarrollo de esta tesis doctoral, divididas en tres categorías diferentes: 5 publicaciones sobre clasificación de imágenes de satélite mediante autómatas celulares a través del algoritmo ACA, 2 publicaciones específicas sobre autómatas celulares relacionadas con los algoritmos RACA y TACA, y 2 publicaciones específicas sobre teledetección relacionadas con la correlación de variables ecológicas mediante *RBFs*. No obstante, el *Apéndice B* muestra otras publicaciones adicionales relacionadas con ámbitos de investigación diferentes a la tesis.

### 5.2.1. Clasificación de imágenes de satélite: algoritmo ACA

Las publicaciones relacionadas con la temática principal de esta tesis doctoral, clasificación de imágenes de satélite mediante el algoritmo ACA, son las siguientes:

- *Artículo 1.* **Espínola M.**, Piedra J.A., Ayala R., Iribarne L., Wang J.Z., “Contextual and hierarchical classification of satellite images based on cellular automata”, *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 53(2):795–809, doi:10.1109/TGRS.2014.2328634, 2015. (**JCR Q1: 2.933**)
- *Artículo 2.* **Espínola M.**, Piedra J.A., Ayala R., Iribarne L., Leguizamón S., Menenti M., “ACA multiagent system for satellite image classification”, *10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2012)*, March 28-30, Salamanca (Spain), in J.M.C. Rodríguez et al. eds., Trends in Practical Applications of Agents and Multiagent Systems, Springer Advances in Intelligent and Soft Computing, 157, pp. 93–100, doi:10.1007/978-3-642-28795-4\_11, 2012.
- *Artículo 3.* **Espínola M.**, Piedra J.A., Ayala R., Iribarne L., Leguizamón S., Menenti M., “A hierarchical and contextual algorithm based on cellular automata for satellite image classification”, *International Conference on Computational Intelligence and Software Engineering (CiSE 2011)*, December 9-11, Wuhan (China), IEEE Computer Society, ISBN: 978-1-4244-6448-7, 2011.
- *Artículo 4.* **Espínola M.**, Ayala R., Leguizamón S., Iribarne L., Menenti M., “Cellular automata applied in remote sensing to implement contextual pseudo-fuzzy classification”, *9th International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini et al. eds, Springer Lecture Notes in Computer Science 6350, pp. 312–321, doi:10.1007/978-3-642-15979-4\_33, 2010.
- *Artículo 5.* **Espínola M.**, Ayala R., Leguizamón S., Menenti M., “Classification of satellite images using the cellular automata approach”, *First World Summit on the Knowledge Society (WSKS 2008)*, September 24-28, Athens (Greece), in M.D. Lytras et al. eds, The Open Knowledge Society: A Computer Science and Information System Manifesto, Springer Communications in Computer and Information Science 19, doi:10.1007/978-3-540-87783-7\_66, pp. 521–526, 2008.

### 5.2.2. Autómatas celulares: algoritmos RACA y TACA

Las publicaciones relacionadas con los algoritmos RACA y TACA son las siguientes:

- *Artículo 6.* **Espínola M.**, Piedra J.A., Ayala R., Iribarne L., Leguizamón S., “Modeling rainfall features dynamics in a DEM satellite image with cellular automata”, *11th International Conference on Cellular Automata for Research and Industry (ACRI 2014)*, September 22-25, Kraków (Poland), in J.Was et al. eds, Springer Lecture Notes in Computer Science 8751, pp. 238–247, doi:10.1007/978-3-319-11520-7\_25, 2014.
- *Artículo 7.* Leguizamón S., **Espínola M.**, Ayala R., Iribarne L., Menenti M., “Characterization of texture in images by using a cellular automata approach”, *Third World Summit on the Knowledge Society (WSKS 2010)*, September 22-24, Corfu (Greece), in M.D. Lytras et al. eds, The Open Knowledge Society: Organizational, Business, and Technological Aspects of the Knowledge Society, part II, Springer Communications in Computer and Information Science, 112, pp. 522–533, doi:10.1007/978-3-642-16324-1\_61, 2010.

### 5.2.3. Teledetección: correlación de variables ecológicas con *RBFs*

Las publicaciones relacionadas específicamente con la teledetección son las siguientes:

- *Artículo 8.* Cruz M., **Espínola M.**, Iribarne L., Ayala R., Peralta M., Torres J.A., “Ecological sectorization process improvement through neural networks: synthesis of vegetation data from satellite images using RBFs”, *9th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2010)*, August 18-20, Yamagata (Japan), pp. 513–516, doi:10.1109/ICIS.2010.118, 2010.
- *Artículo 9.* Cruz M., **Espínola M.**, Ayala R., Peralta M., Torres J.A., “How can neural networks to speed up ecological regionalization friendly? Replacement of field studies by satellite data using RBFs”, *International Conference on Neural Computation (ICNC 2010)*, October 24-26, Valencia (Spain), 2010.

La *Tabla 5.1* muestra la correspondencia entre las publicaciones descritas en este apartado y las aportaciones enumeradas en el apartado anterior.

Publicaciones	Aportaciones a la comunidad científica
<i>Artículo 1</i>	Aportaciones 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, 16 y 17
<i>Artículos 2, 3 y 4</i>	Aportaciones 1, 4, 6, 9, 10, 12, 16 y 17
<i>Artículo 5</i>	Aportaciones 4 y 17
<i>Artículo 6</i>	Aportaciones 19, 20, 21, 22, 23, 24
<i>Artículo 7</i>	Aportaciones 25, 26 y 27
<i>Artículos 8 y 9</i>	Aportación 28
<i>En fase de publicación</i>	Aportaciones 8, 13 y 18

Tabla 5.1: Correspondencia entre las publicaciones y las aportaciones.

### 5.3. LÍNEAS DE INVESTIGACIÓN ABIERTAS

En este apartado se muestran las distintas líneas de investigación abiertas en las que se puede profundizar en un futuro, con el objetivo de mejorar y ampliar el trabajo de investigación que se ha realizado durante el desarrollo de esta tesis doctoral. Al igual que los apartados anteriores, dichas líneas de investigación abiertas se han dividido en los tres algoritmos principales implementados: la clasificación supervisada de imágenes de satélite mediante el algoritmo ACA, la simulación de precipitaciones en imágenes *DEM* mediante el algoritmo RACA, y la caracterización de texturas de imágenes-AC y naturales mediante el algoritmo TACA.

#### 5.3.1. Clasificación de imágenes de satélite con ACA

Aunque los resultados obtenidos por parte del algoritmo ACA han sido muy satisfactorios desde varios puntos de vista, existen líneas de investigación abiertas que se pueden estudiar para mejorar su funcionalidad y ampliar los resultados obtenidos. Las líneas de investigación abiertas relacionadas con la clasificación de imágenes de satélite mediante el algoritmo ACA son las siguientes:

- *Línea 1.* Desarrollar nuevas versiones del algoritmo de ACA, con nuevas configuraciones de estados y reglas del autómata celular distintas, para personalizar aún más el proceso de clasificación, y poder ajustar lo máximo posible los resultados deseados a la zona de estudio utilizada.
- *Línea 2.* Implementar un algoritmo de clasificación ACA difuso, utilizando funciones adicionales con información espacial y espectral, que complementen la fórmula del grado de pertenencia de la clasificación jerárquica obtenida mediante las iteraciones del autómata celular.
- *Línea 3.* Desarrollar nuevas versiones del algoritmo ACA basadas en clasificadores distintos a los algoritmos *paralelepípedos* y *mínima distancia* clásicos, con el objetivo de mejorar también su funcionamiento y experimentar nuevos resultados.
- *Línea 4.* Añadir un nuevo nivel de clasificación al algoritmo ACA, la clasificación textural, basada en texturas. De este modo, tendríamos dos niveles distintos de clasificación: nivel de píxel (espectral-contextual) y nivel regional (textural).
- *Línea 5.* Utilizar agentes software para reducir el coste computacional del algoritmo ACA, recorriendo y procesando varias regiones de la imagen en paralelo, a lo largo de las iteraciones en las que se divide el proceso de clasificación.
- *Línea 6.* Implementar una versión no supervisada del algoritmo ACA, de tal modo que no necesite un conjunto de entrenamiento de entrada, y pueda determinar el número de clases de la imagen de satélite automáticamente.
- *Línea 7.* Desarrollar un *plugin* con una interfaz gráfica y un editor de reglas-estados del algoritmo ACA para la herramienta *Erdas Imagine*, que permita una clasificación personalizada basada en autómatas celulares.



### 5.3.2. Simulación de precipitaciones con el algoritmo RACA

Las líneas de investigación abiertas relacionadas con la simulación de precipitaciones mediante el algoritmo RACA son los siguientes:

- *Línea 8.* Añadir a la simulación generada por el algoritmo RACA el proceso de evaporación del agua almacenada en la imagen *DEM*, teniendo en cuenta las diferentes variables que influyen en dicho proceso, y utilizando tanto gráficas de precipitación como gráficas de temperaturas.
- *Línea 9.* Simular el proceso de erosión del terreno tras la caída de lluvia durante un largo intervalo de tiempo, para comprobar cómo modifica el paso del agua el aspecto del paisaje de la zona de estudio.
- *Línea 10.* Crear un sistema de posicionamiento óptimo automatizado del muro de una presa, con el objetivo de maximizar el caudal de agua acumulado tras el proceso de las precipitaciones.

### 5.3.3. Caracterización de texturas con el algoritmo TACA

Las líneas de investigación abiertas relacionadas con la caracterización de texturas de imágenes-AC y naturales mediante el algoritmo TACA son los siguientes:

- *Línea 11.* Mejorar la selección de primitivas más importantes del *HPH* y el *HPV*, con el objetivo de automatizar el proceso, y de este modo obtener unos resultados más optimizados.
- *Línea 12.* Extender la aplicación del algoritmo TACA a texturas naturales no binarias, para poder caracterizar texturas con un mayor grado de realismo.



---

# APÉNDICES

---



---

APÉNDICE A  
TRABAJO DE CAMPO

---



# Apéndice A

## TRABAJO DE CAMPO

### A.1. IMÁGENES *LANDSAT*

Este apartado incluye el trabajo de campo, realizado en el ámbito del proyecto *SOLERES* y avalado por expertos ecólogos [84], para las provincias de Almería y Granada. La *Figura A.1* muestra los sectores climáticos, con las zonas de estudio remarcadas.

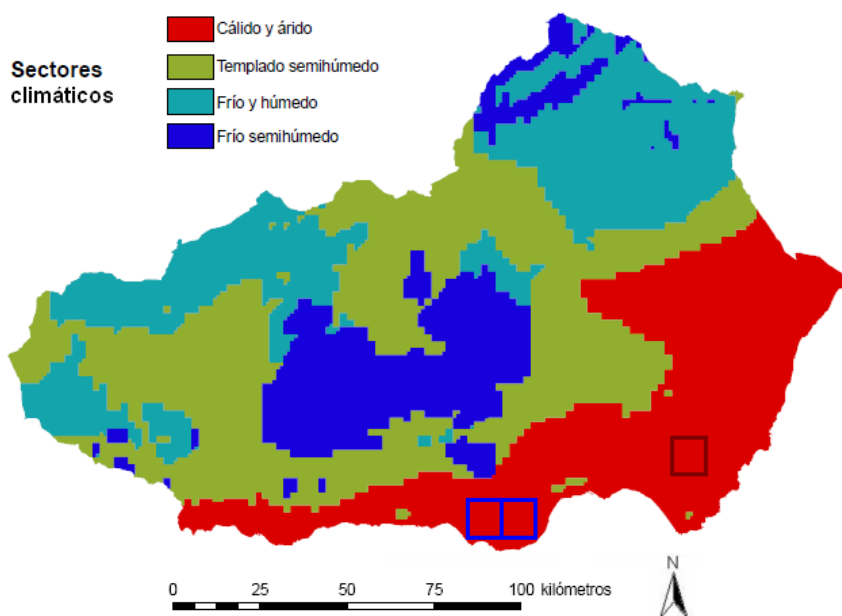


Figura A.1: Sectores climáticos. *Níjar* (cuadrado rojo) y *El Ejido* (cuadrados azules).

Como se puede observar, las zonas de estudio se caracterizan por la presencia exclusiva del clima cálido y árido. Por otro lado, la *edafología* es la parte de la ciencia que se encarga de estudiar la naturaleza y composición del suelo, en su relación con las plantas y el entorno que lo rodea. En lo que respecta a los sectores edáficos, las tres zonas de estudio se caracterizan por una mezcla de inceptisols, entisols y suelos esqueléticos. La *Figura A.2* muestra los sectores edáficos, con las zonas de estudio remarcadas.

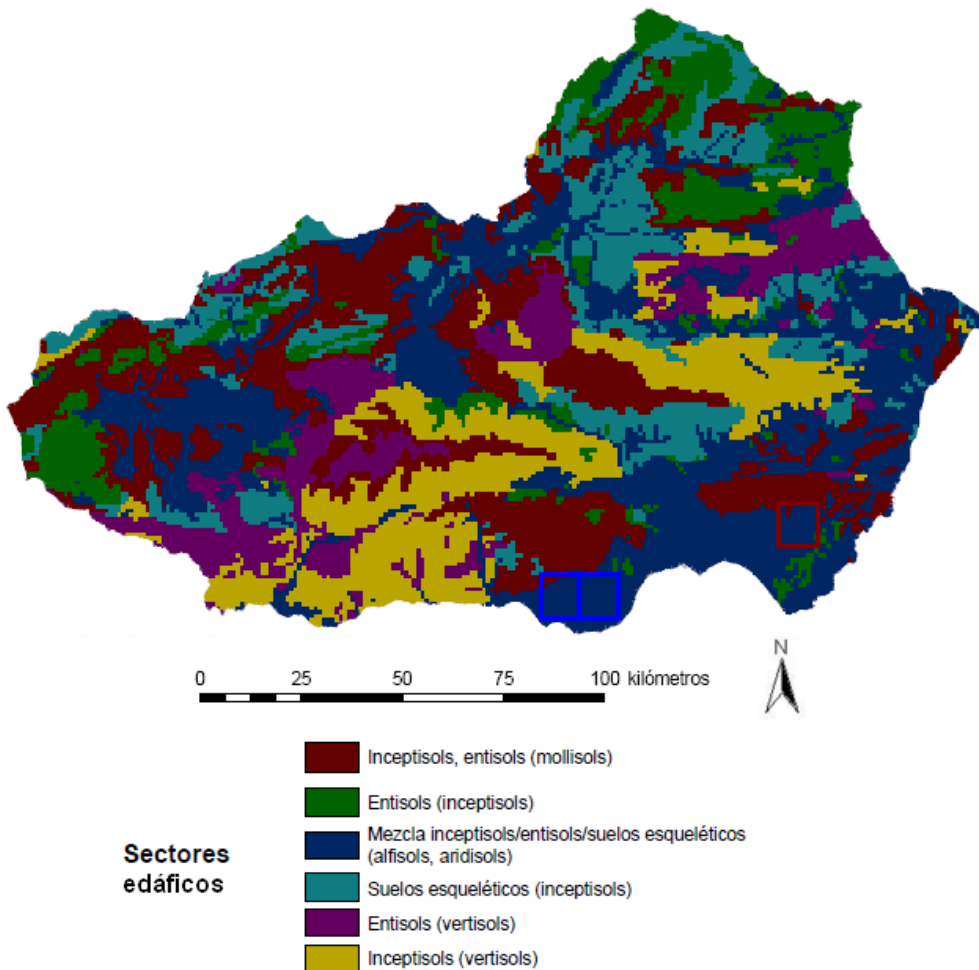


Figura A.2: Sectores edáficos. *Níjar* (cuadrado rojo) y *El Ejido* (cuadrados azules).



Por otro lado, la *litología* es la parte de la ciencia que se encarga del estudio de las rocas, analizando diversos parámetros, como la composición mineralógica, la distribución espacial, la textura, el tamaño de grano, el tamaño de las partículas y las características tanto físicas como químicas. En lo que respecta a los sectores litológicos, las tres zonas de estudio se caracterizan por una gran presencia de rocas sedimentarias, y en menor cantidad de rocas metamórficas y volcánicas. La *Figura A.3* muestra los sectores litológicos, con las zonas de estudio remarcadas.

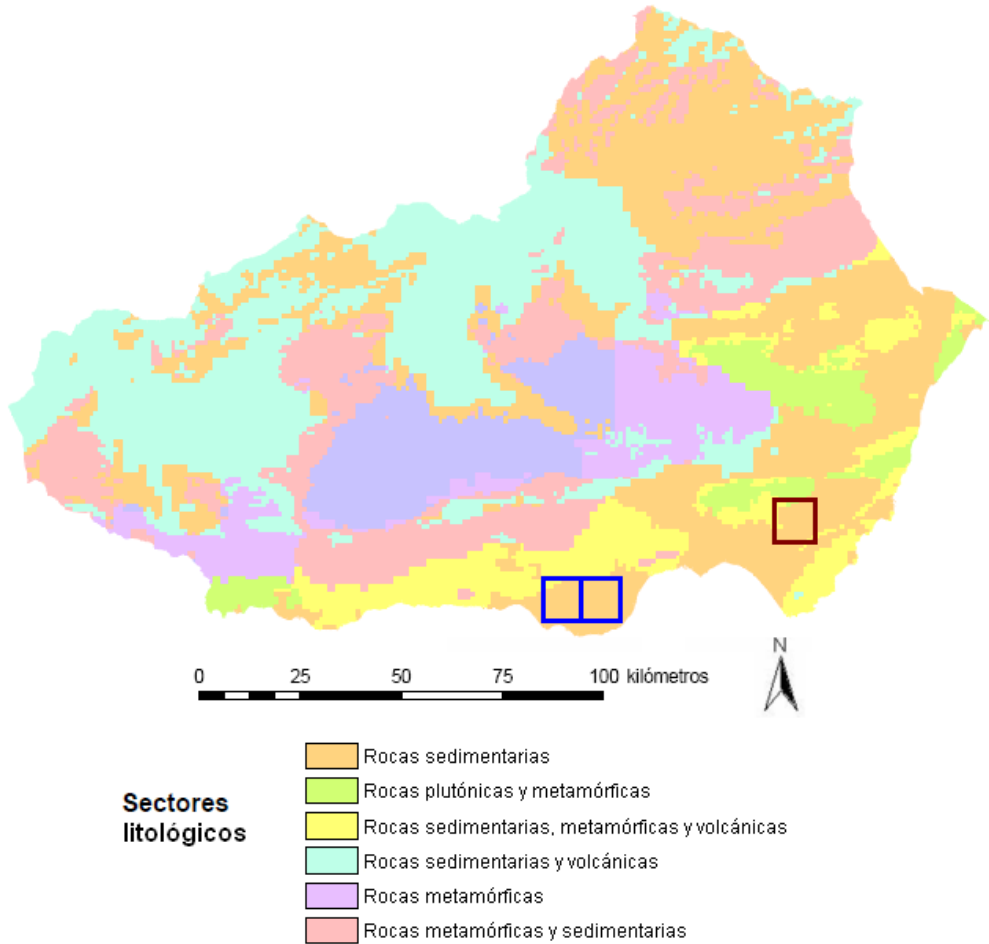


Figura A.3: Sectores litológicos. *Níjar* (cuadrado rojo) y *El Ejido* (cuadrados azules).

En lo que respecta a los sectores de vegetación, las tres zonas de estudio se caracterizan por una gran presencia de agricultura intensiva, sobre todo invernaderos, además de superficies construidas y alteradas. También se caracterizan, en menor cantidad, por la presencia de matorral disperso, roca y suelo desnudo. La *Figura A.4* muestra los sectores de vegetación, con las zonas de estudio remarcadas.

Por último, en lo que respecta a los sectores geomorfológicos, las tres zonas de estudio se caracterizan por una gran presencia de llanos y pendientes muy suaves. También se caracterizan, en menor cantidad, por la presencia de pendientes suaves y medias, y muy pocas pendientes escarpadas. La *Figura A.5* muestra los sectores geomorfológicos, con las zonas de estudio remarcadas.

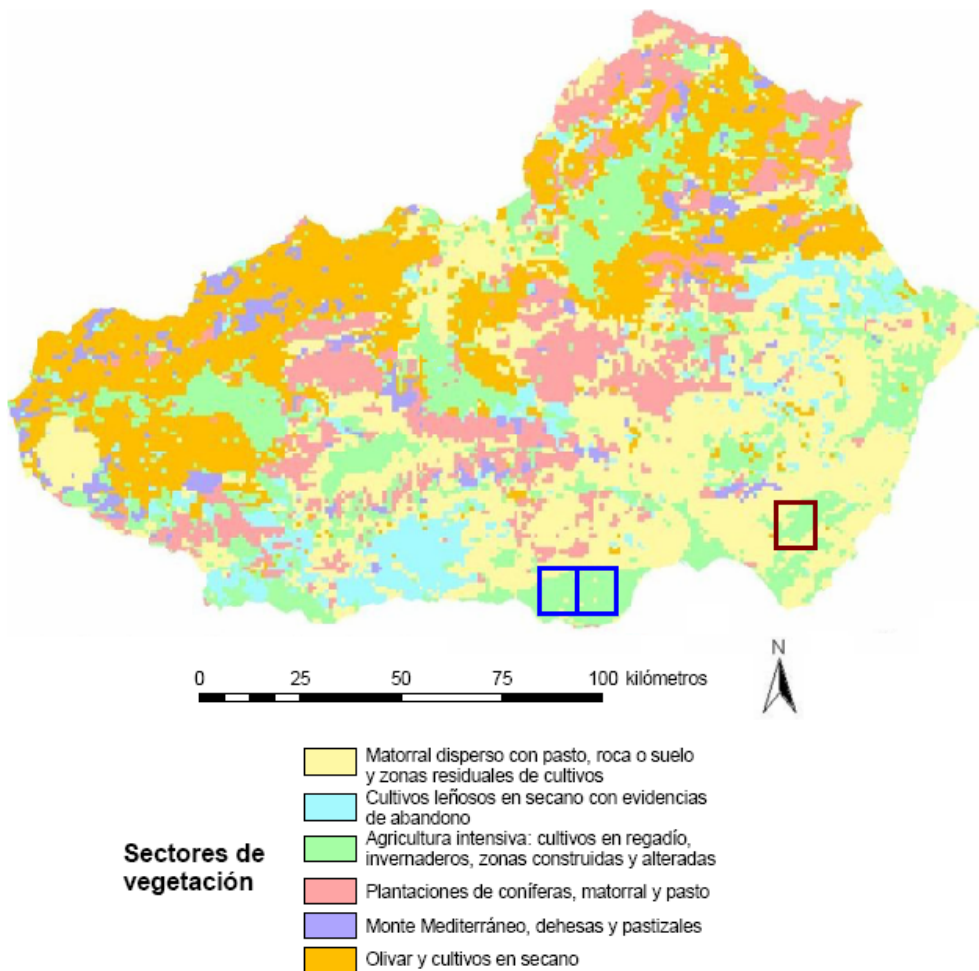


Figura A.4: Sectores de vegetación. *Níjar* (cuadrado rojo) y *El Ejido* (cuadrados azules).

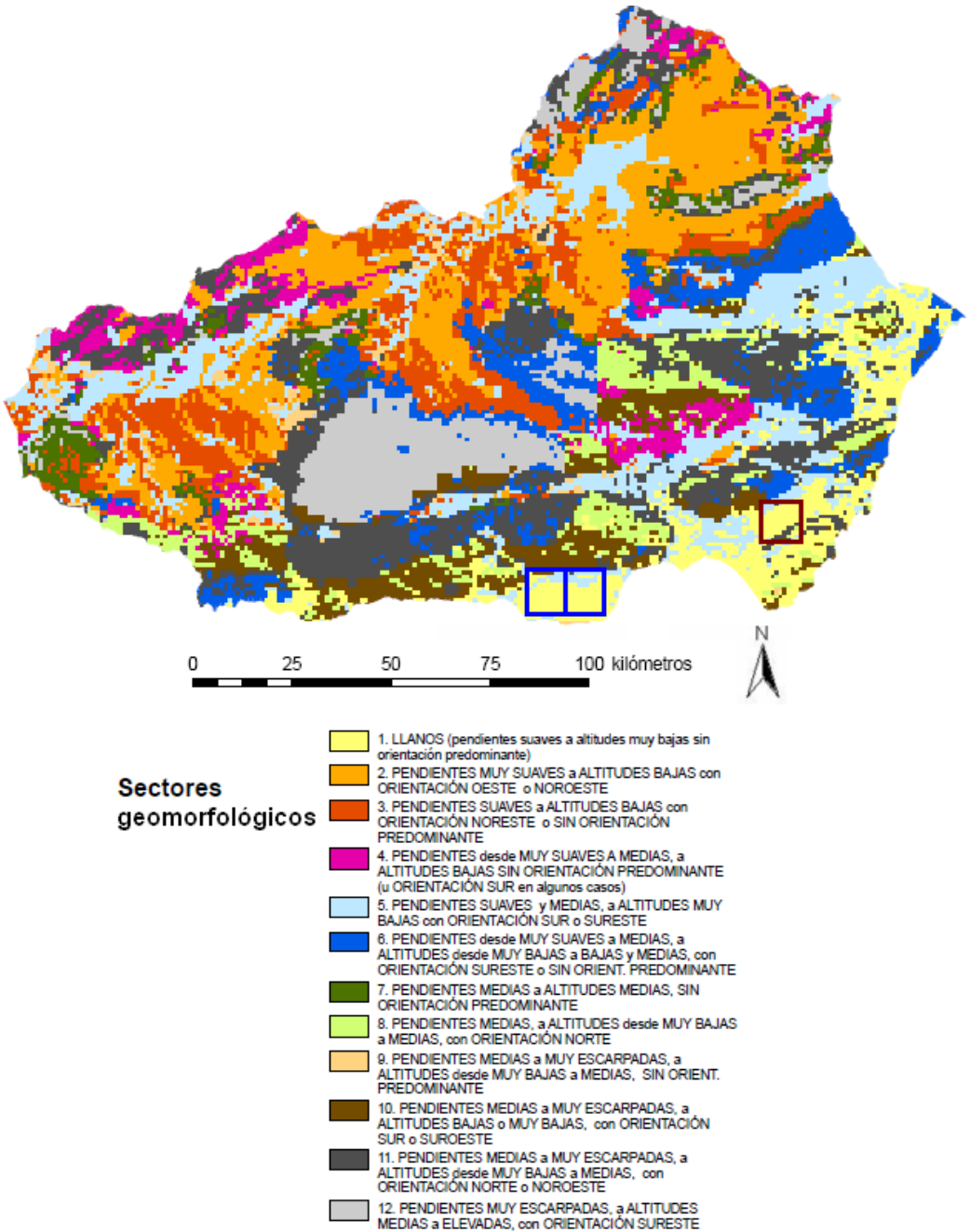


Figura A.5: Sectores geomorfológicos. *Níjar* (cuadrado rojo) y *El Ejido* (azules).

## A.2. IMAGEN DEM

Las imágenes de satélite *DEM* de la misión *SRTM* de la *NASA* han sido clasificadas por latitudes y longitudes de 1 grado por 1 grado, tomando siempre como referencia la distancia horizontal, es decir, el sentido de los paralelos. La dimensión espacial de dichas imágenes de satélite depende de la región geográfica de la superficie del planeta. La resolución espacial es de 1 seg-arco (equivalente a 30 metros) para las regiones de Estados Unidos, y de 3 seg-arco (equivalente a 90 metros) para todas las regiones del resto del planeta.

Por lo tanto, en el caso de la imagen *S32W070*, perteneciente a la región de Argentina, y utilizada para realizar los experimentos del algoritmo RACA, la resolución espacial de la imagen original es de 3 seg-arco (90m). Estos píxeles fueron remuestreados por la *NASA/USGS* para que cubrieran una región de 1 grado por 1 grado, que para la latitud de la zona de estudio ofrece una imagen de un tamaño aproximado de  $100 \times 100$  km, considerando un total de 600 píxeles. Con esos datos, cada píxel tiene una dimensión espacial de:  $100000\text{m}/600 \text{ píxeles} = 166,66 \text{ m/píxel}$ . La *Figura A.6* muestra la región geográfica que abarca la imagen *S32W070*, dentro del país de Argentina, con los nombres de otras regiones de alrededor, para poder tomarlas como referencia.

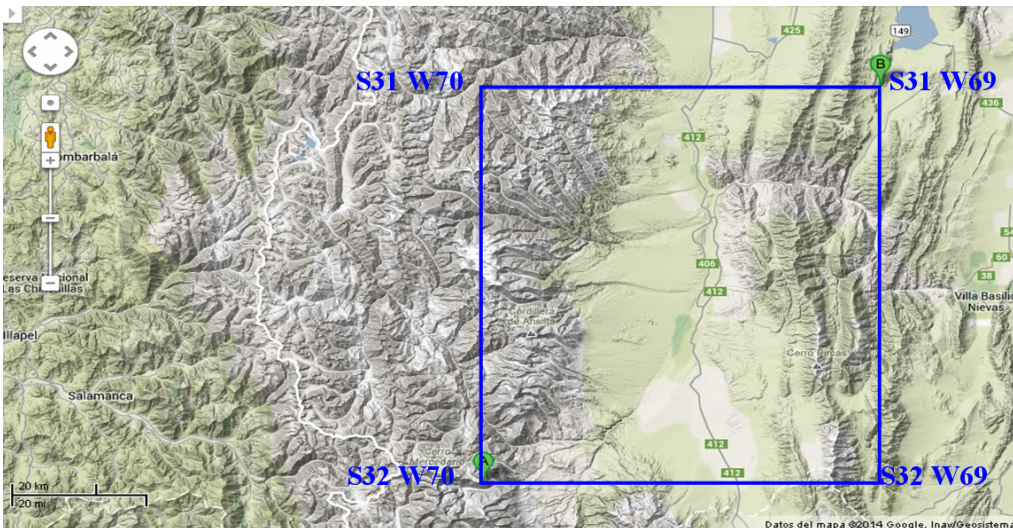


Figura A.6: Zona de estudio de la imagen *S32W070* (Fuente: *Google Maps*).

Dicha región geográfica corresponde al *Valle de Calingasta*, situado a una altitud de más de 1000 metros por encima del nivel del mar, y delimitado por las regiones montañosas del *Cordón de Ansilta* al oeste, y por la *Señora del Tontal* al este, situadas a más de 4000 metros de altitud. La *Figura A.7* muestra el perfil topográfico de la zona de estudio, tomando como dirección de oeste a este, y donde se ha remarcado en azul la región que abarca la imagen *S32W070*.

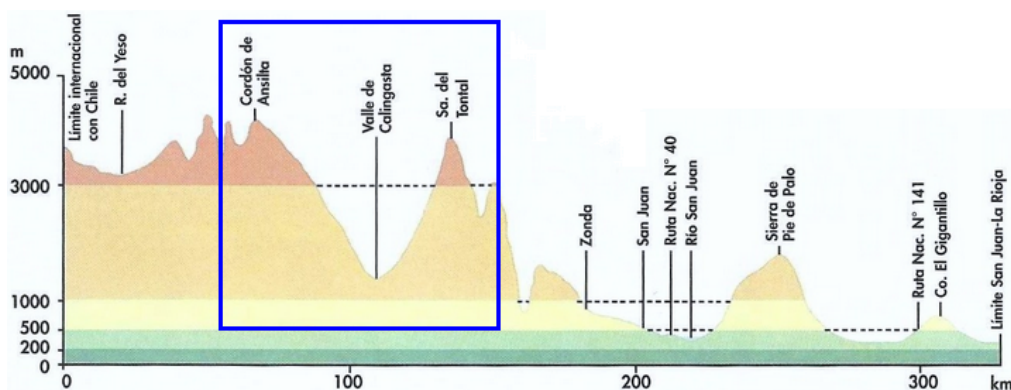


Figura A.7: Perfil topográfico con la zona de estudio de la imagen *S32W070* en azul (Fuente: *Atlas Geográfico de la República de Argentina - Instituto Geográfico Militar*).

Como consecuencia, debido al elevado tamaño de la imagen *S32W070*, su resolución espacial ha sido degradada, con el objetivo de que aumente a mayor velocidad el caudal de agua acumulado, utilizando las distintas gráficas de precipitaciones elegidas de la región de Argentina, y de este modo poder comparar mejor los diferentes tipos de climas utilizados para la investigación.



---

APÉNDICE B  
PUBLICACIONES ADICIONALES

---





# Apéndice B

## PUBLICACIONES ADICIONALES

### B.1. PROYECTOS I+D

Este apartado muestra una lista de los proyectos I+D más importantes, aparte de los proyectos *SOLERES* e *iSOLERES*, en los que también se ha participado:

- *Desarrollo de un agente Web inteligente de información medioambiental (TIC-6114)*, del grupo de investigación de Informática Aplicada (TIC-211) del departamento de Informática de la universidad de Almería.
- *MII SCHOOL. Comorbilidad psiquiátrica, acoso escolar y drogadicción en centros educativos: desarrollo de un soporte informático para su detección*, en colaboración con el grupo de investigación de Análisis Clínico y Experimental de los Trastornos del Espectro Esquizofrénico (HUM-760) del departamento de Personalidad, Evaluación y Tratamiento Psicológico de la universidad de Almería.
- *EVEMEH 1: estudio de un entorno virtual 3D para la evaluación de memoria espacial en humanos*, en colaboración con el grupo de investigación de Psicofarmacología, Neurotoxicología y Neuropsicología (CTS-280) del departamento de Neurociencia y Ciencias de la Salud de la Universidad de Almería.
- *EVEMEH 2: desarrollo de nuevos entornos virtuales 3D para la evaluación de memoria espacial en humanos*, en colaboración con el grupo de investigación de Psicofarmacología, Neurotoxicología y Neuropsicología (CTS-280) del departamento de Neurociencia y Ciencias de la Salud de la Universidad de Almería.
- *EVEMEH 3: un simulador de realidad virtual 3D para la evaluación de memoria espacial en humanos en nuevos escenarios*, en colaboración con el grupo de investigación de Psicofarmacología, Neurotoxicología y Neuropsicología (CTS-280) del departamento de Neurociencia y Ciencias de la Salud de la Universidad de Almería.

## B.2. PUBLICACIONES

Como se puede observar en la página anterior, a excepción del primer proyecto, el resto están relacionados con la aplicación de la computación gráfica al ámbito de la psicología, y tienen como objetivo realizar diversos estudios experimentales con sujetos humanos. Como vimos a lo largo del *Capítulo 5*, se han publicado un total de 9 trabajos de investigación directamente relacionados con el contenido de esta tesis doctoral. Aparte de dichos trabajos, también se han publicado otros 12 trabajos adicionales, relacionados con estos otros proyectos de investigación:

- *Artículo 1.* Carmona J.A., **Espínola M.**, Cangas A.J., Iribarne L., “MII-School: a 3d videogame for the early detection of abuse of substances, bullying, and mental disorders in adolescents”, *European Journal of Education and Psychology*, 4(1):75–85, ISSN: 1888-8992, 2011.
- *Artículo 2.* Carmona J.A., **Espínola M.**, Cangas A.J., Iribarne L., “Applying virtual reality (VR) to the detection and treatment of clinical problems in educational settings”, *Technology Enhanced Learning For People With Disabilities: Approaches And Applications*, London (England), in J. Zhao Ed., Information Science Publishing, pp. 194–202, doi: 10.4118/978-1-61520-92-1.ch014, 2011.
- *Artículo 3.* Carmona J.A., **Espínola M.**, Cangas A.J., Iribarne L., “Detecting drug use in adolescents using a 3D simulation program”, *Psychology, society and education*, 2(2):143–153, ISSN: 2171-2085, 2010.
- *Artículo 4.* Carmona J.A., **Espínola M.**, Cangas A.J., Iribarne L., “Mii school: new 3D technologies applied in education to detect drug abuses and bullying in adolescents”, *1st International Conference on Technology Enhanced Learning, Quality of Teaching and Educational Reform (TECH-EDUCATION 2010)*, May 19-21, Athens (Greece), 73, pp. 65–72, doi:10.1007/978-3-642-13166-0\_10, 2010.
- *Artículo 5.* **Espínola M.**, “Making of Mii School”, *Blender Art Magazine*, Issue 26: Blender & Gaming, pp. 30–35, 2010.
- *Artículo 6.* Cánovas R., **Espínola M.**, Iribarne L., Cimadevilla J.M., “A new virtual task to evaluate human place learning”, *Behavioural Brain Research Journal*, 190:112–118, doi:10.1016/j.bbr.2008.02.024, 2008. (**JCR Q1: 3.214**)
- *Artículo 7.* Cánovas R., Iribarne L., **Espínola M.**, León I., Roldán D., Cimadevilla J.M., “A new virtual task to evaluate human place learning”, *Spanish Psychiatry Society Joint Mid-Year Meeting*, July 4-7, Bilbao (Spain), pp. 43–43, 2007.
- *Artículo 8.* **Espínola M.**, Parra G., Iribarne L., Torres J.A., “Designing virtual tours of 3D-adaptative greenhouses”, *Ibero-American Symposium in Computer Graphics (SIACG 2006)*, July 5-7, Santiago de Compostela (Spain), pp. 28–31, 2006.

- *Libro 1.* Gómez J., **Espínola M.**, “Manual de redes inalámbricas”, *Sistemas de Oficina de Almería S.A. Ed.*, libro docente de 200 páginas, ISBN: 84-96270-97-8, 2007.
- *Libro 2.* Iribarne L., **Espínola M.**, Berenguel J.L., Márquez A., “Programación de animaciones en 3D: un motor gráfico en OpenGL”, *Sistemas de Oficina de Almería S.A. Ed.*, libro docente de 200 páginas, ISBN: 84-96270-89-3, 2007.
- *Libro 3.* Iribarne L., **Espínola M.**, Parra G., “Iniciación al diseño y programación de videojuegos 3D”, *Sistemas de Oficina de Almería S.A. Ed.*, libro docente de 150 páginas, ISBN: 84-96270-69-6, 2006.
- *Libro 4.* Iribarne L., **Espínola M.**, “Análisis y diseño del software. Teoría y práctica”, *Sistemas de Oficina de Almería S.A. Ed.*, libro docente de 442 páginas, ISBN: 84-96270-53-X, 2005.

Como se puede observar, hay un total de 8 publicaciones adicionales relacionadas con el ámbito de la investigación, donde 3 de ellas han sido publicadas en revistas científicas de impacto. Además, hay otras 4 publicaciones adicionales, relacionadas con el ámbito de la docencia.



---

GLOSARIO  
DE TÉRMINOS

---



# GLOSARIO DE TÉRMINOS

## **Algoritmo ACA**

Algoritmo de clasificación de imágenes de satélite supervisado basado en autómatas celulares (*classification Algorithm based on Cellular Automata*). El algoritmo ACA, implementado durante el trabajo de investigación desarrollado para esta tesis doctoral, utiliza las distintas propiedades de los autómatas celulares para mejorar el proceso de clasificación de las imágenes de satélite en tres niveles importantes: acierto, flexibilidad y personalización, ya que por un lado obtiene una mejor tasa de acierto durante el proceso de clasificación, mezclando técnicas espectrales-contextuales a través de la vecindad del autómata celular, por otro lado ofrece unos resultados jerárquicos divididos en niveles de fiabilidad, que representan el grado de pertenencia de cada píxel a su clase correspondiente a través de las iteraciones, y finalmente permite personalizar el proceso de clasificación, a través de las reglas del autómata celular.

## **Algoritmo ACA contextual**

Parte del algoritmo ACA que utiliza la vecindad del autómata celular para aplicar una clasificación contextual, así como un conjunto de estados y reglas para personalizar los resultados obtenidos durante proceso de clasificación. El algoritmo ACA contextual se basa en los resultados obtenidos mediante el algoritmo ACA espectral, que son mejorados mediante técnicas contextuales, y personalizados mediante el uso de reglas y estados del autómata celular. En esta parte del algoritmo se consiguen dos de los tres objetivos planteados para el algoritmo ACA. Por un lado, se consigue mejorar la tasa de acierto de la clasificación obtenida mediante técnicas contextuales (*objetivo #1*), concretamente aplicando las reglas #1 y #4 (para píxeles ruidosos e inciertos). Por otro lado, también se consigue definir el tipo de píxel: incierto, ruidoso, borde y foco, aplicando las 4 reglas del autómata celular (*objetivo #3*).

## **Algoritmo ACA espectral**

Parte del algoritmo ACA basado en algoritmos clásicos supervisados mejorados mediante las técnicas de los autómatas celulares, gracias a la división del proceso de clasificación en varias iteraciones, lo que provoca que se obtenga una clasificación jerárquica dividida en distintas capas con un nivel de fiabilidad cada una, debido a que se va incrementando el *threshold* a las clases en cada iteración. Es decir, los píxeles clasificados en las primeras iteraciones son más fiables que los clasificados en las siguientes iteraciones. En esta parte del algoritmo, por lo tanto, se consigue el *objetivo #2*.

**Algoritmo ACA principal**

Parte del algoritmo ACA que ejecuta las iteraciones del autómata celular. En cada iteración, primero se realiza una clasificación espectral (algoritmo ACA espectral) de todos los píxeles que aún no han sido clasificados en la imagen, y posteriormente una clasificación contextual (algoritmo ACA contextual) de aquellos que se encuentran en el *threshold* permitido.

**Algoritmo RACA**

Algoritmo de simulación de precipitaciones sobre imágenes de satélite *DEM* tridimensionales basado en autómatas celulares (*Rainfall Algorithm with Cellular Automata*). El algoritmo RACA, implementado en *MATLAB*, utiliza las distintas propiedades de los autómatas celulares para realizar una simulación del caudal de agua acumulado en una región geográfica concreta, tras aplicar una o varias gráficas de precipitaciones.

**Algoritmo TACA**

Algoritmo de caracterización de texturas de imágenes-AC y naturales basado en autómatas celulares (*Texture Algorithm with Cellular Automata*). El algoritmo TACA, implementado en *MATLAB*, utiliza un proceso de ingeniería inversa para obtener las reglas de *Stephen Wolfram* que definen las texturas.

**Analista experto**

Persona especialista en el ámbito de la teledetección, que se encarga de desarrollar alguna tarea relacionada con la clasificación de imágenes de satélite, ya sea en alguna fase previa (elaboración del conjunto de entrenamiento para la clasificación supervisada) o posterior (verificación de la tasa de acierto a través del trabajo de campo y análisis de los resultados obtenidos durante el proceso de clasificación).

**Autómata celular**

Modelo matemático que consiste en una malla de celdas, distribuidas normalmente en forma de matriz, que tiene las siguientes características básicas: estados, reglas, función de transición, vecindad e iteraciones.

**Banda espectral**

Franja del espectro electromagnético, almacenada en una imagen de satélite, para cada uno de los píxeles. Existen imágenes de satélite con un número reducido de bandas (imágenes multiespectrales), y otras con un número elevado (imágenes hiperespectrales).

**Binarización**

Transformación que consiste en asignar los valores 0 y 1 a la imagen de salida, para todos los píxeles de la imagen original cuyos niveles digitales sean superiores o inferiores a un determinado umbral establecido de antemano.

**Clase espectral**

Cada una de las categorías en las que etiquetan los píxeles de una imagen de satélite durante el proceso de clasificación.



**Clasificación contextual**

Algoritmo que se aplica, normalmente después del propio proceso de clasificación, para refinar los píxeles más difíciles de clasificar, teniendo en cuenta sus píxeles vecinos.

**Clasificación de imágenes de satélite**

Proceso que consiste en agrupar los píxeles de una imagen de satélite en un número finito de clases, basándose en los valores espectrales de las distintas bandas, convirtiendo de este modo la información captada por los sensores del satélite como niveles digitales a una escala categórica fácil de interpretar por los analistas expertos.

**Clasificación no supervisada**

Tipo de clasificación en la que el analista experto sólo especifica al algoritmo el número de clases en las que desea dividir la imagen, y el propio algoritmo se encarga de agrupar píxeles similares, basándose únicamente en la información espectral almacenada en las bandas de cada píxel.

**Clasificación pseudodifusa**

Clasificación ACA jerárquica, dividida en niveles de fiabilidad basados en la proximidad espectral dentro del espacio de características, de modo que en cada iteración del autómata celular sólo son clasificados aquellos píxeles de la imagen que están a una distancia máxima del centro de su clase correspondiente, y dicha distancia va aumentando en cada iteración. Por lo tanto, los píxeles clasificados en una iteración particular son más fiables que los que se clasifican en una iteración posterior, y así sucesivamente.

**Clasificación supervisada**

Tipo de clasificación en la que el analista experto selecciona muestras de píxeles de cada clase (conjunto de entrenamiento), para que el algoritmo pueda realizar una clasificación más precisa, basada en el conocimiento humano experto.

**Combinación de bandas**

Elección de determinadas bandas de una imagen de satélite, que permite observar determinadas características de un modo más optimizado. Así, por ejemplo, la combinación 3,2,1 (color verdadero) es ideal para realizar estudios del agua, mientras que la combinación 4,3,2 es ideal para observar variaciones en la vegetación.

**Conjunto de entrenamiento**

Muestras de píxeles representativas tomadas para cada clase espectral por parte del analista experto, con el objetivo de usarlas posteriormente en una clasificación supervisada basada en el conocimiento experto humano, y de este modo obtener mejores resultados durante el proceso de clasificación.

**Digital Elevation Model (DEM)**

Imagen de satélite que almacena, para cada píxel, la altura a la que se encuentra la superficie terrestre en ese punto determinado de la imagen, ofreciendo la posibilidad de realizar representaciones tridimensionales de la misma.

**Erdas Imagine**

Herramienta software relacionada con el ámbito de la teledetección, que proporciona las técnicas más avanzadas de análisis de imágenes de satélite y modelado espacial para la generación de nueva información.

**Error atmosférico**

Error en la captación de imágenes de satélite, provocado por la interacción que se produce entre la radiación electromagnética y la atmósfera.

**Error geométrico**

Error en la captación de imágenes de satélite, que puede estar provocado por diferentes motivos: por la propia plataforma satelital, por los sensores instalados en ella o por el fenómeno de rotación terrestre.

**Error radiométrico**

Error en la captación de imágenes de satélite, provocado por la descalibración de algún detector que los sensores poseen por cada una de las bandas electromagnéticas que captan, lo que puede producir efectos de bandeo en la imagen obtenida.

**Espectro electromagnético**

Distribución energética de las longitudes de onda que puede adoptar la radiación electromagnética. Se extiende desde las ondas electromagnéticas de menor longitud, como los rayos gamma y los rayos X, pasando por la luz ultravioleta, luz visible y rayos infrarrojos, hasta las radiaciones de mayor longitud de onda, como las ondas de radio.

**Estado**

Valor alfanumérico que se asigna a cada celda del autómata celular. Existe un conjunto finito de estados posibles, que van cambiando mediante la aplicación de las reglas de la función de transición, teniendo en cuenta la vecindad de cada celda.

**Estado #1 (ACA)**

El primer estado es (*clase*), y corresponde a la clase en la que es clasificado cada píxel de la imagen utilizando no sólo sus valores espectrales, sino también información contextual. Este estado nos permite mejorar el acierto de la clasificación (*objetivo #1*).

**Estado #2 (ACA)**

El segundo estado es (*calidad*), y corresponde al número de iteración del autómata celular en el que es clasificado cada píxel de la imagen. Este estado nos permite obtener la clasificación jerárquica basada en niveles de fiabilidad (*objetivo #2*).

**Estado #3 (ACA)**

El tercer estado es (*tipo*), y corresponde al tipo de píxel con el que estamos tratando: incierto, ruidoso, borde o foco (los que no son bordes espaciales de las clases). Este estado nos permite obtener una lista detallada de los píxeles inciertos, ruidosos y bordes de clases (*objetivo #3*).

**Estándar**

Documento disponible públicamente que define las especificaciones para las interfaces, servicios, procesos, protocolos o formatos de datos, y que es establecido y mantenido por consenso por un grupo. Ejemplo de estándar son los que emite la organización *ISO*.

**Fase de asignación**

Agrupación de los píxeles de la imagen de satélite en las clases espectrales definidas.

**Fase de entrenamiento**

Definición de las clases en las que se van a agrupar los píxeles de la imagen de satélite en una clasificación supervisada. Para ello se toman píxeles de muestra de cada una de las clases, para obtener una clasificación basada en el conocimiento experto humano.

**Firma espectral**

Manera de reflejar o emitir energía de un determinado objeto o cubierta. Depende de las características físicas o químicas del objeto que interactúa con la energía electromagnética, y varía según las longitudes de onda.

**Fotografía aérea**

Sistema de percepción remota, desde un avión, que utiliza la reflexión natural del sol.

**Fractal**

Objeto geométrico cuya estructura básica, fragmentada o irregular, se repite infinitamente a diferentes escalas.

**Función de transición**

Función del autómata celular que utiliza las reglas para cambiar los estados de cada celda, a lo largo de las distintas iteraciones, teniendo en cuenta la vecindad.

**Georreferenciación**

Procedimiento mediante el cual se dota de validez cartográfica a una imagen de satélite, corrigiendo geoméricamente la posición de los píxeles, y atribuyéndoles coordenadas según un sistema de referencia.

**Grado de heterogeneidad**

Característica de una clase espectral que indica si sus píxeles presentan un abanico espectral reducido o amplio. De este modo, en una clase con un bajo grado de heterogeneidad, los píxeles presentan unas características espectrales similares entre sí, por lo que normalmente son clases más fáciles de clasificar. Por otro lado, en una clase con un alto grado de heterogeneidad, los píxeles presentan unas características espectrales algo distintas entre sí, por lo que abarcan un mayor abanico espectral amplio, y como consecuencia son clases más difíciles de clasificar mediante técnicas espectrales.

**Histograma**

Gráfica de las frecuencias de los valores digitales.

**Imagen-AC**

Representación matricial en forma de imagen de las distintas generaciones producidas por un autómata celular unidimensional. Cada generación del autómata celular unidimensional está representada por un renglón de la matriz, donde la primera generación corresponde con el primer renglón, y las siguientes se ubican de manera correlativa. Según la notación de *Stephen Wolfram*, podemos aplicar 256 reglas básicas utilizando autómatas celulares unidimensionales, y de cada una de las reglas disponibles surge una imagen-AC distinta.

**Imagen de satélite**

Representación gráfica de la medición de la energía electromagnética registrada por los sensores de un satélite, y no por medios fotográficos convencionales. Así, mientras las fotografías normales se toman dentro de un cierto rango del espectro electromagnético (luz visible), los satélites amplían este rango para captar muchísima más información, que posteriormente podrá ser analizada.

**Imagen hiperespectral**

Imagen de satélite con un número elevado de bandas, en algunos casos cientos de ellas.

**Imagen multiespectral**

Imagen de satélite con un número reducido de bandas.

**Índice normalizado de vegetación**

Indicador de la actividad fotosintética en una imagen de satélite.

**Isodata**

Algoritmo de clasificación no supervisado que permite, no sólo especificar un número máximo de iteraciones para realizar el proceso de *clustering* o agrupamiento de los píxeles en clases, sino además establecer un umbral de convergencia que determine el porcentaje de píxeles mínimo que se debe modificar entre dos iteraciones para continuar con el algoritmo.

**Iteración**

Fase actual en la que se encuentra la aplicación del autómata celular.

**Juego de la vida**

Simulación del proceso de vida basada en autómatas celulares bidimensionales creada por el matemático *John Horton Conway*, donde cada celda tiene la posibilidad de nacer, reproducirse, mantenerse con vida o morir.

**K-medias**

Algoritmo de clasificación no supervisado que tiene como objetivo realizar una partición de un conjunto de elementos de entrada (píxeles) en un número de *clusters* (clases) determinado de antemano, de tal modo que cada *cluster* incorpora los elementos con la media más cercana.

**Litología**

Parte de la ciencia que se encarga del estudio de las rocas, analizando diversos parámetros, como la composición mineralógica, la distribución espacial, la textura, el tamaño de grano, el tamaño de las partículas y las características físicas-químicas.

**Mathematica**

Herramienta de software matemático que se utiliza en áreas científicas, de ingeniería, matemáticas y áreas computacionales. Fue creado originalmente por *Stephen Wolfram*, quien continúa siendo el líder del grupo de matemáticos y programadores que desarrollan el producto en *Wolfram Research*.

**MATLAB**

Herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio, el lenguaje M, de tal modo que el usuario puede programar sus propios *scripts* de código fuente. Entre sus prestaciones básicas destacan la manipulación de matrices, la representación de funciones, la implementación de algoritmos, la creación de interfaces de usuario y la comunicación con otros lenguajes.

**Matriz de confusión**

Tabla de contingencia donde se muestran los resultados del muestreo de control de la calidad de un proceso de clasificación, comparando los resultados de la clasificación con los valores reales obtenidos mediante trabajo de campo.

**Máxima verosimilitud**

Algoritmo de clasificación supervisado que considera que los niveles digitales de cada clase se ajustan a una distribución normal. Esto nos permite describir esa clase mediante una función de probabilidad, a partir de su vector de medias y matriz de covarianza. Así se puede calcular la probabilidad de que un determinado píxel pertenezca a una clase concreta.

**Mínima distancia**

Algoritmo de clasificación supervisado que asigna a cada píxel la clase con la que existe una menor distancia espectral respecto a su centroide, teniendo en cuenta todas las bandas que intervienen en el análisis. Para calcular la distancia entre el píxel y las clases se utiliza la fórmula de la *distancia euclídea*.

**Nivel digital**

Valor numérico (normalmente entre 0 y 255) que se asigna a cada una de las bandas de cada posición espacial de una imagen de satélite.

**Objetivo #1 (ACA)**

*Mejorar la tasa de acierto de la clasificación.* El algoritmo ACA debe clasificar los píxeles problemáticos, teniendo en cuenta no sólo sus datos espectrales (ambiguos para los píxeles inciertos, erróneos para los píxeles ruidosos), sino también sus datos contextuales vecinos, por lo que se mejora la tasa de acierto final.

**Objetivo #2 (ACA)**

*Construir una clasificación jerárquica.* El algoritmo ACA debe clasificar, en cada iteración, sólo aquellos píxeles que están dentro de una distancia espectral máxima en el espacio de características con respecto al centro de su clase correspondiente, y tal distancia debe aumentar en la siguiente iteración. El algoritmo ACA ofrecerá una clasificación jerárquica dividida en capas de fiabilidad, donde las primeras capas ofrecen más fiabilidad que las últimas, y estos resultados pueden ser muy útiles para la posterior interpretación de los resultados por parte de los analistas expertos. Por otra parte, este objetivo ayuda al primero, ya que se utiliza como vecindad píxeles más fiables clasificados en iteraciones anteriores.

**Objetivo #3 (ACA)**

*Detectar bordes y píxeles inciertos/ruidosos.* El algoritmo ACA debe personalizar el proceso de clasificación de la imagen de satélite mediante la inclusión de reglas, como por ejemplo para poder obtener una lista detallada de los píxeles inciertos y ruidosos, lo cual puede ser útil si el proceso de clasificación falla incluso cuando se utilizan técnicas contextuales, y obtener una lista de los píxeles que determinan los bordes espaciales de las clases de la imagen.

**Ortorectificación**

Corrección geométrica de una imagen derivada de una perspectiva convencional de la imagen por rectificación diferencial o simple, para que los desplazamientos causados por la inclinación del sensor y el relieve del terreno sean removidos.

**Pancromática**

Imagen de una sola banda que cubre gran parte del espectro visual (p.e., imagen convencional en blanco y negro).

**Paralelepípedos**

Algoritmo de clasificación supervisado en el que el usuario establece un área de dominio para cada clase espectral, teniendo en cuenta sus valores de centralidad y dispersión. Posteriormente, cada píxel es asignado a una clase si sus niveles digitales están dentro del área de dominio de dicha clase, para todas las bandas que intervienen en el proceso de clasificación.

**Píxel**

Cada uno de los puntos que componen una imagen. En el caso de las imágenes de satélite, cada píxel puede tener un número reducido de valores espectrales (imágenes multispectrales) o un número elevado (imágenes hiperespectrales).

**Píxel borde**

Tipo de píxel que es clasificado por el algoritmo ACA como frontera o borde espacial. Para ello el algoritmo ACA contextual tiene en cuenta la vecindad de cada píxel, de tal modo que aquellos píxeles cuyos vecinos tengan alguna clase distinta a la suya, son bordes espaciales de sus clases correspondientes.

**Píxel foco**

Tipo de píxel que es clasificado por el algoritmo ACA como lo contrario de frontera o borde espacial. Para ello el algoritmo ACA contextual tiene en cuenta la vecindad de cada píxel, de tal modo que aquellos píxeles cuyos vecinos tengan todas sus clases iguales a la suya, son focos espaciales de sus clases correspondientes.

**Píxel incierto**

Tipo de píxel que es clasificado por el algoritmo ACA como dudoso a nivel espectral, es decir, que por las características de sus niveles digitales, podría pertenecer a 2 ó más clases. Para clasificar correctamente este tipo de píxeles, el algoritmo ACA contextual tiene en cuenta su vecindad, y le asigna la clase mayoría de sus vecinos de entre las clases dudosas.

**Píxel ruidoso**

Tipo de píxel que es clasificado por el algoritmo ACA como defectuoso, es decir, que presenta algún tipo de error en las características de sus niveles digitales. Para clasificar correctamente este tipo de píxeles, el algoritmo ACA contextual tiene en cuenta su vecindad, y le asigna la clase mayoría de sus vecinos.

**Precisión**

Valor que indica el acierto determinado de que una clase específica ha sido predicha correctamente.

**Raster**

Formato de representación de imágenes de satélite, en el que la disposición de los píxeles está en forma de matriz numérica de niveles digitales.

**Recall**

Valor que indica la capacidad del modelo de predicción para seleccionar una clase concreta del conjunto de entrenamiento.

**Regla**

Norma que se aplica en cada iteración del autómata celular a través de la función de transición, para cambiar el estado de cada celda teniendo en cuenta no sólo su estado, sino también su vecindad.

**Resolución espacial**

Superficie terrestre que almacena cada píxel de la imagen. También se suele usar el concepto de IFOV, o campo instantáneo de visión, que se define como la sección angular en radianes observada en un momento determinado.

**Resolución espectral**

Número y anchura de las bandas electromagnéticas captadas por el sensor del satélite. A mayor número de bandas, dispondremos de mayor número de variables que puedan describir cada píxel de la imagen.

**Resolución radiométrica**

Sensibilidad para discriminar entre pequeñas variaciones en la radiación captada por el sensor del satélite. Esta resolución se suele expresar como el número de bits necesarios para almacenar cada píxel, lo que nos ofrece el número de niveles digitales del sensor.

**Resolución temporal**

Intervalo de tiempo que transcurre entre cada imagen obtenida por el sensor del satélite en la misma zona de la superficie terrestre. Esta resolución puede ser la deseada en el caso de los aviones, cada media hora en el caso de los satélites geosíncronos y variable en el caso de los satélites heliosíncronos.

**Ruido**

Error, normalmente aleatorio, introducido en la imagen de satélite durante el proceso de adquisición, transmisión o grabación, produciendo alteraciones en los niveles digitales.

**Satélite artificial**

Artefacto de creación humana enviado al espacio mediante un vehículo de lanzamiento con el objetivo de captar información de la superficie de la Tierra u otro planeta, a través de una serie de sensores que recogen la radiación electromagnética emitida por dicha superficie.

**Sensor activo**

Tipo de sensor que provee su propia fuente de energía que emite hacia los cuerpos, para posteriormente recibir la señal de retorno (p.e., radar).

**Sensor pasivo**

Tipo de sensor que recibe la señal de una fuente de luz o energía externa reflejada sobre la superficie de la tierra (p.e., espectro visible).

**Sistema complejo**

Sistema compuesto por varias partes interconectadas cuyos vínculos crean información adicional no visible antes para el observador. Como consecuencia, surgen nuevas propiedades, que no pueden explicarse a partir de los elementos aislados.

**Sistema de Información Geográfico (SIG)**

Herramienta, normalmente basada en el uso de la teledetección, que tiene como objetivo solucionar problemas de diversa índole, como por ejemplo el cálculo del crecimiento del suelo urbano en las ciudades en un intervalo de tiempo determinado, la monitorización de la calidad ambiental después de desastres naturales o la creación de mapas *GPS*.

**Teledetección**

Observación de la superficie terrestre desde el espacio, utilizando las propiedades de las ondas electromagnéticas emitidas, reflejadas o difractadas por los objetos observados, para fines de mejoramiento de la ordenación de los recursos naturales, de utilización de tierras y de protección del medio ambiente.



**Textura**

Frecuencia de cambio y disposición que presentan los niveles de gris en una imagen.

**Tipos de ACs**

Clasificación de los autómatas celulares unidimensionales, teniendo en cuenta su comportamiento dinámico: predecibles (*tipo 1*), periódicos (*tipo 2*), caótico (*tipo 3*) e imprevisible (*tipo 4*).

**Trabajo de campo**

Estudio realizado, normalmente por expertos medioambientales o ecólogos, sobre las características de una región geográfica concreta. Para el caso concreto de la clasificación de imágenes de satélite, el trabajo de campo se suele utilizar para verificar los resultados obtenidos mediante un algoritmo de clasificación.

**Vecindad de Moore**

Vecindad que toma 8 vecinos para cada celda del autómata celular.

**Vecindad de Moore extendida**

Vecindad que toma 24 vecinos para cada celda del autómata celular.

**Vecindad de von Neumann**

Vecindad que toma 4 vecinos para cada celda del autómata celular.

**Verificación**

Comprobación de los resultados obtenidos después de un proceso de clasificación de una imagen de satélite. Para ello, normalmente se aplica la matriz de confusión entre la imagen clasificada y otra imagen obtenida mediante trabajo de campo.

**Weka**

Herramienta software que se utiliza para el aprendizaje automático y la minería de datos. Incorpora una gran cantidad de algoritmos de clasificación, preprocesamiento y análisis de datos, *clustering*, regresión y modelado predictivo, así como aplicaciones de visualización de datos y resultados. La herramienta *Weka* está desarrollada en la *Universidad de Waikato*, está implementada en *Java*, por lo que es multiplataforma, y se distribuye como software libre bajo licencia *GNU-GPL*.



---

# ACRÓNIMOS

---



# ACRÓNIMOS

AC	Autómata Celular
ACA	classification Algorithm based on Cellular Automata
ACA MD	ACA Mínima Distancia
ACA P	ACA Paralelepípedos
API	Application Programming Interface
AVHRR	Advanced Very High Resolution Radiometer
C1, ..., C6	Clases en las que se etiquetan los píxeles
CA	Cellular Automata
COPUOS	Committee On the Peaceful Uses of Outer Space
D3, ..., D85	Imágenes del Álbum Brodatz
DEM	Digital Elevation Model
DTM	Digital Terrain Model
ERDAS	Earth Resources Data Analysis System
ERS	European Remote sensing Satellite
ESA	European Space Agency
ETM	Enhanced Thematic Mapper
EUMETSAT	EUropean organisation for the exploitation of METeorological SATellites
GIS	Geographic Information System
GPS	Global Positioning System
HP	Histograma de Primitivas
HPH	Histograma de Primitivas Horizontal
HPV	Histograma de Primitivas Vertical

---

IFOV	Instantaneous Field Of Vision
IMG	IMaGe data
IRS	Indian Remote sensing Satellite
ISODATA	Iterative Self-Organizing Data Analysis Technique Algorithm
ISS	International Space Station
LANDSAT	LAND SATellite
LASER	Light Artificially Stimulated Electromagnetic Radiation
MATLAB	MATrix LABoratory
MRF	Markov Random Fields
NASA	National Aeronautics and Space Administration
ND	Nivel Digital
NOAA	National Oceanic and Atmospheric Administration
R1, ..., R255	Reglas de Stephen Wolfram
RACA	Rainfall Algorithm with Cellular Automata
RADAR	RAdio Detection And Ranging
RBF	Radial Basis Function
RGB	Red Green Blue
S32W070	South 32° West 070°
SIG	Sistema de Información Geográfico
SPOT	Systeme Pour l'Observation de la Terre
SRTM	Shuttle Radar Topography Mission
TACA	Texture Algorithm with Cellular Automata
TIROS	Television Infra-Red Observation Satellite
TM	Thematic Mapper
USGS	United States Geological Survey
WEKA	Waikato Environment for Knowledge Analysis

---

# BIBLIOGRAFÍA

---





# Bibliografía

- [1] Agüera F., Aguilar M.A., Aguilar F.J., “Detecting greenhouse changes from Quick-Bird imagery on the Mediterranean coast”, *International Journal of Remote Sensing*, 27(21):4751–4767, 2006.
- [2] Aha D., Kibler D., Albert M.K., “Instance-based learning algorithms”, *Machine Learning*, 6(1):37–66, 1991.
- [3] Aponte A., Moreno J.A., “Cellular automata and its application to the modeling of vehicular traffic in the city of Caracas”, *7th International Conference on Cellular Automata for Research and Industry (ACRI 2006)*, September 20-23, Perpignan (France), in S. El Yacoubi *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 4173, pp. 502–511, 2006.
- [4] Asamblea General de las Naciones Unidas, “Principios relativos a la teleobservación de la Tierra desde el espacio”, *Resolución 41/65 de 1986*, principio I, pag. 120, 1986.
- [5] Avolio M.V., Errera A., Lupiano V., Mazzanti P., Di Gregorio S., “Development and calibration of a preliminary cellular automata model for snow avalanches”, *The Ninth International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 6350, pp. 83–94, 2010.
- [6] Ayala R., Menenti M., Girolana D., “Evaluation methodology for classification process of digital images”, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2002)*, 24-28 June, Toronto (Canada), Volume 6, pp. 3363–3365, 2002.
- [7] Ayala R., Menenti M., “Alternativas a los problemas presentados en un proceso de clasificación basado en el reconocimiento espectral de patrones”, *Mapping*, 75:72–76, 2002.
- [8] Ayala R., Menenti M., “Metodología para la búsqueda del mejor clasificador de imágenes de satélite”, *IX Congreso Nacional de Teledetección*, 19-21 September, Lleida (Spain), pp. 469–472, 2001.

- [9] Ayala R., Becerra A., Flores I.M, Bienvenido J.F, Díaz J.R., “Evaluation of greenhouse covered extensions and required resources with satellite images and GIS. Almería’s case”, *Second European Conference of the European Federation for Information Technology in Agriculture, Food and the Environment*, Bonn (Germany), pp. 27–30, 1999.
- [10] Ayala R., Becerra A., Iribarne L., Bosh A., Díaz J.R., “GIS Systems as a decision support tool for agricultural planning in arid zones of Spain”, *International Commission of Agricultural Engineering*, CIGR electronic Journal I, 1999.
- [11] Ball G.H., Hall D.J., “Isodata: a novel method of data analysis and pattern classification”, *Technical Report of Stanford Research Institute*, Menlo Park, California (United States), Office of Naval Research, Information Sciences Branch, 1965.
- [12] Balzter H., Braun P., Kühler W., “Cellular automata models for vegetation dynamics”, *Ecological Modelling*, 107(2-3):113–125, 1998.
- [13] Bandini S., Bonomi A., Vizzari G., “A cellular automata based modular illumination system”, *The Ninth International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 6350, pp. 334–344, 2010.
- [14] Barberousse A., Franceschelli S., Imbert C., “Cellular automata, modeling, and computation”, *Models and Simulations 2, Conference*, 11-13 October, Tilburg (NL), 2007.
- [15] Barret E.C., Curtis L.F., “Introduction to environmental remote sensing”, *Cheltenham, Stanley Thornes Publishers Ltd*, 1999.
- [16] Bi Y., Zhang Y., Chen Y., “Image classification method based on cellular automata transforms”, *The Sixth World Congress on Intelligent Control and Automation (WCICA 2006)*, 21-23 June, Dalian (China), Volume 2, pp. 10058–10062, 2006.
- [17] Blečić I., Cecchini A., Rizzi P., Trunfio G.A., “Playing with automata. An innovative perspective for gaming simulation”, *5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, October 9-11, Geneva (Switzerland), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 2493, pp. 337–348, 2002.
- [18] Brodatz P., “Textures: a photographic album for artists and designers”, *Dover Publications*, New York, 1966.
- [19] Broomhead D.S., Lowe D., “Radial basis functions, multi-variable functional interpolation and adaptive networks”, *Technical Report*, RSRE 4148, 1988.
- [20] Broomhead D.S., Lowe D., “Multivariate functional interpolation and adaptive networks”, *Complex Systems*, 2:321–355, 1988.

- [21] Carvajal F., Crisanto E., Aguilar F.J., Agüera F., Aguilar M.A., “Greenhouses detection using an artificial neural network with a very high resolution satellite image”, *Proceedings of the ISPRS Symposium Technical Commission II*, 12-14 July, Vienna (Austria), International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences Volume XXXVI - Part 2, pp. 37–42, 2006.
- [22] Castro A., Gómez N., “Self-organizing map and cellular automata combined technique for advanced mesh generation in urban and architectural design”, *International Journal of Information Technologies and Knowledge*, 2:354–360, 2008.
- [23] Chapman S.J., “MATLAB programming with applications for engineers”, *CL-Engineering*, 2012.
- [24] Cheng J., Masser I., “Cellular automata based temporal process understanding of urban growth”, *5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, October 9-11, Geneva (Switzerland), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 2493, pp. 325–336, 2002.
- [25] Ching Fu S., Milne G., “A flexible automata model for disease simulation”, *6th International Conference on Cellular Automata for Research and Industry (ACRI 2004)*, October 25-27, Amsterdam (The Netherlands), in P.M.A. Slood *et al.* eds, Springer, Lecture Notes in Computer Science 3305, pp. 642–649, 2004.
- [26] Chopard B., Lagrava D., “A cellular automata model for species competition and evolution”, *7th International Conference on Cellular Automata for Research and Industry (ACRI 2006)*, September 20-23, Perpignan (France), in S. El Yacoubi *et al.* eds, Springer, Lecture Notes in Computer Science 4173, pp. 277–286, 2006.
- [27] Chuvieco E., Huete A., “Fundamentals of satellite remote sensing”, *CRC Press*, Boca Raton (USA), 2010.
- [28] Chuvieco E., “Teledetección ambiental: la observación de la Tierra desde el espacio”, *Editorial Ariel*, 2008.
- [29] Coppola E., Tomassetti B., Mariotti L., Verdecchia M., Visconti G., “Cellular automata algorithms for drainage network extraction and rainfall data assimilation”, *Hydrological Sciences-Journal-des Sciences Hydrologiques*, 52(3):579–592, 2007.
- [30] Cruz M., Espínola M., Iribarne L., Ayala R., Peralta M., Torres J.A., “Ecological sectorization process improvement through neural networks: synthesis of vegetation data from satellite images using RBFs”, *9th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2010)*, August 18-20, Yamagata (Japan), IEEE Computer Society, pp. 513–516, 2010.
- [31] Cruz M., Espínola M., Iribarne L., Ayala R., Peralta M., Torres J.A., “How can neural networks to speed up ecological regionalization friendly? Replacement of field studies by satellite data using RBFs”, *International Conference on Neural Computation (ICNC 2010)*, October 24-26, Valencia (Spain), 2010.

- [32] D. Ambrosio D., Di Gregorio S., Gabriele S., Gaudio R., “A cellular automata model for soil erosion by water”, *Physics and Chemistry of the Earth (B)*, 26(1):33–39, 2001.
- [33] Das S., Chowdhury D.R., “Generating cryptographically suitable non linear maximum length cellular automata”, *The Ninth International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 6350, pp. 241–250, 2010.
- [34] Doi T., “Quantum cellular automaton for simulating static magnetic fields”, *IEEE Transactions on Magnetics*, 49(5):1617–1620, 2013.
- [35] Dottori F., Todini E., “Developments of a flood inundation model based on the cellular automata approach: testing different methods to improve model performance”, *Physics and Chemistry of the Earth*, 36(7-8): 266–280, 2011.
- [36] Duda R., Hart P., “Pattern classification and scene analysis”, *Wiley*, 1973.
- [37] Dzwinel W., “A cellular automata model of population infected by periodic plague”, *6th International Conference on Cellular Automata for Research and Industry (ACRI 2004)*, October 25-27, Amsterdam (The Netherlands), in P.M.A. Sloot *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 3305, pp. 464–473, 2004.
- [38] Earl G., Johnsonbaugh R., Jost S., “Pattern recognition and image analysis”, *Addison-Wesley*, 1996.
- [39] “Erdas Imagine professional tour guides”, *Leica Geosystems Geospatial Imaging*, Versión 9.2, 2008.
- [40] “Erdas Imagine: macro language reference manual”, *Leica Geosystems Geospatial Imaging*, Versión 9.2, 2008.
- [41] Espínola M., Piedra J.A., Ayala R., Iribarne L., Wang J.Z., “Contextual and hierarchical classification of satellite images based on cellular automata”, *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 53(2):795–809, 2015.
- [42] Espínola M., Piedra J.A., Ayala R., Iribarne L., Leguizamón S., “Modeling rainfall features dynamics in a DEM satellite image with cellular automata”, *11th International Conference on Cellular Automata for Research and Industry (ACRI 2014)*, September 22-25, Kraków (Poland), in J.Was *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 8751, pp. 238–247, 2014.
- [43] Espínola M., Piedra J.A., Ayala R., Iribarne L., Leguizamón S., Menenti M., “ACA multiagent system for satellite image classification”, *10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2012)*, March 28-30, Salamanca (Spain), in J.M.C. Rodríguez *et al.* eds., Trends in Practical Applications of Agents and Multiagent Systems, Springer, Advances in Intelligent and Soft Computing, 157, pp. 93–100, 2012.

- [44] Espínola M., Piedra J.A., Ayala R., Iribarne L., Leguizamón S., Menenti M., “A hierarchical and contextual algorithm based on cellular automata for satellite image classification”, *International Conference on Computational Intelligence and Software Engineering (CiSE 2011)*, December 9-11, Wuhan (China), IEEE Computer Society, 2011.
- [45] Espínola M., Ayala R., Leguizamón S., Iribarne L., Menenti M., “Cellular automata applied in remote sensing to implement contextual pseudo-fuzzy classification”, *9th International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 6350, pp. 312–321, 2010.
- [46] Espínola M., Ayala R., Leguizamón S., Menenti M., “Classification of satellite images using the cellular automata approach”, *First World Summit on the Knowledge Society (WSKS 2008)*, September 24-28, Athens (Greece), in M.D. Lytras *et al.* eds, The Open Knowledge Society: A Computer Science and Information System Manifesto, Springer, Communications in Computer and Information Science, CCIS 19, pp. 521–526, 2008.
- [47] Fix E., Hodges J.L., “Discriminatory analysis, nonparametric discrimination: consistency properties”, *Technical Report 4*, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
- [48] Fúster-Sabater A., Caballero-Gil P., “Chaotic cellular automata with cryptographic application”, *The Ninth International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 6350, pp. 251–260, 2010.
- [49] Gardner M., “The fantastic combination of John Conways new solitaire game Life”, *Scientific American*, 223(4):120–123, 1970.
- [50] Ghaemi M., Naderi O., Zabihinpour Z., “A novel method for simulating cancer growth”, *The Ninth International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 6350, pp. 142–148, 2010.
- [51] Guzman O.F., Gómez I.D., “Fundamentos físicos de teledetección”, *Instituto Geográfico Agustín Codazzi, IGAC*, 2007.
- [52] Haralick R.M., “Statistical and structural approaches to texture”, *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [53] Haralick R.M., Shanmugam K., Dinstein I., “Textural features for image classification”, *IEEE Transactions On System, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

- [54] He D.C., Wang L., “Texture unit, texture spectrum, and texture analysis”, *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 28(4):509-512, 1990.
- [55] IMAGINE Developers Toolkit Network, <http://developer.lggi.com>.
- [56] Kaluza B., “Weka how-to”, *Packt Publishing*, 2013.
- [57] Karafyllidis I., Thanailakis A., “A model for predicting forest fire spreading using cellular automata”, *Ecological Modeling*, 99(1):87-97, 1997.
- [58] Kari J., “Theory of cellular automata: a survey”, *Theoretical Computer Science*, 334(1-3):3-33, 2005.
- [59] Karim F., Walus K., “Efficient simulation of correlated dynamics in quantum-dot cellular automata (QCA)”, *IEEE Transactions on Nanotechnology*, 13(2):294-307, 2014.
- [60] Kowsuwan N., Kanongchaiyos P., “3D cloud animation using CA based method”, *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2009)*., pp. 387-392, 2009.
- [61] Le Cam L., “Maximum likelihood: an introduction”, *International Statistical Review* 58, Volume 2, pp. 153-171, 1990.
- [62] Leguizamón S., Espínola M., Ayala R., Iribarne L., Menenti M., “Characterization of texture in images by using a cellular automata approach”, *Third World Summit on the Knowledge Society (WSKS 2010)*, September 22-24, Corfu (Greece), in M.D. Lytras *et al.* eds, *The Open Knowledge Society: Organizational, Business, and Technological Aspects of the Knowledge Society*, part II, Springer, Communications in Computer and Information Science, CCIS 112, pp. 522-533, 2010.
- [63] Leguizamón S., “Modeling land features dynamics by using cellular automata techniques”, *Proceedings of the ISPR Technical Comision 7 Mid-Term Symposium From pixels to Processess*”, Enschede (The Netherlands), pp. 497-501, 2006.
- [64] Leguizamón S., “Simulation of snow-cover dynamics using the cellular automata approach”, *Proceedings of the 8th International Symposium on High Mountain Remote Sensing Cartography*, La Paz (Bolivia), pp. 87-91, 2005.
- [65] Leguizamón S., “Characterization of texture in remotely sensed images by using the wavelet transform”, *Proceedings of the IV International Symposium on HMRS Cartography*, University of Karlstad, Karlstad, 1996.
- [66] Leguizamón S., “Description of terrain textures by fractal and Markov random fields techniques”, *Proceedings of the 2nd International Symposium on HMRS Cartography*, Chinese Academy of Sciences, Astronautic Publ. House, P.R. of China, Beijing, 1993.

- [67] Li X., Gar-on Yeh A., “Neural-network-based cellular automata for simulating multiple land use changes using GIS”, *International Journal of Geographical Information Science*, 16(4):323–343, 2002.
- [68] Lobitz B., Beck L., Huq A., Woods B., Fuchs G., Faruque A., Colwell R., “Climate and infectious disease: use of remote sensing for detection of vibrio cholerae by indirect measurement”, *Proceedings of the National Academic of Sciences of the USA*, 97(4):1438–43, 2000.
- [69] MacQueen J.B., “Some methods for classification and analysis of multivariate observations”, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, Volume 1: Statistics, pp. 281–297, 1967.
- [70] Mahajan Y., Venkatachalam P., “Neural network based cellular automata model for dynamic spatial modeling in GIS”, *Computational Science and Its Applications (ICCSA 2009)*, Seoul (Korea), June 29 - July 2, Springer, Lecture Notes in Computer Science, LNCS 5592, pp. 341–352, 2009.
- [71] Maji P., Shaw C., Ganguly N., Sikdar B., Chaudhuri P., “Theory and application of cellular automata for pattern classification”, *Fundamenta Informaticae*, 58(3-4):321–354, 2003.
- [72] Marcou O., El Yacoubi S., Chopard B., “A bi-fluid lattice boltzmann model for water flow in an irrigation channel”, *7th International Conference on Cellular Automata for Research and Industry (ACRI 2006)*, September 20-23, Perpignan (France), in S. El Yacoubi *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 4173, pp. 373–382, 2006.
- [73] Mather P., Tso B., “Classification methods for remotely sensed data”, *CRC Press*, 2 Edition, 2009.
- [74] Messina J., Walsh S., “Simulating land use and land cover dynamics in the ecuadorian Amazon through cellular automata approaches and an integrated GIS”, *Open Meeting of the Human Dimensions of Global Environmental Change Research Community*, Rio de Janeiro (Brazil), pp. 6–8, 2001.
- [75] Minsky M., “Steps toward artificial intelligence”, *Transactions on Institute of Radio Engineers*, 49:8–30, 1961.
- [76] Mojaradi B., Lucas C., Varshosaz M., “Using learning cellular automata for post classification satellite imagery”, *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 35(4):991–995, 2004.
- [77] Moreno N., Quintero R., Ablan M., Barros R., Dávila J., Ramírez H., Tonella G., Acevedo M., “Biocomplexity of deforestation in the Caparo tropical forest reserve in Venezuela: an integrated multi-agent and cellular automata model”, *Environmental Modelling and Software*, 22(5):664–673, 2007.

- [78] Muzy A., Innocenti E., Aiello A., Santucci J.F., Santonio P.A., Hill D., “Modelling and simulation of ecological propagation processes: application to fire spread”, *Environmental Modelling and Software*, 20(7):827–842, 2005.
- [79] Muzy A., Innocenti E., Aiello A., Santucci J.F., Santonio P.A., Hill D., “Dynamic structure cellular automata in a fire spreading application”, *First International Conference on Informatics in Control, Automation and Robotics*, Setubal (Portugal), pp. 143–151, 2004.
- [80] Piedra-Fernández J.A., Cantón-Garbín M., Wang J.Z., “Feature selection in AVHRR ocean satellite images by means of filter methods”, *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 48(12):4193–4203, 2010.
- [81] Popovici A., Popovici D., “Cellular automata in image processing”, *Proceedings of the 15th International Symposium on the Mathematical Theory of Networks and Systems*, University of Notre Dame, August 12-16, 2002.
- [82] Preston K., Duff M., “Modern cellular automata, theory and applications”, *Plenum Press*, New York, 1984.
- [83] Proyecto SOLERES, <http://acg.ual.es/soleres>.
- [84] Proyecto SOLERES, Información ecológica, <http://acg.ual.es/soleres/ecology.html>.
- [85] Proyecto iSOLERES, <http://acg.ual.es/isoleres/>.
- [86] Quartieri J., Mastorakis N.E., Iannone G., Guarnaccia C., “Cellular automata application to traffic noise control”, *Proceedings of the 12th WSEAS International Conference on Automatic Control, Modeling and Simulation (ACMOS10)*, Catania (Italy), pp. 29–31, 2010.
- [87] Quinlan J.R., “Improved use of continuous attributes in C4.5”, *Journal of Artificial Intelligence Research*, 4(1):77–90, 1996.
- [88] Quinlan J.R., “C4.5: programs for machine learning”, *Morgan Kaufmann Publishers*, 1993.
- [89] Rees W.G., “Physical principles of remote sensing”, *Cambridge University Press*, 2nd Edition, 2001.
- [90] Riener A., Fullerton M., Maag C., Mark C., Beltran Ruiz C., Minguez Rubio J.J., Zia K., “Modular simulation-based physical and emotional assessment of ambient intelligence in traffic”, *IEEE Transactions on Human-Machine Systems*, 44(2):286–292, 2014.
- [91] Rinaldi P., Dalponte D., Vénere M., Clause A., “Cellular automata algorithm for simulation of surface flows in large plains”, *Simulation Modelling Practice and Theory*, 15(3):315–327, 2007.



- [92] Rosenblatt F., “Principles of neurodynamics: perceptrons and the theory of brain mechanisms”, *Spartan Books*, Washington DC, 1961.
- [93] Rosenblatt F., “The perceptron: a probabilistic model for information storage and organization in the brain”, *Psychological Review*, 65:386–408, 1958.
- [94] Sarhan E., Khalifa E., Nabil A.M., “Post classification using cellular automata for Landsat images in developing countries”, *International Conference on Image Information Processing (ICIIP 2011)*, 3-5 November, Himachal Pradesh (India), pp. 1–4, 2011.
- [95] Schonfisch B., Kinder M., “A fish migration model”, *5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, October 9-11, Geneva (Switzerland), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 2493, pp. 210–219, 2002.
- [96] Schowengerdt R.A., “Techniques for image processing and classification in remote sensing”, *Academic Press*, 1985.
- [97] Shiraishi T., Motohka T., Thapa R.B., Watanabe M., Shimada M., “Comparative assessment of supervised classifiers for land use-land cover classification in a tropical region using time-series PALSAR mosaic data”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(4):1186–1199, 2014.
- [98] Skurikhin A.N., Wilson C.J., Liljedahl A., Rowland J.C., “Recursive active contours for hierarchical segmentation of wetlands in high-resolution satellite imagery of Arctic landscapes”, *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI 2014)*, 6-8 April, San Diego, California (USA), pp. 137–140, 2014.
- [99] Sloot P., Chen F., Boucher C., “Cellular automata model of drug therapy for HIV infection”, *5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, October 9-11, Geneva (Switzerland), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 2493, pp. 282–293, 2002.
- [100] Spataro W., D. Ambrosio D., Rongo R., Trunfio G.A., “An evolutionary approach for modelling lava flows through cellular automata”, *6th International Conference on Cellular Automata for Research and Industry (ACRI 2004)*, October 25-27, Amsterdam (The Netherlands), in P.M.A. Sloot *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 3305, pp. 725–734, 2004.
- [101] Toffoli T., Margolus N., “Invertible Cellular Automata: A Review”, *Physica D*, 45(1-3):229–253, 1990.
- [102] Topa P., “Towards a two-scale cellular automata model of tumour-induced angiogenesis”, *7th International Conference on Cellular Automata for Research and Industry (ACRI 2006)*, September 20-23, Perpignan (France), in S. El Yacoubi *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 4173, pp. 337–346, 2006.

- [103] Tsiftsis A., Sirakoulis G., Lygouras J., “Design of a cellular automaton model for railway traffic flow with GPS module”, *The Ninth International Conference on Cellular Automata for Research and Industry (ACRI 2010)*, September 21-24, Ascoli Piceno (Italy), in S. Bandini *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 6350, pp. 373–384, 2010.
- [104] Umarani C., Ganesan L., Radhakrishnan S., “A combined statistical and structural approach for texture representation”, *Asian Journal of Information Technology*, 5(12):1434–1440, 2006.
- [105] von Neumann J., “Theory of self-reproducing automata”, *University of Illinois Press*, Champaign, 1966.
- [106] Wang J., Kropff M., Lammert B., Christensen S., Hansen P., “Using CA model to obtain insight into mechanism of plant population spread in a controllable system: annual weeds as an example”, *Ecological Modelling*, 166(3):277–286, 2003.
- [107] Wang Z., Nasrabadi N.M., Huang T.S., “Semisupervised hyperspectral classification using task-driven dictionary learning with laplacian regularization”, *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 53(3):1161–1173, 2015.
- [108] Wolfowitz J., “Estimation by the minimum distance method”, *Annals of the Institute of Statistical Mathematics*, Volume 5, pp. 9–23, 1953.
- [109] Wolfram S., “A new kind of science”, *Wolfram Media*, Inc., Champaign, Illinois, 2002.
- [110] Wolfram S., “Universality and complexity in cellular automata”, *Physica D*, 10:1–35, 1984.
- [111] Wolfram S., “Cellular automata”, *Los Alamos Science*, 9, pp. 2–21, 1983.
- [112] Wolfram S., “Cellular automata as simple self-organizing systems”, *Caltech preprint CALT*, pp. 68–938, 1982.
- [113] Xia Z., Jiang Y., “A novel artificial life ecosystem environment model”, *6th International Conference on Cellular Automata for Research and Industry (ACRI 2004)*, October 25-27, Amsterdam (The Netherlands), in P.M.A. Sloot *et al.* eds, Springer, Lecture Notes in Computer Science, LNCS 3305, pp. 650–659, 2004.
- [114] Yang L., Yang S., Jin P., Zhang R., “Semi-supervised hyperspectral image classification using spatio-spectral laplacian support vector machine”, *IEEE Geoscience and Remote Sensing Letters*, 11(3):651–655, 2014.
- [115] Zhang H., Shi W., Wang Y., Hao M., Miao Z., “Spatial-attraction-based Markov random field approach for classification of high spatial resolution multispectral imagery”, *IEEE Geoscience and Remote Sensing Letters*, 11(2):489–493, 2014.

Este documento ha sido generado con  $\text{\LaTeX}$ .

Tesis doctoral: "Clasificación de Imágenes de Satélite mediante Autómatas

Antonio Moisés Espínola Pérez  
Grupo de investigación de Informática Aplicada  
Departamento de Informática  
Universidad de Almería  
Almería, 21 de Noviembre de 2014  
<http://www.ual.es/personal/moises.espinola/>

