

INSTITUTO TECNOLÓGICO DE CHIHUAHUA
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

***“SEGMENTACIÓN DE OBJETOS EN
SECUENCIAS DE VIDEO BAJO ENFOQUE
NEUROINSPIRADO”***

TESIS

QUE PARA OBTENER EL GRADO DE

***MAESTRO EN CIENCIAS
EN INGENIERÍA ELECTRÓNICA***

PRESENTA:

ING. RAÚL RANGEL GONZÁLEZ

**DIRECTOR DE LA TESIS:
DR. JUAN ALBERTO RAMÍREZ QUINTANA**

CHIHUAHUA, CHIH., DICIEMBRE 2019



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®





Chihuahua, Chih. 5 de diciembre de 2019

C. RAÚL RANGEL GONZÁLEZ
PRESENTE

Por este conducto le comunico que a propuesta del Jurado de Examen, la División de Estudios de Posgrado e Investigación ha concedido autorización para la impresión de su tesis para obtener el grado de Maestro en Ciencias en Ingeniería Electrónica, cuyo título es:

"Segmentación de objetos en secuencias de video bajo enfoque neuroinspirado"

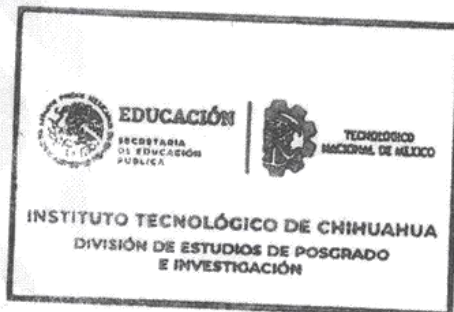
La tesis presenta el siguiente contenido de capítulos:

- I Antecedentes
- II Segmentación de objetos mediante agrupamiento
- III Segmentación neuroinspirada
- IV Resultados y conclusiones

ATENTAMENTE

Excelencia en Educación Tecnológica®
"La técnica por el Engrandecimiento de México"

MTRO. LUIS CARDONA CHACÓN
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN





Chihuahua, Chih. 5 de diciembre de 2019

MTRO. LUIS CARDONA CHACÓN
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
PRESENTE

Por medio de la presente notificamos a usted que en cumplimiento de los requerimientos para la obtención del grado de Maestro en Ciencias en Ingeniería Electrónica, el documento de tesis del **C. RAÚL RANGEL GONZÁLEZ**, ha sido aprobado y aceptado para su impresión. El título de la tesis es:

"Segmentación de objetos en secuencias de video bajo enfoque neuroinspirado"

Por lo que proponemos, le sea concedida la autorización de impresión correspondiente.

Agradeciendo la atención a la presente, quedamos de usted:

ATENTAMENTE

*Excelencia en Educación Tecnológica.
"La técnica por el Engrandecimiento de México"*

Juan Ramirez

DR. JUAN ALBERTO RAMÍREZ QUINTANA
DIRECTOR DE TESIS

MTRA. ALMA DELIA CORRAL SÁENZ
MIEMBRO DEL JURADO DE EXAMEN

Graciela Ramirez A.

DRA. GRACIELA MARÍA DE JESÚS
RAMÍREZ ALONSO
MIEMBRO DEL JURADO DE EXAMEN

DR. MARIO IGNACIO CHACÓN MURGUÍA
MIEMBRO DEL JURADO DE EXAMEN





CARTA CESIÓN DE DERECHOS

En la ciudad de Chihuahua el día **12 de diciembre de 2019**, el que suscribe **C. RAÚL RANGEL GONZÁLEZ** de la Maestría en Ciencias en Ingeniería Electrónica, con número de control **G17061503**, adscrito a la **DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**, del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del **Dr. Juan Alberto Ramírez Quintana** y cede los derechos del trabajo titulado **"Segmentación de objetos en secuencias de video bajo enfoque neuroinspirado"**, al Tecnológico Nacional de México y/o Instituto Tecnológico de Chihuahua para su difusión, divulgación, transmisión, reproducción, así como su digitalización con fines académicos y de investigación.

Raúl RG

C. RAÚL RANGEL GONZÁLEZ



Chihuahua, Chihuahua, a 12 de diciembre de 2019

Dra. María Elena Álvarez-Buylla Roces
Directora del CONACYT

Por este conducto aprovecho la ocasión para saludarla e informarle que a la fecha he obtenido el grado de **Maestría en Ciencias de Ingeniería Electrónica** en la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua. Agradezco todo el apoyo brindado por CONACYT ya que el otorgamiento de la beca para estudios de Posgrado, permitió dedicarme de tiempo completo al programa y con ello lograr el cumplimiento del objetivo principal del convenio establecido.

Sin otro particular por el momento, me es grato quedar de usted como su seguro servidor, no sin antes reiterar nuevamente todo mi agradecimiento.

A t e n t a m e n t e



Raúl Rangel González
Exbecario CONACYT

AGRADECIMIENTOS

A mi familia

En especial a mis padres y hermanos porque con su apoyo y amor incondicional, y también por su paciencia, he tenido la posibilidad de terminar esta etapa más en mi carrera.

A mis amigos y compañeros

Agradezco a mis amigos de generación Andrea, Adrián y Daniel, por su gran apoyo y ánimo a lo largo de la maestría. Agradezco a mis amigos y compañeros de laboratorio David, Macias, Luis, Jorge, Abimael, Soto, Lalo, Esquivel, Camacho, Alonso, Javier, Iván y Xavier por el agradable ambiente de trabajo.

A mi comité de tesis

A mi comité de tesis. Agradezco a mi asesor el Dr. Juan Alberto Ramírez Quintana por sus consejos y conocimientos brindados, por guiarme a lo largo de la realización de este trabajo de tesis. Agradezco al Dr. Mario Chacón por los valiosos conocimientos que compartió en su clase de redes neuronales y lógica difusa, y por el asesoramiento que me brindó en mis residencias de ingeniería, que fue lo que me motivo a realizar esta maestría. Agradezco a la M.C. Alma Corral por la calidad con la impartió la materia procesamiento de señales y por sus consejos y observaciones que me brindó en mis comités tutoriales. Agradezco a la Dra. Graciela Ramírez por sus comentarios y aportes a este trabajo.

Al Tecnológico Nacional de México y CONACYT

Agradezco al Tecnológico Nacional de México por el apoyo a este trabajo bajo los proyectos “Diseño de plataforma tecnológica de internet de las cosas para generar conectividad informática en sistemas de procesamiento inteligente de señales enfocados a cuidados médicos y discapacidad” clave 6418.18-P” y “Algoritmo de detección de movimiento en secuencias de video basado en arquitectura de redes neuronales tradicionales o de aprendizaje profundo y lógica difusa”, clave 5162.19-P.

RESUMEN

SEGMENTACIÓN DE OBJETOS EN SECUENCIAS DE VIDEO BAJO ENFOQUE NEUROINSPIRADO

Ing. Raúl Rangel González
Maestría en Ciencias en Ingeniería Electrónica
División de Estudios de Posgrado e Investigación del
Instituto Tecnológico de Chihuahua
Chihuahua, Chih. 2019
Director de Tesis: Dr. Juan Alberto Ramírez Quintana

La percepción de objetos es una habilidad compleja que se ha estudiado mediante diversos modelos neurocomputacionales para entender cómo el cerebro interpreta una escena visual y sus propiedades. Actualmente, dichos modelos han sido utilizados en visión por computadora para proponer nuevos esquemas de procesamiento de video que han tenido resultados comparables con los métodos tradicionales de procesamiento de imágenes. Por lo tanto, contribuyendo con el esfuerzo de proponer nuevas formas de procesamiento basados en modelos neurocomputacionales, en esta tesis se proponen dos métodos para obtener información de la composición de los objetos que se encuentran en escenarios variantes en el tiempo.

Un método se denomina Detección de Objetos mediante Agrupamiento (DObjA) y detecta los objetos de un escenario mediante un algoritmo basado en la detección de bordes y agrupamiento estadístico. El esquema de DObjA se obtuvo de los resultados de un experimento con señales de electroencefalografía (EEG) que se realizó para analizar los potenciales evocados de un sujeto cuando se expone a estímulos relacionados al reconocimiento de objetos. El segundo método se denomina Segmentación de Objetos Neuroinspirada (NISeg) que consiste en obtener características de bordes, color y movimiento para la segmentación de objetos. Este método se basó en los modelos LAMINART, LISSOM, Wilson & Cowan, redes oscilatorias y redes integra-dispara. Estos son los modelos neurocomputacionales más populares en la literatura que explican diversos aspectos del comportamiento de la corteza visual.

Para las pruebas y los experimentos, se utilizó la base de datos de *Video Segmentation Benchmark* (VSB100) debido a que contiene videos de escenarios variados, toma en

consideración todos los objetos del escenario y utiliza 2 métricas para analizar la calidad de la coherencia espacial y temporal de modelos de segmentación de objetos no supervisados.

DObjA y NISeg se probaron en esta base de datos y se observó que DObjA quedó con resultados aceptables en la métrica de coherencia espacial, teniendo como una de sus ventajas el bajo costo computacional que se requiere para ejecutarlo. También sirvió para conocer el desempeño de los algoritmos de agrupamiento que se basan en las características de bajo nivel que utiliza el sistema de percepción visual humano, mientras que NISeg si logró resultados aceptables en ambas métricas, lo que indica la factibilidad de implementar algoritmos de segmentación de objetos basados en el proceso de percepción visual humana.

CONTENIDO

LISTA DE FIGURAS.....	x
LISTA DE TABLAS	xiv
INTRODUCCIÓN	xv
 CAPÍTULO I ANTECEDENTES	 1
1.1. Corteza visual.....	1
1.2. Segmentación de objetos.....	7
1.2.1. Segmentación semántica.....	7
1.2.2. Segmentación de saliencia	10
1.2.3. Segmentación mediante métodos no supervisados.....	11
1.2.4. Segmentación de objetos en secuencias de video.....	12
1.3. Estado del arte de segmentación neuroinspirada de imágenes y video.....	14
 CAPÍTULO II SEGMENTACIÓN DE OBJETOS MEDIANTE AGRUPAMIENTO ..	 17
2.1. Detección de objetos y electroencefalografía	17
2.1.1. Estímulo visual y esquema de experimentación	19
2.1.2. Sujeto de pruebas	21
2.1.3. Sistema de adquisición	23
2.1.4. Procesamiento.....	25
2.1.5. Conclusiones del experimento EEG	27
2.2. Algoritmos propuestos	29
2.2.1. Preprocesamiento y obtención de características	30
2.2.2. Algoritmo K-means	33
2.2.3. Algoritmo FCM	35
2.2.4. Algoritmo de EM.....	37
2.2.5. Determinación del número de grupos y segmentación de objetos	39
 CAPÍTULO III SEGMENTACIÓN NEUROINSPIRADA	 42
3.1. Retina y NGL	43

3.1.1.	Modelos neurocomputacionales de la retina y los NGL	43
3.1.2.	Modelo de la retina y los NGL en NISeg	45
3.2.	Corteza visual primaria (V1) y secundaria (V2)	47
3.2.1.	Color	47
3.2.1.1.	Modelos neurocomputacionales de color	47
3.2.1.2.	Modelo de color en NISeg	49
3.2.2.	Orientación	54
3.2.2.1.	Modelos neurocomputacionales de orientación	54
3.2.2.2.	Modelo de orientación en NISeg	55
3.2.3.	Movimiento	59
3.2.3.1.	Modelos neurocomputacionales de movimiento	59
3.2.3.2.	Modelo de movimiento en NISeg	59
3.3.	Área MT	63
3.3.1.	Modelo neurocomputacional del área MT	63
3.3.2.	Modelo de MT en NISeg para coherencia temporal	64
3.4.	Área V4	68
3.4.1.	Modelo neurocomputacional del área V4	68
3.4.2.	Modelo de V4 en NISeg para coherencia espacial	69
CAPÍTULO IV RESULTADOS Y CONCLUSIONES		72
4.1.	Base de datos VSB100	72
4.1.1.	Métricas de evaluación	73
4.2.	Resultados con VSB100	77
4.3.	Conclusiones	82
REFERENCIAS		84

LISTA DE FIGURAS

CAPÍTULO I ANTECEDENTES	1
Figura 1.1. Áreas de la corteza visual [10].	2
Figura 1.2. Capa de la retina y las células que la conforman.....	2
Figura 1.3. Camino que siguen las señales eléctricas de los conos y bastones hasta las células ganglionares.	3
Figura 1.4. a) CR a nivel celular, b) CR de células sensibles a la luminancia, c) CR de células sensibles ante colores oponentes.....	4
Figura 1.5. Patrón de disparo de un CR ON/OFF ante 4 diferentes estímulos.	4
Figura 1.6. Esquema sobre la vía óptica del ser humano [1].	5
Figura 1.7. Enfoque clásico del procesamiento visual.....	6
Figura 1.8. Segmentación semántica.....	7
Figura 1.9. Arquitectura de una red semántica basada en regiones [19].	8
Figura 1.10. Segmentación semántica basada en FCN [21].	9
Figura 1.11. a) Imagen original, b) Imagen de saliencia respectiva.	10
Figura 1.12. a) Imagen original, b) Segmentación K-means en 2 grupos, c) Detección del tumor [28].	11
Figura 1.13. a) Imagen original, b) Materia blanca, c) Materia gris, d) Fluido cerebroespinal [29].	12
 CAPÍTULO II SEGMENTACIÓN DE OBJETOS MEDIANTE AGRUPAMIENTO ..	17
Figura 2.1. Etapas para la realización del experimento de detección de objetos mediante EEG.	19
Figura 2.2. Estímulos visuales experimentales.	20
Figura 2.3. Par de figuras para cada secuencia.	20
Figura 2.4. Presentación de las secuencias.	21
Figura 2.5. Habitación donde se realizan las tomas de muestras.	23
Figura 2.6. Electrodo seco del sistema de adquisición OpenBCI® [52].	23
Figura 2.7. Electrodo utilizado del sistema internacional 10-20.	24
Figura 2.8. Dispositivo de toma de muestras.	25

Figura 2.9. Señal P300 extraída de un canal del vector $x_{sp}^{n_e}(m_s)$	27
Figura 2.10. Ejemplo de segmentación de objetos basada en la sincronización/desincronización de las neuronas, a) Imagen original, b) Oscilación de las neuronas, c) Segmentación de objetos.	29
Figura 2.11. Esquema de los métodos propuestos de agrupamiento.	30
Figura 2.12. Filtro Gaussiano pasa bajas propuesto.	31
Figura 2.13. a) Imagen original $I_{RGB}^t(x, y)$, b) Imagen de luminancia $I_L^t(x, y)$, c) Imagen filtrada $I_G^t(x, y)$	31
Figura 2.14. a) Cuadro de luminancia $I_L^t(x, y)$, b) Imagen de bordes $I_{LB}^t(x, y)$ del cuadro de luminancia.	32
Figura 2.15. Diagrama del algoritmo de K-means.....	35
Figura 2.16. Diagrama del algoritmo de FCM.....	37
Figura 2.17. Diagrama del algoritmo de mezclas gaussianas usando EM.....	39
Figura 2.18. Cuadros de bordes. a) $I_{LB}^t(x, y)$, b) $I_{B_1}^t(x, y)$, c) $I_{B_2}^t(x, y)$ y d) $I_{B_3}^t(x, y)$	41
Figura 2.19. a) Imagen original $I_{RGB}(x, y, z)$, b) Imagen segmentada con K-means $I_{A_1}^t(x, y)$, b) Imagen segmentada con FCM $I_{A_2}^t(x, y)$, c) Imagen segmentada con modelado de mezclas gaussianas $I_{A_3}^t(x, y)$	41

CAPÍTULO III SEGMENTACIÓN NEUROINSPIRADA42

Figura 3.1. Diagrama propuesto de la percepción visual humana.	43
Figura 3.2. a) Envolvente y centro de un CR, b) DoG que simula un CR mostrado transversalmente.....	44
Figura 3.3. a) Codificación de un cuadro $I_{RGB}^t(x, y, z)$ en componentes de CIELab, b) $I_L^t(x, y)$ luminancia, c) $I_a^t(x, y)$ para rojo, d) $I_a^t(x, y)$ para verde, e) $I_b^t(x, y)$ para amarillo f) $I_b^t(x, y)$ para azul.....	46
Figura 3.4. Organización de un mapa de color. a) Mapa de color selectivo, b) Contornos de color. 48	
Figura 3.5. Conexión de los campos receptivos ante el estímulo rojo.....	48
Figura 3.6. Conexiones laterales de una neurona de selectividad al rojo. a) Mapa, b)	

Histograma [70].	49
Figura 3.7. a) Árbol de expansión con suma de pesos de 11 y b) MST con suma de pesos de 7.	51
Figura 3.8. a) Imagen original dividida en píxeles y b) MST correspondiente [74].	51
Figura 3.9. a) Imagen original $I_{RGB}^t(x, y)$, b) Imagen segmentada $I_{GB}^t(x, y)$, c) Extracción del objeto $I_{GB}^{t, m_{GB}}(x, y)$, d) histograma $h_a^{\Omega_T, r, t}$, e) histograma $h_b^{\Omega_T, r, t}$ y f) histograma total $h_T^{\Omega_T, r, t}$.	54
Figura 3.10. a) Señal Gaussiana, b) Señal senoidal, c) Filtro de Gabor.	56
Figura 3.11. Filtros de Gabor de orientación a) de 0° b) de 45° c) de 90° y d) de 135° .	56
Figura 3.12. Proceso de la CeNN <i>feedforward</i> implementada.	58
Figura 3.13. Estructura del sistema de una CeNN con retroalimentación cero [78].	58
Figura 3.14. a) Imagen original $I_{RGB}^t(x, y, z)$, b) Imagen de luminancia $I_L^t(x, y)$, c) Imagen resultante $I_{CN}^t(x, y)$.	59
Figura 3.15. Ejemplo de expansión radial del flujo óptico [81].	60
Figura 3.16. a) Primer cuadro $I_L^t(x, y)$, b) Segundo cuadro $I_L^{t+1}(x, y)$, c) Cuadro de diferencia temporal $I_T(x_{LK})$, d) Magnitud de flujo óptico resultante $I_{FO}^t(x, y)$.	63
Figura 3.17. Ejemplo del problema de apertura.	64
Figura 3.18. Proceso para obtener la coherencia temporal.	66
Figura 3.19. Proceso para la obtención la de movimiento de los objetos. a) Cuadro original $I_{RGB}^t(x, y, z)$, b) Magnitud de FO $I_{FO}^t(x, y)$, c) Binarización de magnitud del FO R_{B, n_R}^t , d) Regiones de los objetos por GBS $I_{GB}^t(x, y)$, e) Extracción de uno de los objetos $R_{B, r1, n_R}^t$, f) Movimiento del objeto.	68
Figura 3.20. a) Cuadro original $I_{RGB}^t(x, y, z)$, b) Ground truth, c) Cuadro $I_S^t(x, y)$.	71

CAPÍTULO IV RESULTADOS Y CONCLUSIONES.....72

Figura 4.1. a) Cuadro original $I_{RGB}^t(x, y, z)$, b) Segmentación, c) Sobresegmentación.	75
Figura 4.2. VPR explicada por coherencia temporal con los cuadros. a) $I_{RGB}^t(x, y, z)$, b) Segmentación deseada en t , c) Segmentación incorrecta en t , d) $I_{RGB}^{t+1}(x, y, z)$, e)	

Segmentación deseada en $t + 1$, f) Segmentación incorrecta en $t + 1$	76
Figura 4.3. a) Cuadro original $I_{RGB}^t(x, y, z)$, b) <i>Ground truth</i> , c) Segmentación con NISeg $I_S^t(x, y)$, d) <i>K-means</i> $I_{A_1}^t(x, y)$, e) FCM $I_{A_2}^t(x, y)$ y f) EM $I_{A_3}^t(x, y)$	81

LISTA DE TABLAS

CAPÍTULO II SEGMENTACIÓN DE OBJETOS MEDIANTE AGRUPAMIENTO ..17

Tabla 2.1. Cantidad de señales P300 detectadas.....27

CAPÍTULO IV RESULTADOS Y CONCLUSIONES72

Tabla 4.1. Desempeño de diferentes algoritmos de segmentación de objetos.....79

INTRODUCCIÓN

La percepción visual es una habilidad compleja que la mente humana hace de forma intuitiva, ya que el cerebro reconoce en forma instantánea características tales como luz, color, forma, profundidad y movimiento [1][2][3]. Para entender esta habilidad, las neurociencias computacionales han generado una vasta cantidad de conocimientos que permiten tener un entendimiento más profundo del proceso visual a nivel cerebral y mental [4][3]. Estos conocimientos han sido aprovechados en otras áreas como visión por computadora y procesamiento digital de imágenes para proponer nuevos modelos y algoritmos de segmentación, detección de bordes, movimiento, etc.

En consecuencia, existen modelos neurocomputacionales que se desarrollan para explicar teorías de percepción visual y también se han utilizado en procesamiento digital de imágenes, por ejemplo, la Red de Osciladores Estimulados Localmente e Inhibidos Globalmente (LEGION por sus siglas en inglés) [5], Wilson & Cowan [6], Red Neuronal Pulsoacoplada (PCNN por sus siglas en inglés) [7], Mapa Autoorganizado Sinérgico e Interconectado Localmente (LISSOM por sus siglas en inglés) [6], etc. Estos modelos segmentan adecuadamente los objetos en una imagen ya que interpretan y analizan todo el contenido del escenario, pero pueden tener problemas cuando se prueban en videos ya que en este caso, el escenario varía en el tiempo [8][9]. Sin embargo, existen otros algoritmos neuroinspirados que funcionan adecuadamente en procesamiento de video. Estos algoritmos se basan en redes neuronales artificiales y son muy populares en la literatura, pero tienen como desventajas que requieren entrenamiento muy exhaustivo o descartan una gran cantidad de información que brinda el escenario, ya que se enfocan a solucionar problemas muy particulares como detección de movimiento, modelado fondo, segmentación de objetos con pregnancia, etc.

Es decir, los modelos basados en percepción visual segmentan utilizando como información una interpretación detallada del escenario, pero tienen malos desempeños en video. Por otro lado, los algoritmos de procesamiento de video basados en redes neuronales tienen buenos desempeños pero funcionan para problemas muy específicos. Consecuentemente, es necesario desarrollar más investigación para generar nuevos algoritmos de segmentación de objetos que analicen como los modelos de percepción visual pero que tengan buenos resultados en

procesamiento de video [10]. Por lo tanto, contribuyendo con este esfuerzo de proponer nuevos esquemas de segmentación en video, en esta tesis se proponen dos métodos neuroinspirados para obtener información de la composición de los objetos que se encuentran en escenarios variantes en el tiempo. Uno se denomina DObjA y se desarrolló a partir de información obtenida del comportamiento de las señales EEG ante la presentación de ciertos estímulos a un sujeto de prueba, y el otro método es NISeg y consiste en aplicar los modelos de la corteza visual populares en la literatura al problema de la segmentación de objetos. A partir de estos métodos, se pretende explorar una nueva forma de análisis del escenario que resuelva de forma más efectiva problemas que los actuales algoritmos no resuelven bien, como: movimientos de cámara, ruido de adquisición en secuencias de video, cambios de iluminación y analizar e interpretar cualquier objeto del escenario.

Para dar a conocer los algoritmos propuestos, esta tesis se conforma de 4 capítulos. En el capítulo I se describen los antecedentes necesarios para llevar a cabo el proyecto de tesis, donde abarcan temas como la corteza visual y la segmentación de objetos. En los capítulos II, III se presentan los métodos de segmentación de objetos en secuencias de video propuestos. En el capítulo IV se muestran los resultados cuantitativos obtenidos de los diferentes métodos y las conclusiones.

CAPÍTULO I

ANTECEDENTES

Los algoritmos neuroinspirados son modelos o sistemas matemáticos diseñados a partir de los mecanismos perceptuales generados por el sistema nervioso y el cerebro. Para el caso del procesamiento digital de imágenes, estos algoritmos se basan en el comportamiento de las capas corticales de la corteza visual como lo son: la retina, los campos receptivos (CR), los núcleos geniculados laterales (NGL), las áreas de la corteza visual V1, V2, V4 y el área medio temporal (MT). Para simular estas capas o implementar sus mecanismos en cualquier algoritmo de computación, es necesario recurrir a diversos conceptos relacionados a teoría de filtros, análisis en la frecuencia, aprendizaje automático, inteligencia computacional, modelos gráficos paramétricos y no paramétricos, etc.

En el caso de esta tesis, el enfoque es desarrollar métodos neuroinspirados aplicados a la segmentación de objetos y por lo tanto, en este capítulo se presentan diversos conceptos con la estructura siguiente: la sección 1.1 trata sobre los fundamentos sobre la corteza visual. La sección 1.2 describe conceptos básicos de procesamiento de imágenes aplicado a segmentación de objetos. Finalmente, la subsección 1.3 presenta un análisis del estado del arte de los algoritmos de segmentación de objetos que utilizan las teorías de percepción visual.

1.1. Corteza visual

La corteza visual es la parte del cerebro que se encarga del procesamiento del sentido de la visión y está localizada en el lóbulo occipital (parte posterior del cerebro). Esta área del cerebro es la más estudiada y a partir de ella se han inspirado gran cantidad de algoritmos y modelos para el procesamiento de imágenes y video.

De acuerdo a la neurociencia, el sistema de visión del ser humano empieza en la retina, luego se encuentra el NGL y la corteza visual, quien se divide a su vez en las áreas definidas como primaria (V1), secundaria (V2), cuarta (V4) y medio temporal (MT) o V5 [11][12], como se muestra en la Figura 1.1.

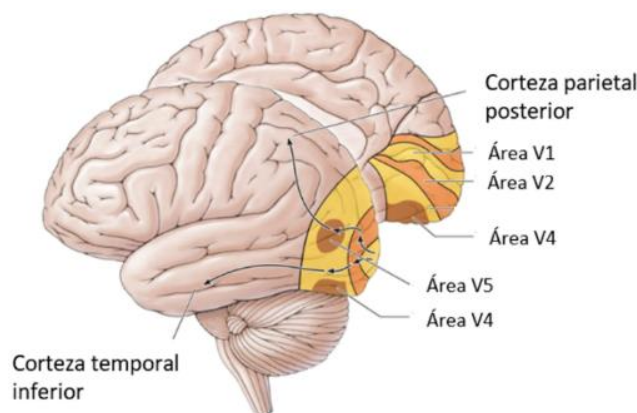


Figura 1.1. Áreas de la corteza visual [13].

La retina es la región sensorial del ojo que inicia el procesamiento visual y se localiza en la parte posterior del globo ocular, como se observa en la Figura 1.2. Histológicamente, tiene 3 capas que contienen 4 tipos de células diferentes, separadas por dos interfaces sinápticas comunicadas por un neurotransmisor denominado glutamato, tal como se muestra en la Figura 1.3. La capa más externa de la retina contiene células fotorreceptoras llamadas conos y bastones, y se encargan de absorber la luz para transformarla en señales eléctricas (fototransducción). Estas señales pasan a la segunda capa, que contiene las neuronas bipolares y como se observa en la Figura 1.3, cada una de ellas puede recibir la señal de más de una célula fotorreceptora para generar como salida una señal electroquímica con polaridad ON y otra señal con polaridad contraria llamada OFF. Estas señales ON y OFF pasan a la tercera capa de la retina denominada células ganglionares (RGC por sus siglas en inglés) y su función es alterar su patrón de disparo cuando cambia la información de color y luminancia del estímulo visual.

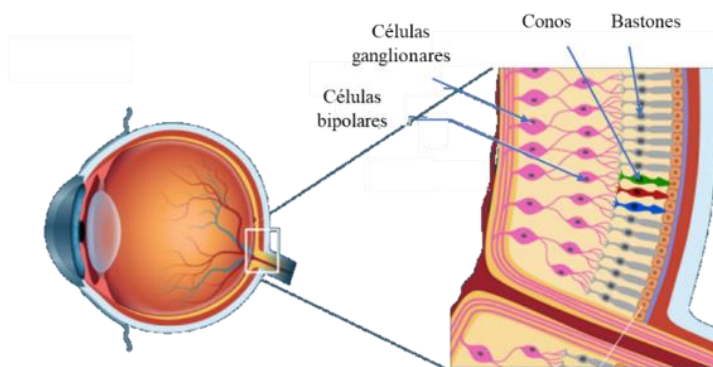


Figura 1.2. Capa de la retina y las células que la conforman.

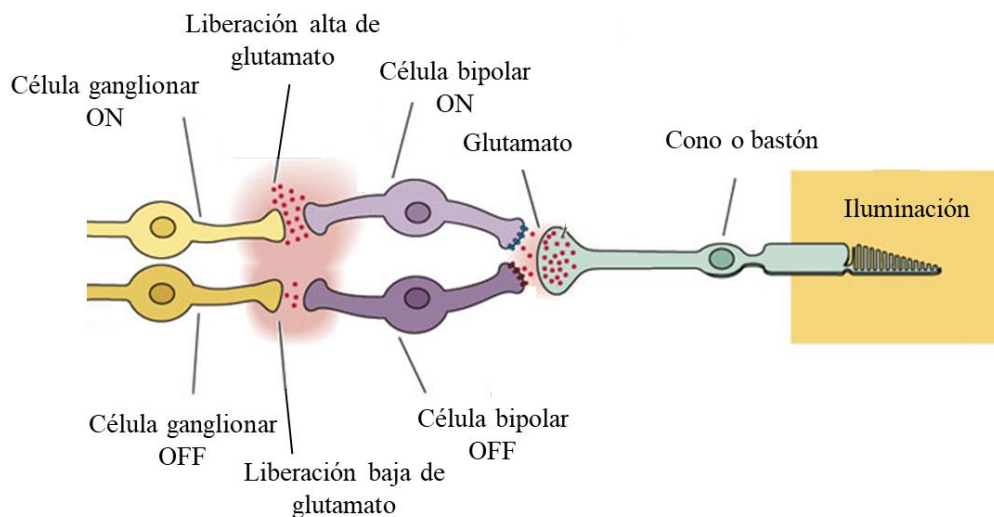


Figura 1.3. Camino que siguen las señales eléctricas de los conos y bastones hasta las células ganglionares.

Las RGC se comportan como un tipo especial de neuronas que se denominan campos receptivos (CR). Este tipo de neuronas alteran su comportamiento cuando cambia un estímulo presente en una porción del espacio visual y se extienden desde las RGC hasta la corteza visual.

Los CR se pueden representar como se muestra en la Figura 1.4, donde se tienen dos círculos concéntricos, uno pequeño en el centro y uno más amplio alrededor del centro llamado envolvente. Estos CR caen en dos categorías los ON-centro/OFF-envolvente (ON/OFF) y los OFF-centro/ON-envolvente (OFF/ON) y a su vez se pueden dividir en células que son selectivas a luminancia y color de la siguiente forma:

- Luminancia ON/OFF: células sensibles a un punto brillante sobre un fondo oscuro.
- Luminancia OFF/ON: células sensibles un punto oscuro sobre un fondo brillante.
- Verde/Rojo ON/OFF: células sensibles a un punto verde sobre un fondo rojo.
- Verde/Rojo OFF/ON: células sensibles a un punto rojo sobre un fondo verde.
- Azul/Amarillo ON/OFF: células sensibles a un punto azul sobre un fondo amarillo.
- Azul/Amarillo OFF/ON: células sensibles a un punto amarillo sobre un fondo azul.

Estos CR tienen su mayor frecuencia en el patrón de disparo de su señal de salida cuando se estimulan con el color o luminancia al que son selectivos, pero en el caso de que se presente el estímulo inverso, la salida es suprimida. Lo anterior se describe en la Figura 1.5 en la que se muestran 4 diferentes casos de estímulos de luz que afectan la respuesta de un CR ON/OFF. En

el caso a) no existe estímulo de luz, por lo que la salida es un patrón llamado base, en b) se presenta el estímulo en el centro, lo que genera la mayor frecuencia en el patrón de disparo, en c) hay estímulo en el centro como en el envoltorio, lo que genera un patrón de disparo parecido al de base, en d) el estímulo se presenta en el envoltorio pero no en el centro, lo que inhibe el patrón de disparo.

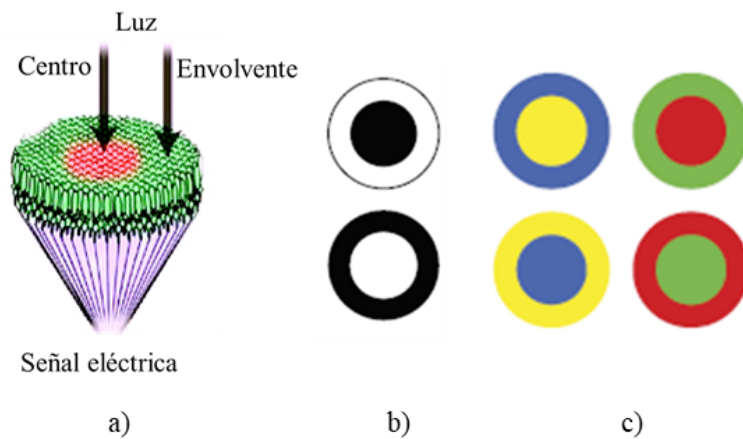


Figura 1.4. a) CR a nivel celular, b) CR de células sensibles a la luminancia, c) CR de células sensibles ante colores oponentes.

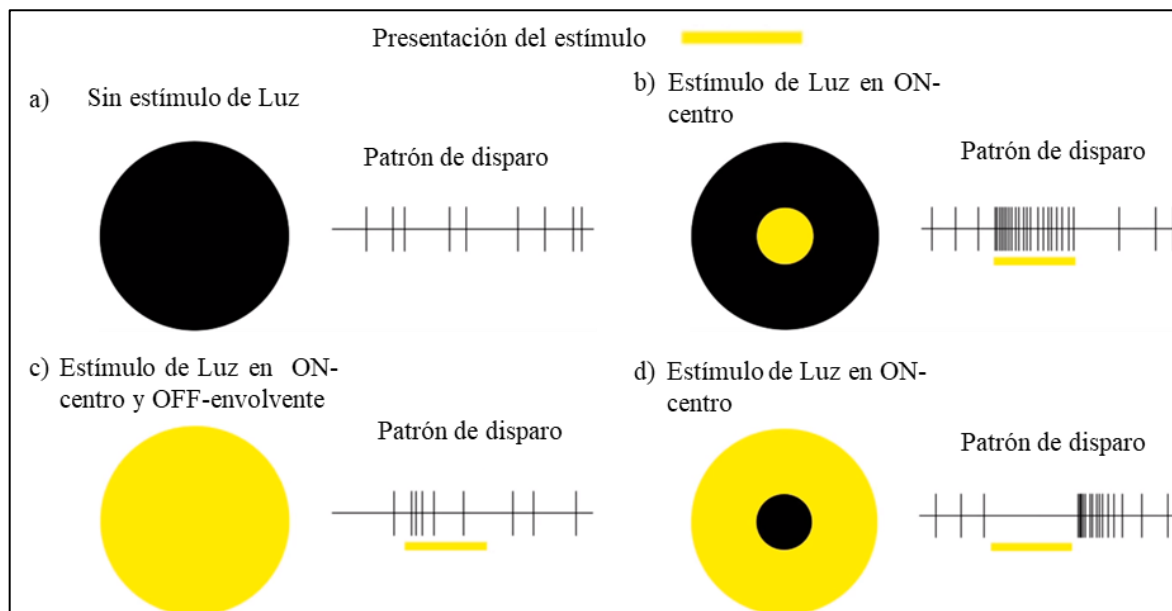


Figura 1.5. Patrón de disparo de un CR ON/OFF ante 4 diferentes estímulos.

La información visual de los RGC se transmite a través de sus axones (que forman el nervio óptico) [12][14] hacia los NGL, como se ve en la Figura 1.6. Los NGL se dividen en tres canales fisiológicos denominados parvocelular, koniocelular y magnocelular. El canal parvocelular se caracteriza por la sensibilidad al bajo contraste y tiene CR de luminancia, rojo-verde y azul-amarillo. El canal magnocelular es caracterizado por tener un área de CR más grande que el parvocelular y también tiene sensibilidad al movimiento, contornos gruesos y al alto contraste de señales de luz. La respuesta fisiológica del canal koniocelular no está bien estudiada, pero se conoce que contribuye en la obtención de información del color y del contraste de iluminación [1][15].

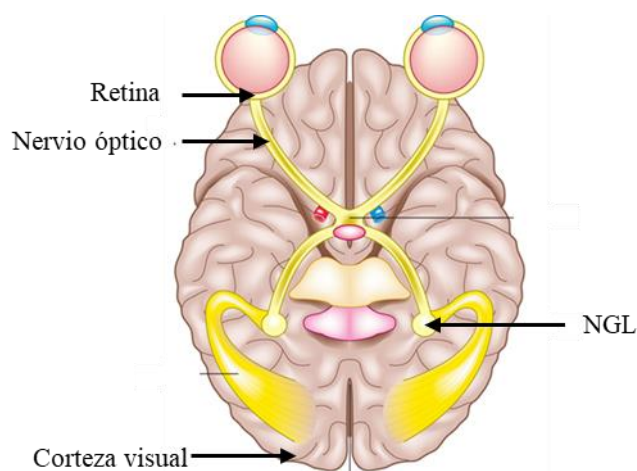


Figura 1.6. Esquema sobre la vía óptica del ser humano [1].

Una vez que la información visual cruza los NGL, esta se dirige hacia V1, que tiene CR que procesan información de estímulos visuales más complejos y se clasifican en:

- Células simples: selectivas a la posición y orientación de los estímulos visuales.
- Células complejas: selectivas a la orientación y dirección de los estímulos visuales.
- Células hipercomplejas: selectivas a la orientación, dirección y longitud del estímulo visual.

De esta manera, V1 procesa con sus CR características como bordes, arreglos cuadrículados, esquinas, movimiento, color y disparidad. V2 contiene neuronas sensibles a orientación, color, disparidad, textura y contornos ilusorios como los que generan el triángulo de Kanizsa [16]. V4 recibe información de V1/V2 y responde selectivamente a orientación,

color, disparidad, figuras simples y contornos definidos por diferencias en velocidad y/o dirección de movimiento. Estas características las combina V4 para la percepción de los objetos. El área MT sintoniza neuronas en dirección de movimiento y velocidad para interpretar el movimiento de objetos y profundidad binocular [17].

El proceso anterior es descrito en [18] como un enfoque clásico del procesamiento visual biológico llamado jerarquías corticales profundas. En la Figura 1.7 se ilustra este enfoque, donde la información empieza a fluir desde la retina a través de 2 vías paralelas, la vía dorsal y la ventral. La primera se encarga de la percepción de movimiento e inicia en las zonas magnocelulares de los NGL, para luego ir a las áreas V1, V2 y MT. La vía ventral está dedicada a la percepción de formas, incluyendo el reconocimiento de objetos, e inicia en las zonas parvocelulares, y posteriormente se dirige a las áreas V1, V2 y V4 [4][18].

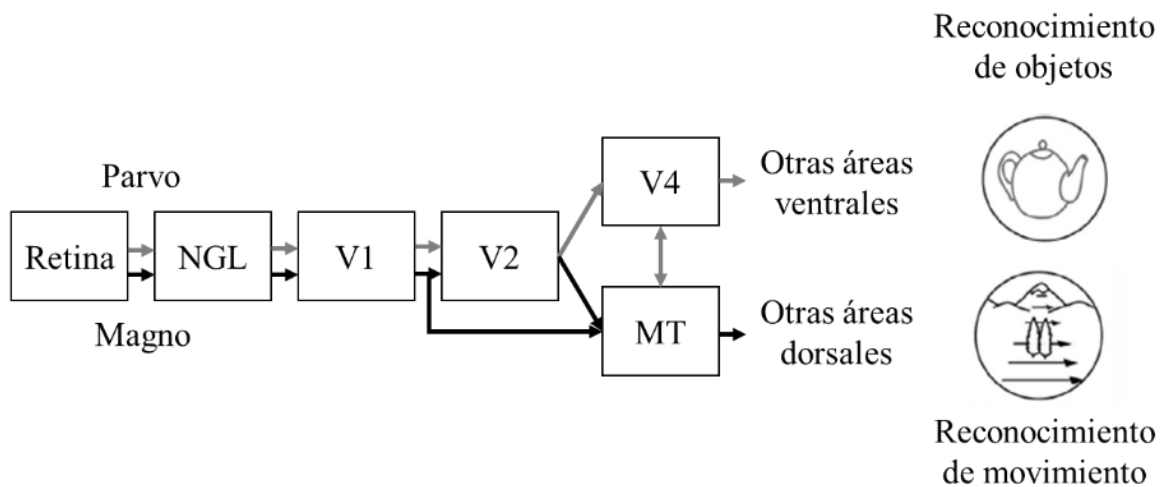


Figura 1.7. Enfoque clásico del procesamiento visual.

Existe una cantidad significativa de modelos computacionales que imitan o estudian la composición de estas capas de la corteza visual y han sido útiles en la comprensión del desarrollo de habilidades que derivan en la percepción visual. Aunado a esto, estos modelos han servido de base para proponer distintos algoritmos y redes neuronales artificiales para diversas tareas de segmentación en procesamiento digital de imágenes. Por lo tanto, para entender la relación existente entre segmentación y los modelos basados en la corteza visual, la sección siguiente presenta los conceptos elementales de segmentación basada en percepción y neurociencias.

1.2. Segmentación de objetos

La segmentación consiste en agrupar los píxeles de una imagen para formar un conjunto de regiones que representan los objetos contenidos en el escenario [19]. Existe una cantidad significativa de métodos para realizar la segmentación, pero aquellos basados en teorías de percepción se pueden dividir en: segmentación semántica, de saliencia y de agrupamiento.

1.2.1. Segmentación semántica

La segmentación semántica se basa en asignar una etiqueta de una clase previamente definida a cada uno de los píxeles de la imagen. Dicha etiqueta es el nombre o concepto de un objeto conocido dentro de una imagen [20]. Como se muestra en la Figura 1.8, los objetos segmentados son divididos en grupos representados por diferentes colores que indican una clase específica, por ejemplo, el cielo dentro de la escena se muestra en color gris. Los métodos de segmentación semántica se pueden dividir en 3 categorías: basada en regiones, redes convolucionales completas (FCN por sus siglas en inglés) y segmentación supervisada débil [21].

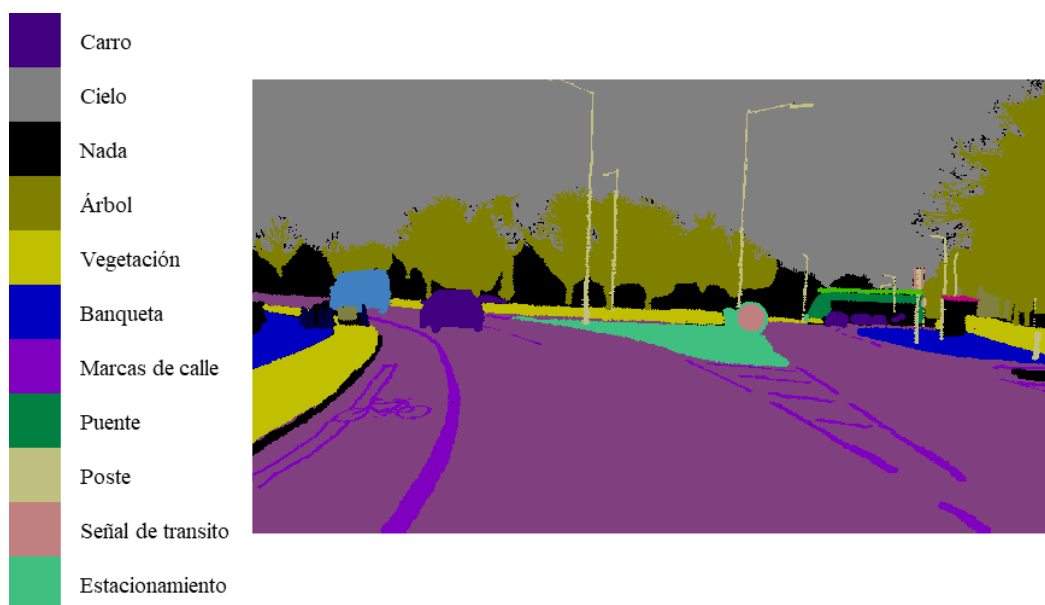


Figura 1.8. Segmentación semántica.

Segmentación semántica basada en regiones. Para esta segmentación, se extraen las regiones de una imagen y luego se clasifican con base en un conocimiento a priori de las características

de los objetos [21]. Es decir, se analiza la información de los píxeles en una imagen para agruparlos en regiones, luego, se asigna la etiqueta de cada región comparando las propiedades de estas contra las características conocidas de los objetos presentes en el escenario. Por ejemplo, Holger *et al.* propone en [22] un método basado en esta forma de segmentación con una arquitectura de redes neuronales convolucionales (CNN por sus siglas en inglés) [23]. Esta red tiene una capa diferenciable de región a píxel y una capa de región de interés (ROI) de forma libre. La capa región a píxel ignora las regiones con activaciones bajas en todas las clases (que se debe a la baja correlación de estas regiones con objetos conocidos), esto para no afectar al etiquetado final. La capa de ROI de forma libre calcula regiones considerando los píxeles que se encuentren dentro de los objetos de interés y eliminando aquellos que queden fuera. Este método tiene mejores resultados que los reportados en el estado del arte en términos de precisión con 64% en SIFT Flow y 49.9% en PASCAL. La arquitectura de esta red convolucional se representa en la Figura 1.9, donde se aíslan las regiones de la moto y el motociclista en la capa ROI y se realizan clasificaciones en la capa región a píxel.

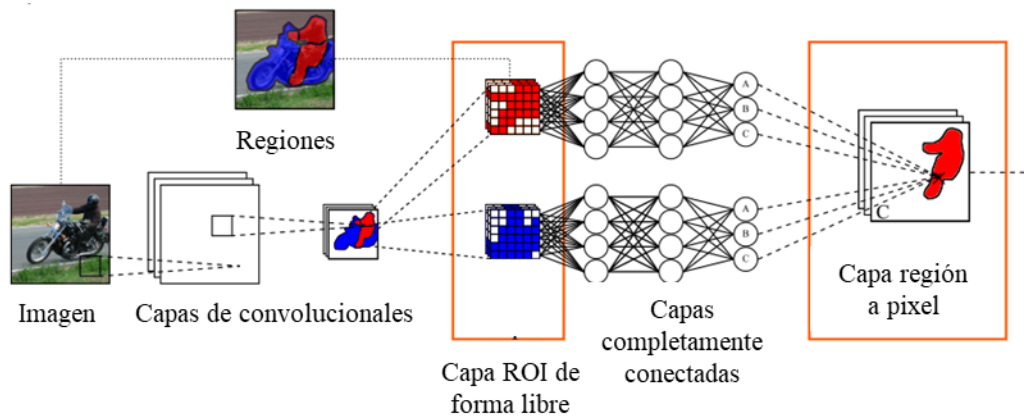


Figura 1.9. Arquitectura de una red semántica basada en regiones [22].

Segmentación semántica basada en FCN. Los métodos de segmentación basados en FCN son una extensión de las CNN, pero en este caso, las FCN remplazan la capa de clasificación (la completamente conectada) por capas de deconvolución que dividen una imagen en regiones que representan los objetos[21][24]. Un trabajo relacionado a esta técnica es el propuesto por Fu *et al.* en [25], donde se desarrolla un método basado en la FCN para la segmentación semántica de imágenes satelitales de alta resolución. Los resultados experimentales mostraron

una precisión de 85% y que el desempeño de esta red es mejor que otros métodos de segmentación. Para ilustrar este tipo de segmentación se presenta la Figura 1.10 que muestra de manera simple el funcionamiento de una red FCN para segmentación semántica.

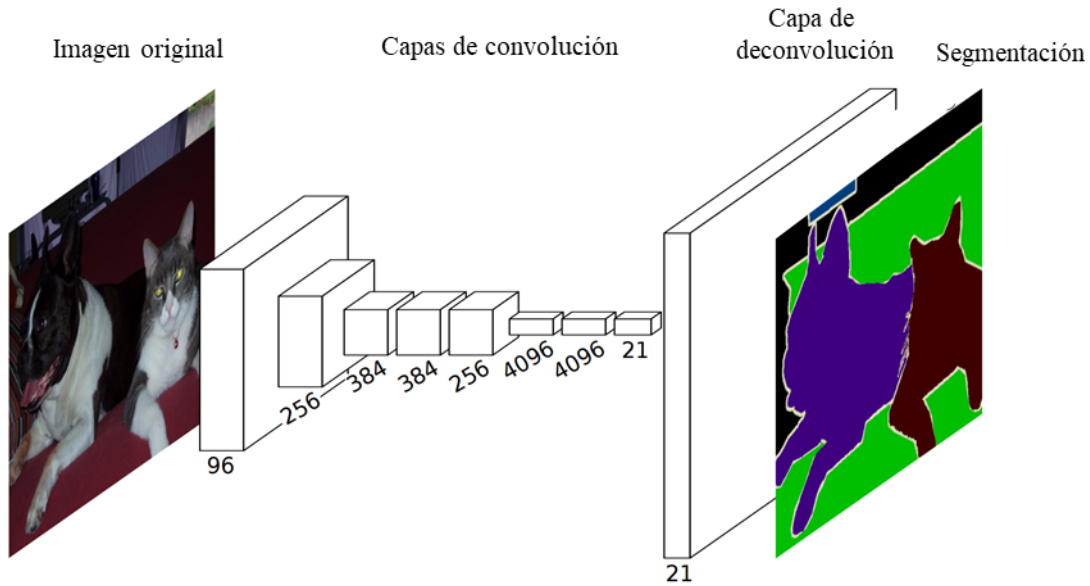


Figura 1.10. Segmentación semántica basada en FCN [24].

Segmentación semántica débilmente supervisada. Los métodos de segmentación basada en regiones y FCN requieren de un gran número de *ground truths* para entrenar la red y tener un buen desempeño. Aunque estos métodos logran excelentes resultados, la elaboración de los *ground truths* es bastante costosa ya que estos métodos requieren de miles a millones de imágenes para que sean entrenadas. Por lo tanto, se han propuesto algunos métodos débilmente supervisados que se dedican a realizar la segmentación semántica mediante el uso de cuadros delimitadores, o incluso etiquetas a nivel de imagen, no a nivel pixel como sucede en los modelos totalmente supervisados [21]. Por ejemplo, Khoreva *et al.* proponen en [26] un nuevo enfoque de segmentación semántica débilmente supervisada con CNN que consiste en generar etiquetas de entrada para el entrenamiento, utilizando anotaciones de cuadros delimitadores. Se utiliza un entrenamiento recursivo que consiste en utilizar las predicciones de una CNN de un entrenamiento previo para ser utilizadas como supervisión para la siguiente iteración. Esto permite mejorar el desempeño y reducir el ruido. Este tipo de segmentación cuenta con la

misma arquitectura mostrada en la Figura 1.10, con la diferencia en las anotaciones en los cuadros de entrenamiento.

1.2.2. Segmentación de saliencia

Saliencia es un término utilizado en procesamiento de imágenes y se refiere a la propiedad de un objeto de sobresalir de los demás presentes en un escenario o imagen. La segmentación de saliencia surge como una forma de predecir ubicaciones de escenarios donde un observador humano podría fijar su mirada. La salida de estos métodos suele ser un mapa de prominencia donde la intensidad de cada pixel de la imagen representa la probabilidad de que ese pixel pertenezca al objeto sobresaliente [27][28]. Esta salida se puede ilustrar en la Figura 1.11 donde en a) tenemos la imagen original y en b) la región saliente obtenida que pone a la flor como el objeto que sobresale en la escena.

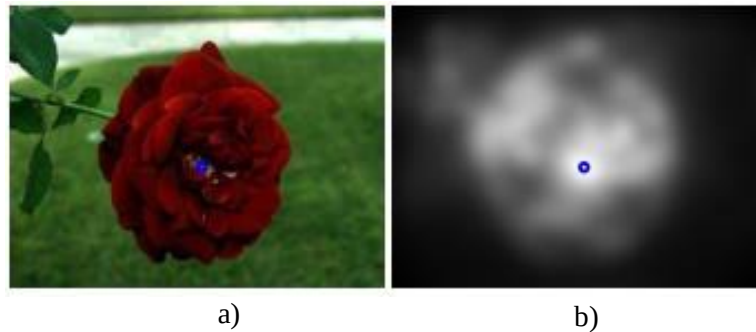


Figura 1.11. a) Imagen original, b) Imagen de saliencia respectiva.

Este tipo de segmentación tiene varias aplicaciones como compresión, segmentación de imágenes, seguimiento visual, descubrimiento de objetos, por mencionar algunos [29]. Un trabajo donde se utiliza este método es [30], donde Kruthiventi *et al.* utilizan una CNN para predecir las fijaciones oculares y segmentar los objetos salientes. Las capas de red iniciales capturan la semántica a nivel de objeto y los aspectos contextuales globales de la prominencia, mientras que las capas más profundas de la red abordan aspectos específicos a la segmentación. La red muestra una mejora sobre el estado del arte para la predicción de la fijación ocular y para la segmentación de objetos salientes en bases de datos especializadas para estas tareas como PASCAL-S y DUT-OMRON.

1.2.3. Segmentación mediante métodos no supervisados

La segmentación de objetos mediante métodos no supervisados consiste en realizar un conjunto de inferencias para agrupar los píxeles en regiones según sus propiedades descriptivas. Para realizar estas inferencias, se suelen utilizar modelos estadísticos en vez de realizar un entrenamiento previo. Dos ejemplos de estos agrupamientos son *K-means* y *fuzzy C-means* (FCM).

Aplicaciones de estos algoritmos se pueden encontrar diversos trabajos. Por ejemplo, en [31] se propone un método para la identificación exitosa de tumores cerebrales usando histogramas normalizados y agrupamiento por *K-means*. La entrada del método son imágenes del cerebro en escala de grises como se muestra en la Figura 1.12 a). Luego se utiliza *K-means* para dividir la imagen en dos grupos, generando salidas como la que se muestra en b), y posteriormente, se realizan varios procesamientos entre filtros y operaciones morfológicas para segmentar el tumor que se presenta en la Figura 1.12 c).

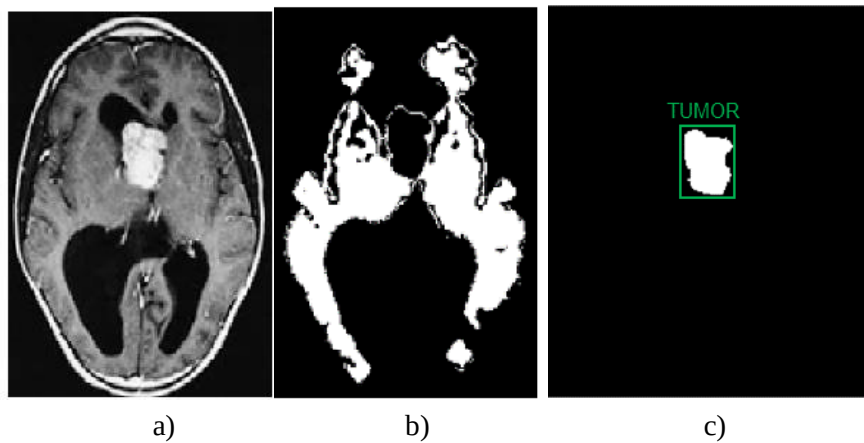


Figura 1.12. a) Imagen original, b) Segmentación *K-means* en 2 grupos, c) Detección del tumor [31].

También, en [32] se propone un método para segmentación de imágenes cerebrales basado en filtros gaussianos y agrupamiento FCM. Al igual que en el trabajo anterior, la entrada es una imagen del cerebro en escala de grises, que se aprecia en la Figura 1.13 a). El resultado es una segmentación de 3 grupos: la materia blanca mostrada en la Figura 1.13 b), la materia gris mostrada (Figura 1.13 c)) y el fluido cerebroespinal, que se ilustra en la Figura 1.13 d).

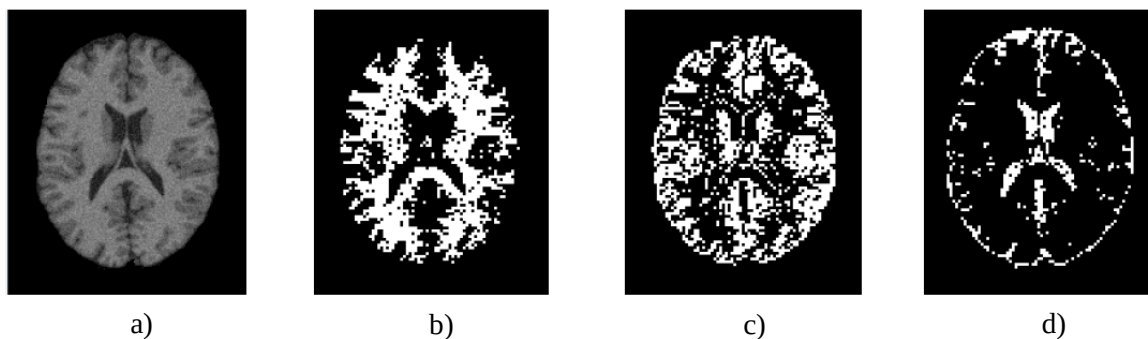


Figura 1.13. a) Imagen original, b) Materia blanca, c) Materia gris, d) Fluido cerebroespinal [32].

1.2.4. Segmentación de objetos en secuencias de video

La segmentación de video se puede ver como un problema de representar y seguir objetos en el volumen espaciotemporal y tiene numerosas aplicaciones como la recuperación de colecciones de videos, reconocimiento de actividades, etc[33]. Entre las técnicas para segmentación en video, se encuentran los modelos basados en gráficos, modelos generativos en capas y eigendecomposición como agrupamiento espectral [34].

Segmentación basada en grafos (GBS). La segmentación de video basada en grafos se basa en generar grupos de pixeles a través de la selección de bordes, donde cada píxel corresponde a un nodo de un grafo, y los bordes representan pesos que miden disimilitud entre pixeles para generar regiones que representen los objetos presentes en el video. Las regiones pueden ser jerárquicas, compuestas por otras regiones más pequeñas y espaciotemporales [34][35]. Con estas características, se puede representar y seguir los cambios de los objetos a través del tiempo. Un ejemplo de este tipo de segmentación se encuentra en el trabajo de Khoreva *et al.* en [36], donde proponen un enfoque empírico para aprender la topología de los bordes y los pesos de los grafos. Al combinar características y calibrar las puntuaciones del clasificador por su precisión, modifican la estructura del grafo seleccionando los bordes más seguros, los cuales se definen como los bordes de mayor gradiente. Su método mejora los resultados del mejor algoritmo de segmentación de video en un 6% en el punto de referencia VSB100, al mismo tiempo que reducen su tiempo de ejecución en un 55%, ya que el grafo obtenido es mucho más simple.

Modelos generativos en capas. Este método consiste en modelar funciones de características para representar el comportamiento de los píxeles. Las funciones de características se modelan en forma a priori con las propiedades de los objetos que se van a detectar. Luego, para predecir qué píxeles corresponden a un objeto, se busca la función que haya tenido el puntaje máximo para la información de cada píxel [34][37]. Un ejemplo de este tipo de segmentación lo realiza Johnander et al. en [37], donde se propone una arquitectura de red que aprende una representación de la apariencia de los objetos. El módulo de apariencia aprende un modelo probabilístico generativo de distribución de características para los objetos. Dada una nueva imagen, predice las probabilidades por clase, proporcionando una alta discriminación que se procesa en módulos de red posteriores. Se demuestra que el enfoque propuesto supera los enfoques publicados en la base de datos de YouTube-VOS.

Agrupamiento espectral. El agrupamiento espectral radica en transformar las características de los píxeles de un espacio cartesiano (cuadro de video) a otro espacio diferente para realizar el agrupamiento. La transformación del espacio reduce el tamaño de las características y en términos generales, las preagrupa en dimensiones ortogonales. Este preagrupamiento no es lineal y permite geometrías no convexas conectadas arbitrariamente, que es la principal ventaja del agrupamiento espectral porque esto permite realizar agrupamientos de píxeles que no estén distribuidos de manera geoméricamente convexa [34][38]. Este método lo utiliza Li et al. en [39], donde presentan un algoritmo de segmentación de superpíxeles (grupo de píxeles) llamado agrupamiento espectral lineal, que produce superpíxeles compactos y uniformes con bajos costos computacionales. Básicamente, se adopta una metodología basada en una métrica de semejanza que mide la similitud de color y la proximidad del espacio entre los píxeles de la imagen. Sin embargo, se aproxima la métrica de semejanza utilizando un *kernel* que conduce a un mapeo explícito de valores de píxeles y coordenadas a un espacio de características de alta dimensión. Posteriormente, se utiliza *K-means* para realizar el agrupamiento en el nuevo espacio. Consiguen con este algoritmo buen desempeño en similitud de bordes y de estructura de imagen global.

1.3. Estado del arte de segmentación neuroinspirada de imágenes y video

Existe en la literatura una gran cantidad de algoritmos neuroinspirados aplicados a visión, y en este trabajo se analizarán aquellos modelos que se basan en el proceso de percepción visual humano [1]. A continuación, se mencionan trabajos en los cuales los autores proponen métodos para la segmentación de imágenes o video basados en modelos neuroinspirados.

Zhou *et al.* simplifican en [7] los parámetros de una red PCNN. Esta red es un modelo que se aplica en segmentación de imágenes y su funcionamiento simula la sincronización/desincronización de neuronas en la corteza visual de pequeños mamíferos. Para ello, se eliminan los factores de retardo exponencial, se reduce la cantidad de parámetros en el cálculo del umbral y se introduce un método de FCM para agrupar píxeles con similitud y proximidad espacial. Los resultados experimentales en imágenes de infrarrojos muestran que el modelo propuesto tiene un alto desempeño en la segmentación y con mejor adaptabilidad que métodos existentes basados en PCNN tradicional.

Hu *et al.* presentan en [1] un método para segmentación de imágenes en escenarios naturales que se compone de un sistema jerárquico que imita la organización del área visual temprana en la corteza visual de primates. Se propone una metodología que consiste de dos etapas: la primera es una estructura modular que incluye tres características para segmentar: color, forma y textura. Después, se aplica una operación llamada segmentación de fusión de características (MFFS) para integrar las 3 características a través de una red neuronal de retropropagación (BPNN). En la última etapa, se simula la operación de la corteza visual primaria. Este modelo se utilizó para segmentar imágenes en dos regiones y se probó su desempeño en la base de datos Caltech101 y WWW, mostrando un buen desempeño.

González-Casillas *et al.* proponen en [40] un modelo para el reconocimiento visual que se asemeja a la forma en que el cerebro humano interactúa para lograr el proceso de identificación de objetos. El modelo describe el procesamiento en V1 y V2 para extraer características de bordes, orientación y contornos. También describe V4 y un proceso de comparación de plantillas en la corteza inferotemporal (ITC) que utiliza información visual temprana de baja frecuencia espacial para sesgar las comparaciones disponibles. La propuesta pretende ser la base para tratar la información visual en una arquitectura cognitiva más amplia.

Avila-Mora *et al.* desarrollan en [2] un método neuroinspirado para la segmentación de objetos en video basado en el modelo Conepvm, que describe la percepción de objetos en movimiento en la corteza visual. El modelo utiliza filtros Gabor para detectar el movimiento mediante la estimación de características de los objetos del escenario como velocidad global, dirección y trayectoria. También utiliza mapas autoorganizados de Kohonen (SOM) para definir cuántos objetos en movimiento se encuentran en la secuencia. Aplicaron sus algoritmos a dos secuencias de video, donde se obtuvo un desempeño de 4 objetos detectados de 5 para el primer video y 7 objetos detectados de 8 en el segundo video. Como conclusión, se menciona que la inicialización de los pesos sinápticos juega un papel importante en la forma en que la red toma decisiones acerca de qué pixel activa cual neurona en la red SOM.

Gao *et al.* proponen en [41] un modelo para detección de saliencia dinámica inspirado en el procesamiento en la corteza visual primaria. Se calculan las características espacio temporales de las secuencias de video a través de filtros Gabor 3D para simular la respuesta de neuronas ante diferentes estímulos en sus campos receptivos. Adicionalmente, se introduce el agrupamiento perceptual para integrar características visuales significativas y eliminar las que son distractoras. Las interacciones entre neuronas son simuladas con convolución y una función de activación tipo rectificador. La evaluación de desempeño se realizó con la base de datos ASCMN y los resultados muestran que su modelo tiene un desempeño aceptable en las métricas de *receiving operating characteristics curve* (ROC) y *Correlation coefficients* (CC), mientras que en la métrica *Normalized Scanpath Saliency* (NSS) se reportan buenos resultados. Se concluye que su modelo se desempeñó mal para videos con movimiento debido a la complejidad del fondo.

Krüger *et al.* revisan en [17] algunos principios y estructuras funcionales de la corteza visual de los primates para impulsar la investigación de la visión artificial. En esta revisión se presentan los principios de las jerarquías de procesamiento del sistema visual de primates, que se caracterizan por una secuencia de diferentes niveles de procesamiento como lo son: la retina, los NGL, y las áreas de la corteza visual V1, V2, V4 y MT, por mencionar los más importantes. Lo anterior constituye una jerarquía profunda que contrasta con las arquitecturas de visión plana que se utilizan en visión por computadoras.

Vacher *et al.* proponen en [42] un modelo probabilístico generativo para segmentación visual que combina el conocimiento sobre:

- La sensibilidad de las neuronas en la corteza visual a las regularidades estadísticas en imágenes naturales.
- La preferencia de los humanos para formar particiones contiguas del espacio visual.

A partir de estos puntos, se desarrolla un algoritmo para entrenamiento e inferencia basado en la maximización de expectativas y se valida en datos sintéticos. Las probabilidades de asignación de píxeles a las segmentaciones de los objetos es la suma de la evidencia (información estadística de píxeles locales) y anteriores (asignaciones de píxeles vecinos). El modelo se probó en la base de datos de imágenes BSD, y aunque no se alcanzan los resultados de los mejores métodos en el estado del arte, su desempeño es aceptable. Los autores justifican este bajo desempeño en que su enfoque principal es proporcionar un marco probabilístico para estudiar la segmentación humana en vez de maximizar el rendimiento.

Yang *et al.* en [43] proponen un nuevo modelo de PCNN heterogénea (HSPCNN) para segmentación de imágenes. HSPCNN segmenta la imagen con base en los niveles de gris de cada píxel y se construye con varios modelos de PCNN simplificados con diferentes parámetros correspondientes a diferentes tipos de neuronas. Se utilizó la base de datos BSD 300 y los resultados mostraron que modelo tiene un mayor desempeño en la segmentación comparado con otros algoritmos que utilizan SPCNN y PCNN.

CAPÍTULO II

SEGMENTACIÓN DE OBJETOS MEDIANTE AGRUPAMIENTO

El esquema más común para desarrollar un algoritmo de visión por computadora neuroinspirado es analizar los modelos computacionales que expliquen el funcionamiento de la corteza visual y posteriormente, generar un método de procesamiento digital de imágenes. Sin embargo, también se puede explorar el uso de interfaces cerebro computadora (BCI por sus siglas en inglés) para adquirir señales que midan la actividad cerebral cuando un sujeto percibe y reconoce objetos. Basado en este último esquema, en este capítulo se presenta el desarrollo de un algoritmo de segmentación de objetos inspirado en los resultados de una serie de experimentos para medir la actividad cerebral cuando un sujeto se expone al reconocimiento de objetos. Consecuentemente, este capítulo se divide en dos secciones: la sección 2.1 reporta un experimento para observar el comportamiento del cerebro ante ciertos estímulos. Con base en los resultados de dicho experimento, se diseñó un método para segmentación de objetos denominado DObjA y se presenta en la sección 2.2.

2.1. Detección de objetos y electroencefalografía

Una BCI mide actividad cerebral y la transforma en señales digitales para que una computadora las pueda interpretar. La técnica más popular para medir dicha actividad es la electroencefalografía (EEG), ya que no es invasiva y es menos costosa que las demás técnicas. La EEG obtiene actividad cerebral con electrodos que se colocan en el cuero cabelludo y se componen de la suma de actividades eléctricas de las poblaciones de las neuronas, con una pequeña contribución de las células gliales [44]. Las BCI basadas en EEG se aplican en el desarrollo de sistemas de control, rehabilitación, terapia y el desarrollo de modelos para entender el funcionamiento del cerebro humano como:

- *La codificación anticipatoria de la posición de un objeto.* Hogendoorn *et al.* realizan en [45], una clasificación de patrones multivariados a los datos de EEG para investigar la codificación anticipada de la posición de un objeto en humanos. Para comparar el curso temporal de la actividad neural que representa la posición de un objeto en movimiento, se aíslan los mecanismos anticipatorios, que son predicciones que realiza el cerebro ante un

estímulo con un patrón de trayectoria conocido, que podrían compensar los retrasos de procesamiento neurales. Se muestra una anticipación de latencia para el movimiento predecible de aproximadamente 16 ms. Los resultados son numéricamente consistentes con trabajos anteriores con animales, sugiriendo que los modelos actuales de codificación predictiva espacial en la corteza visual pueden extenderse efectivamente al dominio temporal.

- *La discriminación de eventos objetivo contra distractores en secuencias de video.* Rosenthal *et al.* analizan en [46] si una neuroimagen no invasiva puede ser usada para detectar correlatos neuronales, que se definen como la relación entre actividad cerebral y el reconocimiento de eventos de importancia presentados en videos en directo. Las señales EEG se adquieren con un sistema BioSemi ActiveTwo y un casco de 32 electrodos. Las señales obtenidas se utilizaron para entrenar los clasificadores que tienen como entrada características temporales y espaciales para diferenciar entre eventos ocurridos en los videos. Los resultados reportaron que la actividad neuronal permite discriminar entre eventos objetivo contra los distractores generados por el comportamiento del sujeto dentro de las secuencias de video simuladas. Esta actividad discriminativa se debe a la respuesta evocada y no a una actividad oscilatoria.
- *El conocimiento de la activación neuronal de las áreas visuales ante estímulos con curvaturas convexas.* Larson *et al.* desarrollan en [47], un experimento para conocer la actividad de capas corticales del área MT cuando un sujeto se somete a estímulos visuales de formas geométricas como curvaturas convexas como la figura ‘V’. Las señales EEG fueron obtenidas con un escáner 3-T GE Sigma EXCITE con 8 canales y posteriormente se analizó el potencial P300. Se concluye que el área MT tiene una mayor actividad en presencia de curvaturas convexas ya que se interpretan como amenazas, sugiriendo que el reconocimiento de peligro puede basarse en señales visuales muy simples y sin contexto.
- *La comprensión del reconocimiento visual de objetos.* Contini *et al.* revisan en [3], la contribución de los métodos de decodificación de series temporales de EEG para comprender cómo el cerebro humano reconoce objetos visuales. Se menciona que de acuerdo con la comprensión actual de la jerarquía de procesamiento visual, las

características visuales de bajo nivel dominan las representaciones de objetos decodificables en las señales EEG.

De acuerdo con estos trabajos, las señales EEG se pueden utilizar en conocer la actividad cerebral cuando un sujeto se expone a experimentos de reconocimiento de objetos. De esta manera, también se realizó en esta tesis un experimento para conocer la respuesta de la corteza visual ante diferentes estímulos que fueron mostrados a un sujeto de pruebas. Como se muestra en la Figura 2.1, para este experimento se utilizó un esquema que consta de cuatro elementos: estímulo visual, sujeto de pruebas, sistema de adquisición y procesamiento. Cada una de estas etapas se describirá en las siguientes subsecciones.

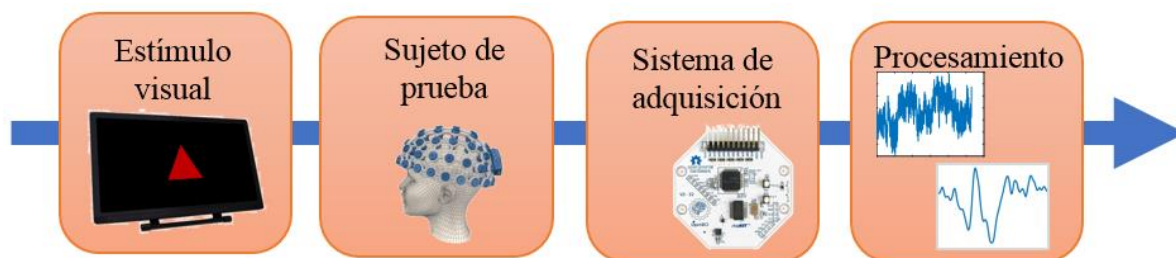


Figura 2.1. Etapas para la realización del experimento de detección de objetos mediante EEG.

2.1.1. Estímulo visual y esquema de experimentación

Los estímulos visuales y el esquema de experimentación fueron seleccionados con base en el trabajo de [48], donde se realizó un experimento en el que un sujeto de pruebas veía imágenes de círculos azules y cuadrados rojos, con la finalidad de conocer que figura estaba observando basándose en sus patrones de activación cerebrales. Las imágenes en el experimento fueron simples figuras geométricas para evitar la variabilidad en color y texturas que se encuentran en imágenes más complejas, lo que complicaría el análisis de los patrones de activación cerebrales. Para este análisis se utilizó una metodología basada en el potencial evocado P300.

Consecuentemente, en este experimento se decidió utilizar un esquema basado en la detección de objetos mediante los potenciales relacionados a eventos (ERP) con deflexiones de 300 ms, que también se le conoce como señal P300. Para que se genere esta señal, es necesario que el sujeto de pruebas fije su atención solo en un estímulo que debe aparecer en forma aleatoria, tal como lo plantea el paradigma *Oddball* [49]. Los estímulos utilizados en el

experimento son triángulos equiláteros de 4 cm de altura y cuadrados de 4 cm por lado, ambas formas geométricas en colores rojo y azul para abarcar las características de color y forma (bordes), tal como se muestran en la Figura 2.2.



Figura 2.2. Estímulos visuales experimentales.

Se generaron seis secuencias diferentes para generar el P300 utilizando como estímulos las formas geométricas de la Figura 2.2, y en cada secuencia aparecen de manera aleatoria dos de estas formas, como se muestra en la Figura 2.3. Las formas asignadas en cada secuencia miden lo siguiente:

Secuencias 1 y 2. Cambio de figura en objetos del mismo color.

Secuencia 3 y 4. Cambio de figura y color en objetos.

Secuencia 5 y 6. Cambio de color en objetos con la misma figura.

Antes de iniciar una secuencia, se le pide al sujeto que seleccione una de las formas de la secuencia y cuente las veces que esta aparece.

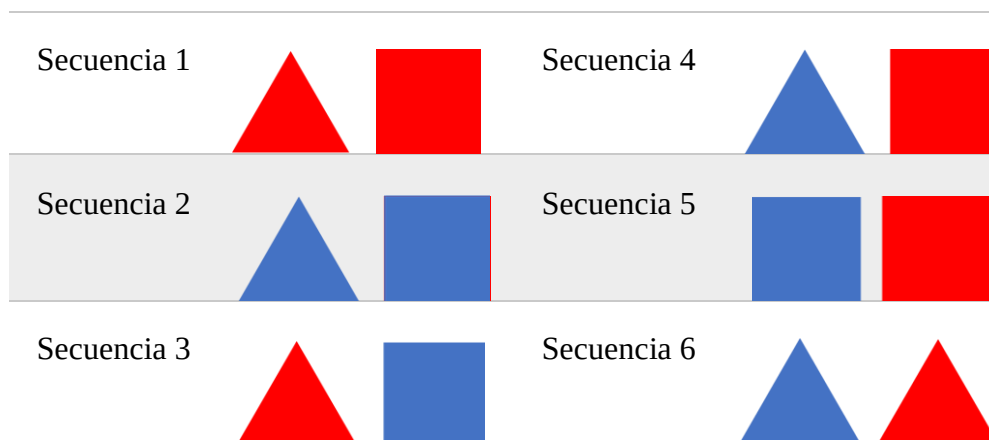


Figura 2.3. Par de figuras para cada secuencia.

Las secuencias tienen un diagrama de tiempo como se muestran en la Figura 2.4 y a continuación se describen brevemente: se inicia con un tiempo de 4 segundos sin actividad para

estabilizar la señal, luego se muestra un estímulo en un intervalo de 100 ms y después se tiene un intervalo de 400 ms sin estímulo. Este último intervalo permite que la señal P300 del sujeto de prueba se alcance a presentar antes de un estímulo nuevo. Las formas geométricas de las secuencias se muestran de manera aleatoria 15 veces cada una.

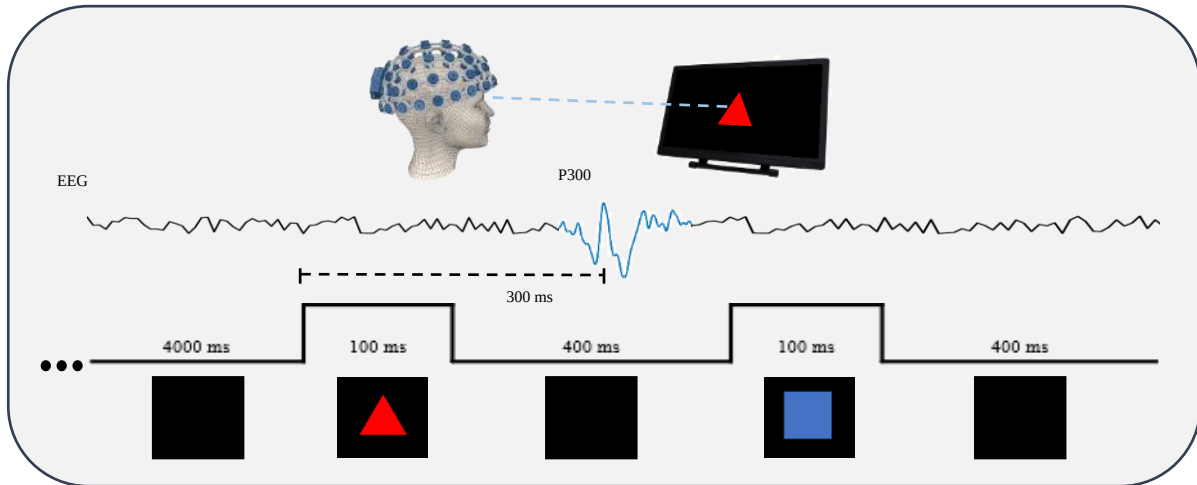


Figura 2.4. Presentación de las secuencias.

2.1.2. Sujeto de pruebas

Las características del sujeto de pruebas se basaron en el esquema propuesto en [50], donde se estableció un criterio de selección para realizar pruebas muy similares a las realizadas en este trabajo. Este criterio establece que para trabajos de EEG que son exploratorios se deben seleccionar sujetos entre 22 y 26 años, diestros y con visión normal o corregida. Por lo tanto, fue seleccionado un hombre sano de 25 años con visión corregida y diestro para los experimentos desarrollados en esta tesis. El sujeto de este experimento fue informado del propósito del estudio, participó bajo su propia voluntad y ya había estado en experimentos similares. Esto último es importante ya que un sujeto con experiencia define mejor el P300 en experimentos que son exploratorios (en este trabajo no se está desarrollando un sistema de control). Cabe mencionar que utilizar un solo sujeto es válido para estudios exploratorios como se muestra en el trabajo [51], en el que se apoyan de un solo sujeto de pruebas para evaluar el rendimiento de una interfaz cerebro-computadora utilizando diferentes configuraciones de electrodos para determinar el uso más eficiente de un dispositivo de adquisición de señales EEG que está limitado a 8 canales. El uso de un sujeto de pruebas evita tener que considerar la

incertidumbre de utilizar varios, y como en este caso no se busca desarrollar un sistema de control o realizar estudios clínicos, sino más bien conocer la actividad más primitiva en la corteza visual, se obtienen las señales de un solo sujeto que este muy preparado en BCI.

De acuerdo con el protocolo desarrollado a lo largo de la toma de muestras, el sujeto de pruebas debe de encontrarse en un estado neutro física y emocionalmente. Para evitar artefactos en la señal EEG, el sujeto no debe realizar movimientos extra al que el encargado de la toma de muestras le indique, en especial movimientos de hombros y de cuello. Adicionalmente, el sujeto debe posicionar su vista a una distancia de 55 cm de la pantalla de los estímulos [52][53]. También es importante definir las condiciones de la habitación donde se va a realizar el experimento. Estas condiciones se basan en los trabajos de [50][48][53] y la finalidad de que sean controladas es que las muestras no se vean afectadas por artefactos que disminuyan la calidad de las señales. Las condiciones para el ambiente de pruebas se describen a continuación:

- La habitación debe contar con una iluminación baja (5 luxes, aproximadamente).
- No contar con luces parpadeantes dentro del campo de visión del usuario.
- Sin elementos distractores.
- Contar con una silla con descansa brazos para el sujeto de pruebas.
- La temperatura de la habitación debe ser agradable para el sujeto de pruebas. Los valores pueden oscilar entre los 22 y 26 °C, aunque no es estricto.

En la Figura 2.5 se muestra la habitación en la que se toman las muestras, donde la iluminación es baja y no cuenta con elementos distractores. También se muestran tres elementos importantes de la habitación: silla con descansabrazos, interfaz con estímulos y equipo para toma de muestras. Adicionalmente, enfrente de la silla se encuentra la pantalla que contiene la interfaz donde aparecerán los estímulos. Durante la toma de muestras es importante verificar que no se presenten artefactos, para esto se utilizó una segunda pantalla donde se grafican en tiempo real las señales de los 8 electrodos mientras se tomaban las muestras.



Figura 2.5. Habitación donde se realizan las tomas de muestras.

2.1.3. Sistema de adquisición

Los electrodos utilizados en este experimento (Figura 2.6) están compuestos por puntas de silicona recubiertos de Ag/AgCl [54][55] y forman parte del sistema de adquisición OpenBCI®. Estos son secos y atraviesan ligeramente la epidermis para hacer contacto directo con el cuero cabelludo.



Figura 2.6. Electrodo secos del sistema de adquisición OpenBCI® [55].

Las posiciones de los electrodos en el cuero cabelludo registran las actividades corticales de las regiones del cerebro que están cerca de ellas, y un estándar para estas posiciones es el sistema internacional 10-20. El término “10-20” se refiere a que la posición de los electrodos

tiene un 10% o 20% de la distancia total entre las ubicaciones especificadas en el cráneo. La nomenclatura de las posiciones se basa en una letra para indicar el lóbulo del cerebro que se explora y un número para indicar el área del hemisferio. Las letras F, P, T, O y C están relacionadas con las áreas frontal, parietal, temporal, occipital y central, respectivamente. Los números par están asociados con el lado izquierdo y los impares con el lado derecho del cerebro. La letra z se usa para indicar que la ubicación del electrodo en el cuero cabelludo cae a lo largo de la línea central que corre entre el nasión y el inión. FP1 y FP2 están respectivamente relacionados con los polos izquierdo y derecho de la frente. También A1 y A2 son regiones en las orejas que son sitios comunes para la colocación de los electrodos de referencia y tierra [56].

Para posicionar los electrodos correctamente, se utilizó el casco Ultracortex Mark IV, que es un casco con 35 posiciones diferentes creado con la intención de trabajar con cualquier tarjeta desarrollada para OpenBCI®. De las posiciones disponibles para los electrodos, se utilizan para el experimento los siguientes: T3, T5, PO3, O1, O2, PO4, T6, T4, y se muestran encerrados en rojo en la Figura 2.7. Estos electrodos se encuentran en las áreas occipital y temporal del cerebro, cubriendo las áreas visuales V1, V2, V4 y V5.

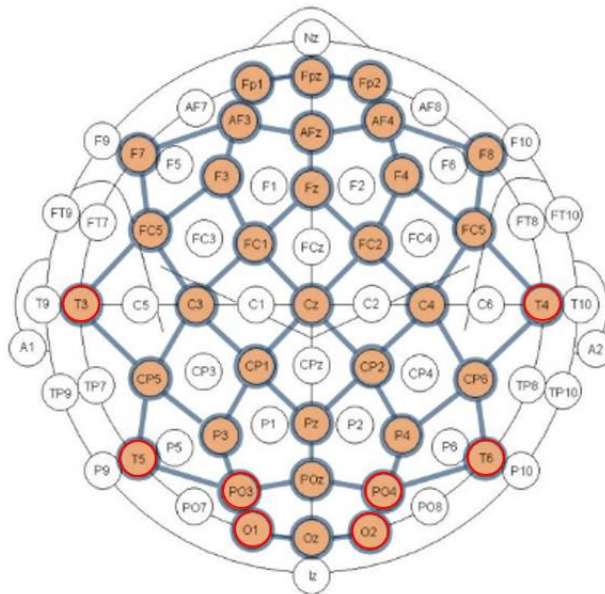


Figura 2.7. Electrodos utilizados del sistema internacional 10-20.

El segundo componente es la tarjeta de adquisición de señales EEG que lleva el nombre de Cyton. Este dispositivo cuenta con un conversor analógico-digital con una frecuencia de muestreo

por canal de 250 Hz, 8 canales de entrada para lectura de señales EEG y una resolución por canal de 24 bits.

La Figura 2.8 muestra el dispositivo de toma de muestras, en el que se encuentran los electrodos que son los elementos de color azul que se indican con flechas amarillas. En total son 8 electrodos, más 1 electrodo de tierra en la posición Fz, el cual se indica como “referencia”. En la parte de la izquierda de la figura se encuentra el dispositivo Cyton de OpenBCI® al que le llega la información de los electrodos. La información obtenida por la tarjeta se envía a una computadora a través de comunicación por bluetooth. Las señales enviadas a la computadora se guardan en archivos de Matlab para posteriores procesamiento.

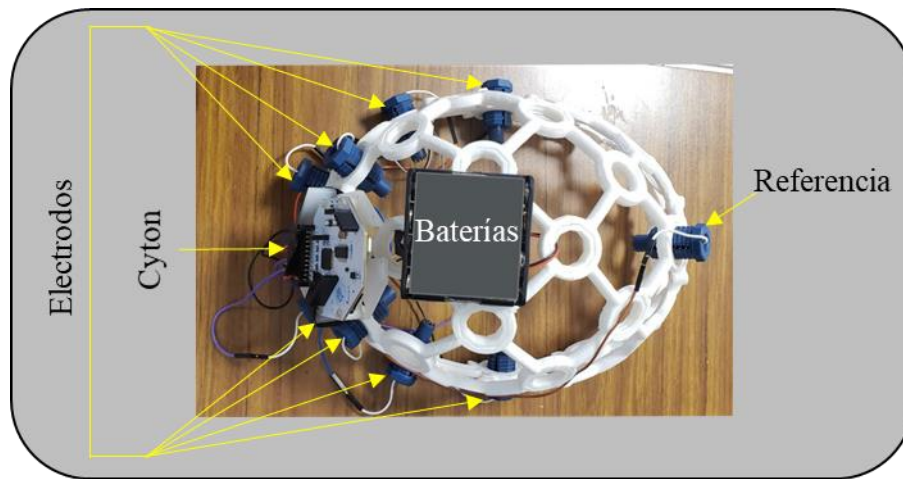


Figura 2.8. Dispositivo de toma de muestras.

2.1.4. Procesamiento

Esta etapa interpreta las señales EEG de las 6 secuencias del experimento para determinar si se presentó la onda P300, y se divide en preprocesamiento y extracción de características. La entrada son las 6 secuencias, cada una con las señales EEG compuestas de 8 canales (los ocho electrodos posicionados en las zonas occipital y temporal). Estas señales se definen como $s_p^{n_e}(n_s)$ donde p es el índice del electrodo $p = 1, 2, \dots, 8$, n_e el índice de cada secuencia $n_e = 1, 2, \dots, 6$, y n_s es el índice de cada una de las muestras tomadas $n_s = 1, 2, \dots, 4750$. Las 4750 muestras se generan debido a que el tiempo por sesión es de 19 segundos y la frecuencia de muestreo es 250 Hz.

El preprocesamiento es un filtro H pasa bandas Butterworth de tercer orden con frecuencias de corte de 1 y 15 Hz, este rango de frecuencias fue seleccionado porque es el ancho de banda de la señal de P300 [57]. Aplicando este filtro se obtiene la señal:

$$H: s_p^{ne}(n_s) \rightarrow s_{fp}^{ne}(n_s) \quad (2.1)$$

Una vez teniendo las señales filtradas, se procede a realizar la etapa de extracción de características. En esta etapa se realiza un agrupamiento que consiste en tomar 15 fragmentos de un segundo de cada señal $s_{fp}^{ne}(n_s)$, donde cada fragmento corresponde a las 15 veces que se presenta el estímulo al que el sujeto le presta atención en cada secuencia. De esta manera, se define un arreglo $R_{p,i}^{ne}(m)$, que contiene cada uno de los fragmentos de un segundo, siendo $i = 1, \dots, 15$ el número de fragmento. Finalmente, se realiza un promediado dado por:

$$x_{sp}^{ne}(m_s) = \frac{1}{15} \sum_{i=1}^{15} R_{p,i}^{ne}(m_s) \quad m_s = 1, 2, \dots, 250 \quad (2.2)$$

donde m_s es un índice para las muestras de cada fragmento. De esta manera, la señal $x_{sp}^{ne}(m_s)$ es un arreglo de 6 promedios de los 15 fragmentos de cada señal EEG de los 8 canales, cada uno con 250 muestras. Se espera que en cada señal de $x_{sp}^{ne}(m_s)$, contenga la señal P300 ya que se están promediando los intervalos donde se supone que se presentó el estímulo al que se le puso atención. En la Figura 2.9 se muestra un ejemplo gráfico del vector $x_{sp}^{ne}(m_s)$ de un canal, y como se puede apreciar en este caso, la señal P300 mostró su pico máximo un poco después de los 300 ms.

Una vez procesados los fragmentos de las señales de las 6 secuencias, se realizó un análisis visual que consistió en detectar si se presentaron deflexiones alrededor de los 300 ms en las señales de $x_{sp}^{ne}(m_s)$. Los resultados se muestran en la Tabla 2.1 como razones de número de señales P300 detectadas sobre el número total de señales. Para verificar la reproducibilidad del experimento se realizaron 3 pruebas definidas como P1, P2 y P3, cada una de ellas consistió en correr las 6 secuencias de estímulos mostradas en la Figura 2.3. La cantidad de señales son 12, ya que se realizó un experimento por cada una de las 2 figuras de las 6 secuencias.

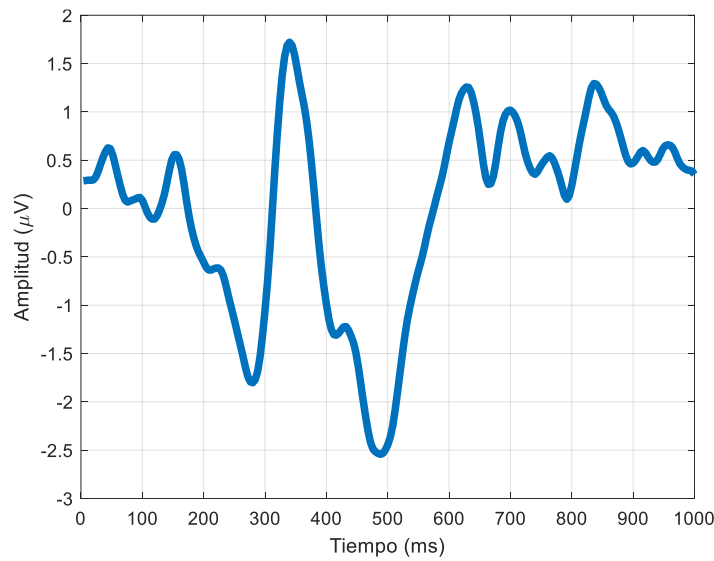


Figura 2.9. Señal P300 extraída de un canal del vector $x_{sp}^{n_e}(m_s)$.

Tabla 2.1. Cantidad de señales P300 detectadas.

Prueba	T3	T5	PO3	O1	O2	PO4	T6	T4
P1	2/12	10/12	9/12	11/12	8/12	10/12	11/12	1/12
P2	2/12	4/12	4/12	3/12	3/12	3/12	3/12	3/12
P3	3/13	3/12	2/12	3/12	4/12	3/12	4/12	3/12
Promedio	2.33/12	5.67/12	5/12	5.67/12	5/12	5.34/12	6/12	2.33/12

2.1.5. Conclusiones del experimento EEG

Se puede observar en la Tabla 2.1 que no siempre se generó el P300, y eso se debe a que no todas las áreas de la corteza visual reaccionan con estímulos de forma y color. No obstante, se puede ver que en las tres pruebas realizadas, las posiciones T5, T6 y O1 son las que presentaron una mayor cantidad de P300. De acuerdo con [40], las áreas T5 y T6 reflejan actividad de V4, quien se encarga de la percepción de objetos, mientras que O1 refleja actividad de V1 y V2 relacionada a percepción de color y bordes. El electrodo PO4 también refleja actividad de V1 relacionada a percepción de bordes.

En las tres pruebas, las señales P300 se presentaron principalmente en las secuencias 1, 2, 3 y 4, que tienen en común la variación de la forma de la figura, por lo que la característica de color no fue estrictamente necesaria para el reconocimiento de los objetos. En cambio, la

característica de forma tuvo consistencia en el reconocimiento de las figuras, que se puede conseguir con la componente de luminancia y los bordes.

De acuerdo a los modelos de LAMINART [6] y PG-LISSOM [6], las neuronas de V1 sensibles a orientación responden en función de los bordes de los objetos y V2 recibe dichos bordes para separarlos en varios niveles de contornos. Además, de acuerdo a los modelos de Wilson & Cowan [6] y LAMINART, V4 toma como información dichos contornos y los compara contra contornos formados en memoria para crear grupos de neuronas que oscilan en diferentes frecuencias en la banda gamma para generar conjuntos corticales que representan los objetos. Al analizar los modelos de PG-LISSOM, LAMINART y Wilson & Cowan, se puede asumir que una forma simplificada de representar la percepción de objetos es tomar la información de luminancia de un cuadro y detectar bordes en distintos niveles para definir los grupos de píxeles que representarán los objetos.

Adicionalmente, los modelos de PG-LISSOM y Wilson & Cowan explican que la percepción de objetos se debe a un fenómeno de sincronización/desincronización de las neuronas en las capas corticales de V4. En este caso, las neuronas oscilan inicialmente a una frecuencia aleatoria, pero cuando se presentan los objetos del escenario, las neuronas en V4 van representando cada uno de los objetos del escenario agrupándose para formar capas corticales que oscilan a la misma frecuencia. Es decir, aquellas neuronas que oscilan a la misma frecuencia se agrupan para representar un objeto y por consiguiente, cada objeto que se percibe genera un grupo de neuronas que oscilan a diferentes frecuencias. En la Figura 2.10 se muestra un ejemplo de una segmentación de objetos basada en los modelos de redes mencionados anteriormente. En la Figura 2.10 a) se presenta una imagen que es la entrada a una red neuronal de 2 capas, inspirada en los modelos de sincronización/desincronización, en b) se muestra la actividad de las neuronas que oscilan sincronizadamente para un mismo objeto y en c), se muestra la segmentación de 4 objetos, donde cada cuadro de gris representa un grupo de neuronas que oscilan a la misma frecuencia. Este proceso ha inspirado múltiples algoritmos no supervisados de segmentación de objetos, como aquellos basados en LEGION [5][58] y Wilson & Cowan [59]. Los resultados mostrados en [5], [58] y [59] son parecidos a la manera en que un algoritmo de agrupamiento define regiones de objetos, y que utilizan la detección de bordes para definir los parámetros del algoritmo. Por lo tanto, se diseñó un método de agrupamiento para la

segmentación de objetos utilizando los bordes para definir el número de objetos y luminancia para la definición de las regiones.

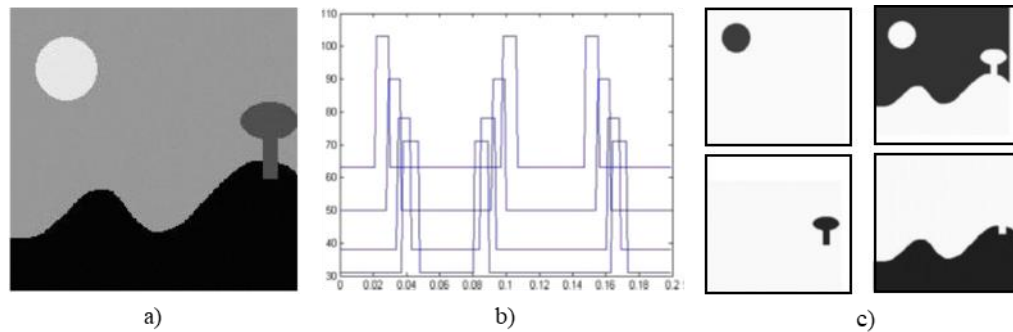


Figura 2.10. Ejemplo de segmentación de objetos basada en la sincronización/desincronización de las neuronas, a) Imagen original, b) Oscilación de las neuronas, c) Segmentación de objetos.

El agrupamiento es un método en el que un conjunto de datos o muestras se divide en grupos que comparten características similares, lo cual es equivalente a la sincronización/desincronización de las neuronas en V4. En la mayoría de los algoritmos de agrupamiento se evalúa la distancia entre los centroides de los grupos a los datos para conocer a que grupo pertenece cada uno. La salida de un algoritmo de agrupamiento es básicamente una descripción de los centroides de agrupamiento con sus respectivos grupos de datos [60].

En este trabajo se utilizan los métodos de agrupamiento basados en particiones, en teoría difusa y en modelos estadísticos debido a que son los más destacados en la literatura por su desempeño en la segmentación de objetos.

2.2. Algoritmos propuestos

Luego del experimento basado en EEG, se implementó un algoritmo para detectar objetos donde se utiliza la información de bordes para definir la cantidad de clases utilizadas para que los algoritmos de agrupamiento generen regiones. Esta idea parte de las conclusiones de los experimentos de EEG, donde se observó que ciertas áreas de la corteza visual muestran actividad relacionada a la percepción de bordes. La Figura 2.11 muestra el esquema del método DObjA, que se divide en preprocesamiento, extracción de características, agrupamiento y determinación del número de grupos. El módulo de agrupamiento se probó con tres algoritmos y para determinar cuál algoritmo de agrupamiento tiene mejor desempeño, el método DObjA

se probó con la base de datos que proporciona el *benchmark* VSB100 (lo cual se analiza en el capítulo de resultados).

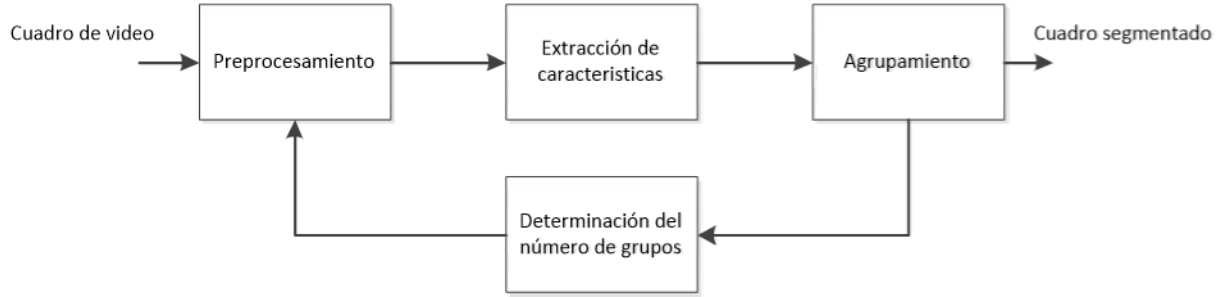


Figura 2.11. Esquema de los métodos propuestos de agrupamiento.

2.2.1. Preprocesamiento y obtención de características

La entrada del algoritmo es un cuadro $I_{RGB}^t(x, y, z)$ de una secuencia de video, donde t es un índice de tiempo de cada cuadro, (x, y) son los índices de la posición de los píxeles y z el índice del componente de color. Dicho cuadro está en el espacio RGB y se transforma al espacio CIELAB, ya que este espacio tiene los siguientes tres componentes: L de luminancia que representa la intensidad de la luz que reflejan los objetos, a que contiene la oponencia rojo/verde y b que contiene la oponencia azul/amarillo. Los valores negativos representan en a el color verde y en b el amarillo. Los valores positivos indican en a el rojo y en b el azul.

Para el método propuesto, se utiliza solo la componente de luminancia, mientras que los componentes a y b se descartan debido a que los resultados del experimento con señales EEG muestran que la detección de los objetos es invariante al color. En DObjA, la componente de luminancia se define con el cuadro $I_L^t(x, y)$, donde cada píxel puede tener un valor dentro del rango de $[0, 100]$, siendo el valor de 0 negro y el valor de 100 blanco. A $I_L^t(x, y)$ se le aplica un filtro pasa bajas Gaussiano para obtener regiones más uniformes y sin ruido. El filtro utilizado se describe como:

$$G(u, v) = e^{\frac{-D^2(u, v)}{2D_0^2}} \quad (2.3)$$

donde $D(u, v)$ es la distancia del centro del filtro a cada píxel con posición (u, v) y se define de la siguiente manera:

$$D(u, v) = \left[\left(u - \frac{U}{2} \right)^2 + \left(v - \frac{V}{2} \right)^2 \right]^{1/2} \quad (2.4)$$

donde U es el número de filas y V el número de columnas. D_0 es la desviación estándar de la función gaussiana del filtro definida como 40, valor que se obtuvo experimentalmente. El filtro Gaussiano generado se muestra en la Figura 2.12.



Figura 2.12. Filtro Gaussiano pasa bajas propuesto.

Aplicando el filtro $G(u, v)$ al cuadro de luminancia $I_L^t(x, y)$, se obtiene:

$$G(u, v): I_L^t(x, y) \rightarrow I_G^t(x, y) \quad (2.5)$$

donde $I_G^t(x, y)$ es el cuadro de luminancia filtrado, que se toma como matriz de características para utilizarse en los algoritmos de agrupamiento. Un ejemplo de aplicar este algoritmo a algunos cuadros de video se muestra en la Figura 2.13.

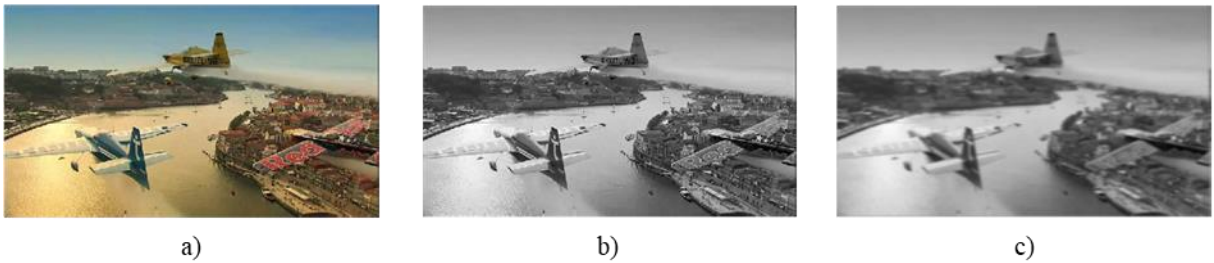


Figura 2.13. a) Imagen original $I_{RGB}^t(x, y)$, b) Imagen de luminancia $I_L^t(x, y)$, c) Imagen filtrada $I_G^t(x, y)$.

Posteriormente, se obtienen los bordes del cuadro $I_L^t(x, y)$ para conocer el número de grupos. Dichos bordes se definen como un cambio abrupto en la intensidad de pixeles vecinos que se organizan típicamente en un conjunto de segmentos de línea o curvas.

De los algoritmos que se encuentran en la literatura para la detección de bordes, destaca el operador Canny, ya que usa múltiples etapas para detectar una amplia gama de bordes incluyendo los bordes diagonales verticales y horizontales, razón por la cual se optó por utilizar este algoritmo.

Los parámetros para aplicar el algoritmo Canny en DObjA se basan en valores genéricos utilizados para obtener bordes en imágenes, estos se definen a continuación:

- Porcentaje de pixeles que no son bordes, $P_{NB} = 0.7$.
- El valor de la desviación estándar del filtro gaussiano de suavizado, $\sigma_{GS} = \sqrt{2}$.
- El umbral para detectar bordes fuertes, HT , se determina con: P_{NB} y MGr que es la magnitud de los gradientes verticales Gr_v y horizontales Gr_h del cuadro $I_L^t(x, y)$, obtenidos con el filtro gaussiano de suavizado.
- El umbral para detectar bordes débiles, LT , es una fracción del umbral HT calculado con $LT = 0.4HT$.

El operador Canny se define como C y se aplica al cuadro $I_L^t(x, y)$ de la siguiente manera:

$$C(\sigma_{GS}, HT, LT): I_L^t(x, y) \rightarrow I_{LB}^t(x, y) \quad (2.6)$$

En la Figura 2.14 se muestra un ejemplo de los bordes obtenidos de aplicar Canny a un cuadro $I_L^t(x, y)$.



Figura 2.14. a) Cuadro de luminancia $I_L^t(x, y)$, b) Imagen de bordes $I_{LB}^t(x, y)$ del cuadro de luminancia.

Una vez preprocesado el cuadro y detectados los bordes, se procede a aplicar 3 modelos de agrupamiento que utilizan la información de $I_G^t(x, y)$ para agrupar e $I_{LB}^t(x, y)$ para determinar el número de grupos, estos modelos son: *K-means*, FCM y *expectation-maximization* (EM). El número de grupos se consigue comparando distintos coeficientes de correlación de Pearson entre los bordes de un cuadro de luminancia y los bordes de segmentaciones de prueba obtenidos con diferentes números de grupos entre 2 y 20. Este proceso se realiza para cada cuadro de video en el módulo de retroalimentación, que se muestra en la Figura 2.11 y que se explica en la subsección 2.2.5.

2.2.2. Algoritmo *K-means*

Se trata de un algoritmo que encuentra el centro de un conjunto de datos que se agrupan para generar clases o categorías. *K-means* calcula dichos centros en forma iterativa hasta que se cumplan algunos criterios de convergencia [61][60]. La explicación del algoritmo se define a continuación: suponer que se tiene una variable aleatoria X de tamaño N , y se refiere a un conjunto de datos que se va a dividir en un número K de grupos, donde K es conocido. Intuitivamente, se puede pensar que un grupo comprende un conjunto de datos cuyas distancias entre ellos son pequeñas en comparación con las distancias de los puntos de otros grupos. Para formalizar esta noción, se introduce primero un conjunto de vectores $\mu_{Km}(k)$, donde cada vector es un prototipo o centro asociado con el k -ésimo grupo y $k = 1, \dots, K$. El objetivo es asignar los datos a cada grupo mediante la diferencia cuadrática mínima de cada dato $x_G(n)$ respecto a cada prototipo $\mu_{Km}(k)$ donde $n = 1, 2, \dots, N$ se refiere al índice por pixel que corresponde al número de observaciones.

Para el caso del método DObjA, *K-means* se implementa de la siguiente forma: los datos son los pixeles del cuadro $I_G^t(x, y)$ y los grupos se definen en el cuadro $I_{A_{\delta n}}^t(x, y)$ con $\delta n = 1$ (δn se utiliza para asignar el método de agrupamiento, donde $\delta n = 1$ para *K-means*, $\delta n = 2$ para FCM y $\delta n = 3$ para EM). $I_G^t(x, y)$ se transforma a una dimensión y se denota por $x_G(n)$. Luego, se define un arreglo binario $r^{n,k} \in \{0, 1\}$, y representa a cuál de los grupos k se le asigna el pixel $x_G(n)$. Es decir, si $x_G(n)$ se asigna al clúster k_1 , entonces $r^{n,k_1} = 1$, y $r^{n,j_1} =$

0 para $j_1 \neq k_1$. El criterio que se utiliza para la asignación de los pixeles a los grupos consta de minimizar la siguiente función de costo:

$$J = \sum_{n=1}^{N_o} \sum_{k=1}^K r^{n,k} \|x_G(n) - \mu_{K_m}(k)\|^2. \quad (2.7)$$

donde $\|*\|$ es la norma que expresa la similitud entre cualquier pixel y un centro. Consecuentemente, *K-means* encuentra los valores de $r^{n,k}$ y $\mu_{K_m}(k)$ para minimizar J . Se puede hacer a través de un procedimiento iterativo con los siguientes pasos:

Paso 1: se proponen la cantidad K de grupos y los centros iniciales $\mu_{K_m}(k)$. Si no existen antecedentes de $\mu_{K_m}(k)$ entonces se elige de forma aleatoria.

Paso 2: se minimiza J . Para ello, se asigna cada $x_G(n)$ al centro del clúster más cercano. Más formalmente, esto se puede expresar como:

$$r^{n,k} = \begin{cases} 1 & \text{si } k = \operatorname{argmin}_k \|x_G(n) - \mu_{K_m}(k)\|^2 \\ 0 & \text{de otra forma} \end{cases}. \quad (2.8)$$

Paso 3: se realiza la optimización de $\mu_{K_m}(k)$ a través de $r^{n,k}$. La función de costo J es una función cuadrática de $\mu_{K_m}(k)$, y puede minimizarse estableciendo su derivada con respecto a $\mu_{K_m}(k)$ a cero como se muestra en (2.9)

$$2 \sum_{n=1}^N r^{n,k} (x_G(n) - \mu_{K_m}(k)) = 0 \quad (2.9)$$

y se refiere a la derivada de la ecuación (2.7). Despejando $\mu_{K_m}(k)$ en (2.9) se obtiene

$$\mu_{km}(K) = \frac{\sum_n r^{n,k} x_G(n)}{\sum_n r^{n,k}}. \quad (2.10)$$

El resultado de la ecuación (2.10) tiene una interpretación simple: establecer $\mu_{K_m}(k)$ como el centro de todos los pixeles $x_G(n)$ asignados al k -ésimo grupo. Por esta razón, el procedimiento se conoce como el algoritmo *K-means*.

Paso 4: los pasos 2 y 3 iteran hasta que se cumpla la condición $\max\{|\mu_{km}^{t+1}(K) - \mu_{km}^t(K)|\} < \varepsilon$, donde ε es un criterio de terminación definido experimentalmente con un valor de 0.005. Este procedimiento converge a un mínimo que puede ser local o global, dependiendo del valor inicial de los centros [62]. En la Figura 2.15 se muestra un diagrama del algoritmo *K-means* para ilustrar los pasos correspondientes a este algoritmo.

Una vez obtenidos los valores de las medias $\mu_{Km}(k)$ y terminado el algoritmo de *K-means*, se procede a clasificar las características $x_G(n)$ utilizando la siguiente ecuación,

$$x_{KM}(n) = \underset{k}{\operatorname{argmin}} \|x_G(n) - \mu_{Km}(k)\| \quad (2.11)$$

donde $x_{KM}(n)$ es un nuevo vector en el que cada dato representa el grupo al que pertenecen los pixeles de $x_G(n)$. Para finalizar, se convierte el vector $x_{KM}(n)$ a una imagen $I_{A_{\delta n}}^t(x, y)$ donde $\delta n = 1$, que es la imagen segmentada final.

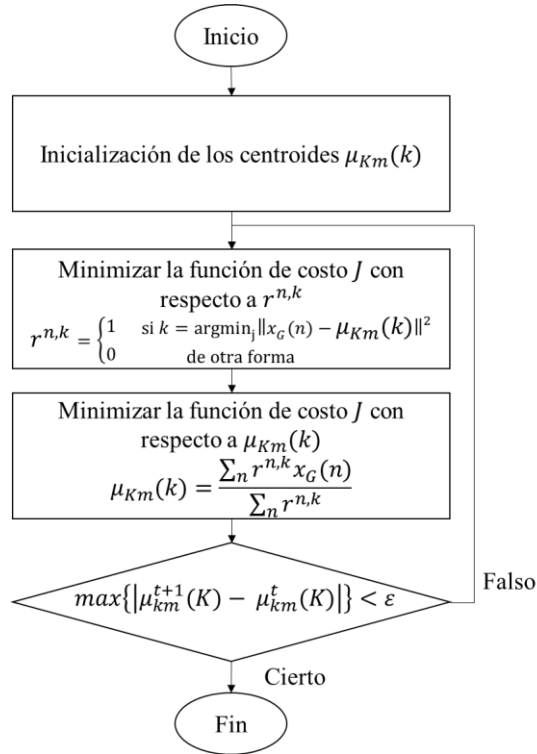


Figura 2.15. Diagrama del algoritmo de *K-means*.

2.2.3. Algoritmo FCM

El algoritmo FCM es un método de agrupamiento basado en que los datos de cualquier conjunto X pertenecen a todos los grupos, pero la asignación de un dato a un clúster se asigna con el centro que tiene el mayor grado de membresía entre este y cada dato. En consecuencia, el valor discreto del grado de pertenencia con valores de $\{0, 1\}$, se cambia al intervalo continuo $[0, 1]$ para describir la relación de pertenencia entre los datos y los grupos. De esta manera, la idea central de FCM es ajustar funciones de membresía por cada grupo y determinar el grado

de pertenencia de cada dato a los grupos [61][60]. Este método se basa en la minimización de la siguiente función de costo:

$$J_C = \sum_{n=1}^N \sum_{k=1}^K u_{n,k}^{m_c} \|x_G(n) - c(k)\|^2, \quad 1 \leq m_c < \infty \quad (2.12)$$

donde m_c es cualquier número real mayor a 1, $u_{n,k}^{m_c}$ es el grado de membresía de $x_G(n)$ en el clúster k , $c(k)$ es el centro del clúster. A continuación, se muestran los pasos que se realizan para la aplicación de este algoritmo:

Paso 1: se inicializan las funciones de membresía $u_{n,k}^{m_c}$ de forma aleatoria.

Paso 2: se calculan los centros $c(k)$ utilizando $u_{n,k}^{m_c}$, a través de la ecuación (2.13), que regresa la media de todos los pixeles, ponderados por su grado de pertenencia al clúster k .

$$c(k) = \frac{\sum_{n=1}^N u_{n,k}^{m_c} \cdot x_G(n)}{\sum_{n=1}^N u_{n,k}^{m_c}} \quad (2.13)$$

Paso 3: la partición difusa se lleva a cabo a través de la optimización iterativa de la función objetivo que se muestra en la ecuación (2.12). Para ello, se utiliza la actualización de la membresía $u_{n,k}^{m_c}$ y los centros de clúster $c(k)$ dadas por:

$$u_{n,k}^{m_c} = \frac{1}{\sum_{k=1}^K \left(\frac{\|x_L(n) - c(k)\|}{\|x_L(n) - c(k_c)\|} \right)^{\frac{2}{m_c-1}}} \quad (2.14)$$

Paso 4: los pasos 2 y 3 iteran hasta que se cumpla la condición $\max\{|u_{n,k}^{m_c}(t+1) - u_{n,k}^{m_c}(t)|\} < \varepsilon$, donde ε se definió experimentalmente con un valor de 0.005. Al igual que K-means, este procedimiento converge a un mínimo que puede ser local o global, dependiendo del valor inicial de los centros. Los pasos del algoritmo FCM se ejemplifican en el diagrama de la Figura 2.16.

Para clasificar las características $x_G(n)$ se aplica la siguiente ecuación,

$$x_{FCM}(n) = \underset{k}{\operatorname{argmin}} \|x_G(n) - c(k)\| \quad (2.15)$$

donde $x_{FCM}(n)$ es un nuevo vector en el que cada dato representa el grupo al que pertenecen los pixeles de $x_G(n)$. Para finalizar, se convierte el vector $x_{FCM}(n)$ a una matriz de dos

dimensiones obteniendo el cuadro $I_{A_{\delta n}}^{(t)}(x, y)$ que es la imagen segmentada final, donde $\delta n = 2$.

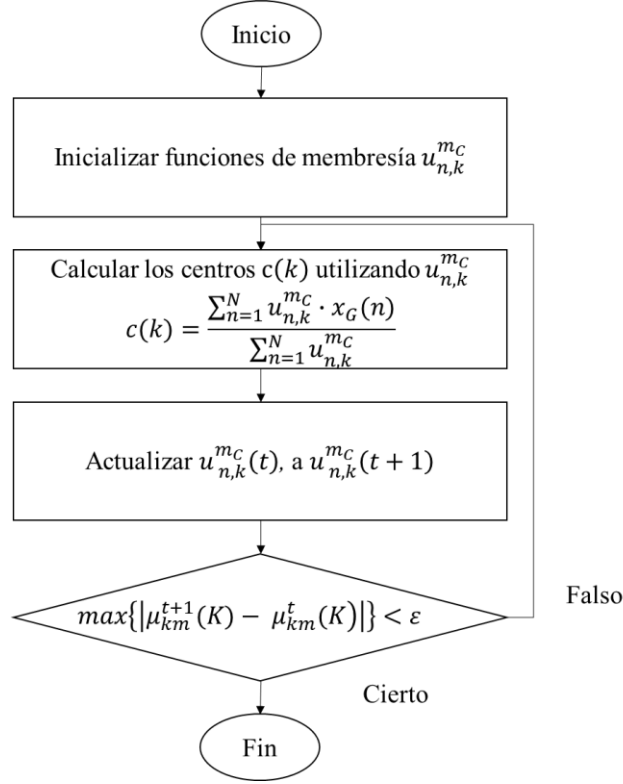


Figura 2.16. Diagrama del algoritmo de FCM.

2.2.4. Algoritmo de EM

En este método, cada clase que define los objetos en la imagen se caracteriza por una función paramétrica de densidad de probabilidad que asigna a cada pixel a un grupo. La ventaja de este método es que permite definir los grupos basándose en mediciones de tendencia central y dispersión, pero la desventaja es el alto costo computacional para conseguir un buen ajuste de los parámetros de los modelos [60][61][63]. La forma más común de implementar EM es utilizar pdf gaussianas dadas por:

$$p_G(x_G(n)|k) = (2\pi)^{-\frac{1}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (x_G(n) - \mu_G(k))^T |\Sigma_k|^{-1} (x_G(n) - \mu_G(k)) \right] \quad (2.16)$$

donde Σ_k es la covarianza y $\mu_G(k)$ es la media de cada distribución gaussiana. Los pasos para aplicar este algoritmo se muestran a continuación:

Paso 1: Iniciar $\mu_G(k)$, $\sigma_G(k)$, $p_G(k)$ en forma aleatoria, donde $k = 1, 2, \dots, K$, y K es el número de grupos igual al número de componentes gaussianas.

Paso 2: Llamado *E-step* y consiste en calcular la expectativa de que un pixel sea parte de cada gaussiana por medio del cálculo de las probabilidades posteriores utilizando la siguiente ecuación:

$$e^{(n,k)} = (2\pi)^{-\frac{1}{2}} \Sigma_G^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (x_G(n) - \mu_G(k))^T \Sigma_G^{-1} (x_G(n) - \mu_G(k)) \right] p_G(k) \quad (2.17)$$

Paso 3: Llamado *M-step* y consiste en calcular parámetros para reajustar cada componente gaussiana utilizando las ecuaciones (2.18)-(2.20) [62].

$$\mu_G(k) = \frac{\sum_{n=1}^N e^{n,k} x_G(n)}{\sum_{n=1}^N e^{n,k}} \quad (2.18)$$

$$\Sigma_G(k) = \frac{\sum_{n=1}^N e^{n,k} (x_G(n) - \mu_G(k))^T (x_G(n) - \mu_G(k))}{\sum_{n=1}^N e^{n,k}} \quad (2.19)$$

$$p_G(k) = \frac{1}{N} \sum_{n=1}^N e^{n,k} \quad (2.20)$$

Paso 4: Finalmente, se verifica la convergencia de los parámetros $\mu_G(k)$, $\sigma_G(k)$ y $p_G(k)$. Esto se consigue comparando la expectativa anterior con la actual, de la siguiente manera,

$$\Delta e = \left\| \sum_n \sum_k e^{n,k}(t) - e^{n,k}(t+1) \right\| \quad (2.21)$$

Si $\Delta e < \varepsilon$ detener, sino regresar al paso 2, donde ε se definió experimentalmente con un valor de 0.0001. La Figura 2.17 se muestra un diagrama del algoritmo utilizado para implementar el modelado de mezclas gaussianas utilizando el método EM.

Para clasificar las características $x_G(n)$, se obtiene la probabilidad de cada una de las distribuciones gaussianas. Aquella con la probabilidad más alta establece la clase a la que pertenece, esto se consigue con la siguiente ecuación:

$$x_{EM}(n) = \underset{k}{\operatorname{argmax}}(p_G(x_G(n)|k)) \quad (2.22)$$

donde $x_{EM}(n)$ es un nuevo vector en el que cada dato representa el grupo al que pertenecen los pixeles de $x_G(n)$. Para finalizar, se transforma el vector $x_{EM}(n)$ a un cuadro $I_{A_{\delta n}}^{(t)}(x, y)$ donde $\delta n = 3$, que es el cuadro segmentado final.

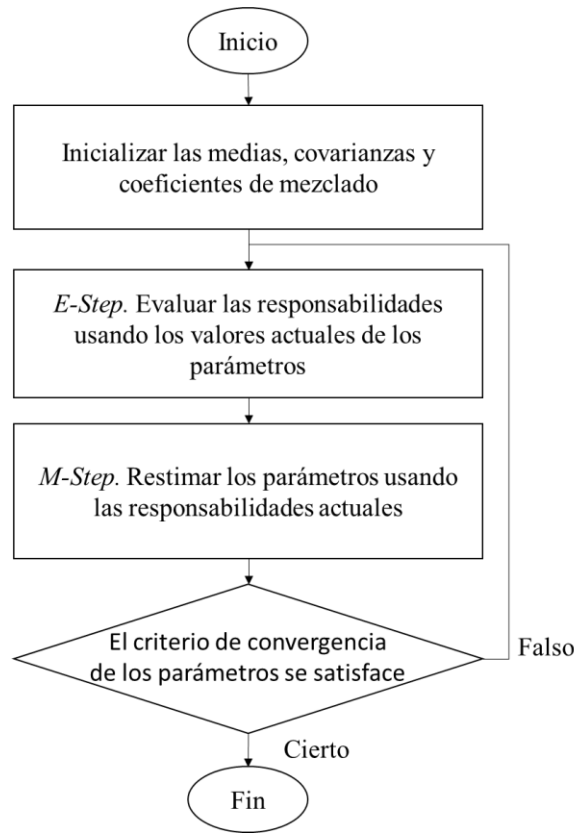


Figura 2.17. Diagrama del algoritmo de mezclas gaussianas usando EM.

2.2.5. Determinación del número de grupos y segmentación de objetos

Todos los algoritmos de agrupamiento tienen como parámetro el número de grupos en los que se va a dividir el cuadro. Inicialmente para cada cuadro se propone una cantidad de grupos igual a 2, que se va incrementando para buscar la cantidad ideal. Este paso se basó en el experimento de EEG, donde para el reconocimiento de objetos se captura la información de bordes y se hace un proceso similar al agrupamiento, como se menciona en la sección 2.1.5. Por lo tanto, para la determinación de K según los bordes y para realizar la segmentación de objetos se realizan los siguientes pasos para cada uno de los cuadros de video:

Paso 1: primeramente, se obtiene $I_{LB}^t(x, y)$ con la ecuación (2.6), se elige $K = 2$ y se proponen los parámetros en forma aleatoria, que son: para *K-means* $\mu_{Km}(k)$, para FCM $u_{n,k}^{mc}$ y para EM son $\mu_G(k)$, $\sigma_G(k)$ y $p_G(k)$.

Paso 2: se realiza el agrupamiento en forma iterativa actualizando el valor de los parámetros mencionados en el paso 1. Luego se calcula $I_{A_{\delta n}}^t(x, y)$ para $\delta n = 1$ (K-means), $\delta n = 2$ (FCM) y $\delta n = 3$ (EM).

Paso 3: se obtienen los bordes de $I_{A_{\delta}}^t(x, y)$ como sigue,

$$C: I_{A_{\delta n}}^t(x, y) \rightarrow I_{B_{\delta n}}^t(x, y) \quad (2.23)$$

Paso 4: se encuentra la similitud mediante correlación entre $I_{LB}^t(x, y)$ y los bordes $I_{B_{\delta n}}^t(x, y)$. Este análisis se realiza por cada uno de los cuadros de la secuencia de video. El valor de K se va incrementando hasta que se llega a $K = 20$ y en cada cuadro se obtiene la correlación de Pearson de $I_{LB}^t(x, y)$ e $I_{B_{\delta n}}^t(x, y)$ de la siguiente manera,

$$\rho_{\delta n, k}(t) = \frac{\sigma_{LBB_{\delta n}}}{\sigma_{LB}\sigma_{B_{\delta n}}} \quad (2.24)$$

donde $\sigma_{LBB_{\delta n}}$ es la covarianza de $I_{LB}^t(x, y)$ e $I_{B_{\delta n}}^t(x, y)$, σ_{LB} es la varianza de $I_{LB}^t(x, y)$ y $\sigma_{B_{\delta n}}$ es la varianza de $I_{B_{\delta n}}^t(x, y)$.

Paso 5: utilizando los coeficientes $\rho_{\delta, k}(t)$ se calcula la cantidad de grupos que se obtiene por la correlación más alta entre $I_{LB}^t(x, y)$ e $I_{B_{\delta n}}^t(x, y)$, de la siguiente manera:

$$\Lambda_{\delta} = \underset{k}{\operatorname{argmax}} \rho_{\delta n, k}(t) \quad (2.25)$$

Los pasos anteriores se realizan para encontrar la cantidad adecuada de grupos que definan las regiones que más se asemejan a los objetos del escenario.

En la Figura 2.18 se muestran ejemplos de cómo quedan los cuadros de bordes de $I_{LB}^t(x, y)$ e $I_{B_{\delta n}}^t(x, y)$ al segmentar $I_{A_{\delta n}}^t(x, y)$ con la K obtenida de la correlación mayor.

Paso 6: finalmente, se utiliza el valor de K calculado para obtener los cuadros segmentados $I_{A_{\delta n}}^t(x, y)$.

La Figura 2.19 muestra los resultantes de aplicar los 3 algoritmos de segmentación con valor de $k = 5$ en un cuadro de la base de datos VSB100. Perceptualmente, en el cuadro de la Figura 2.19 a) y en otros de la misma base de datos, los algoritmos de K-means en b) y FCM en c) tienen resultados similares espacial y temporalmente, ya que los contornos de las regiones segmentadas son parecidos. En el caso de EM en d), este tiende a generar menos regiones

cuando la luminancia tiene valores bajos y más regiones cuando son altos a diferencia de los otros 2 algoritmos, lo que afecta la segmentación temporal.

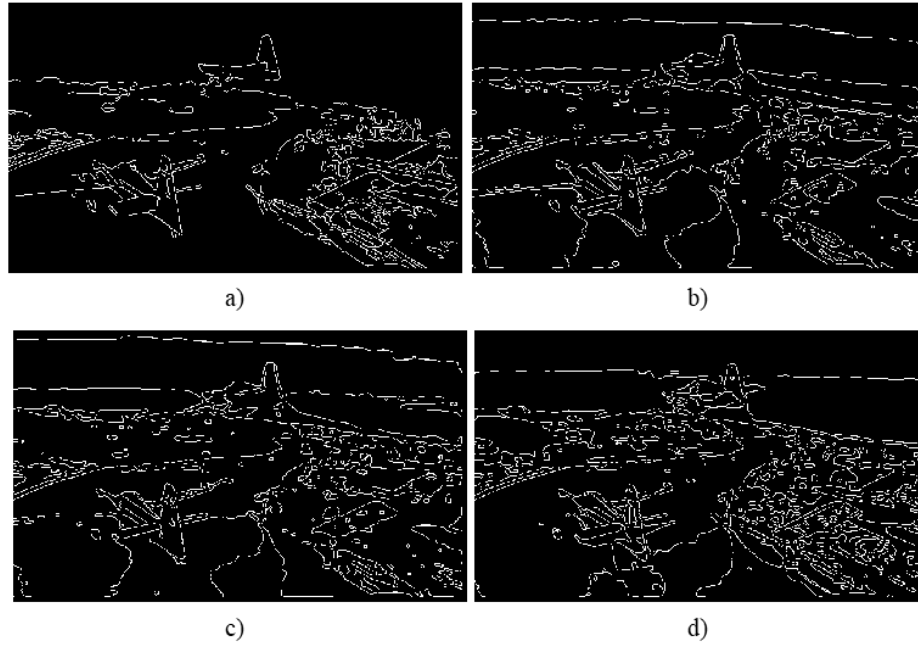


Figura 2.18. Cuadros de bordes. a) $I_{LB}^t(x, y)$, b) $I_{B_1}^t(x, y)$, c) $I_{B_2}^t(x, y)$ y d) $I_{B_3}^t(x, y)$.

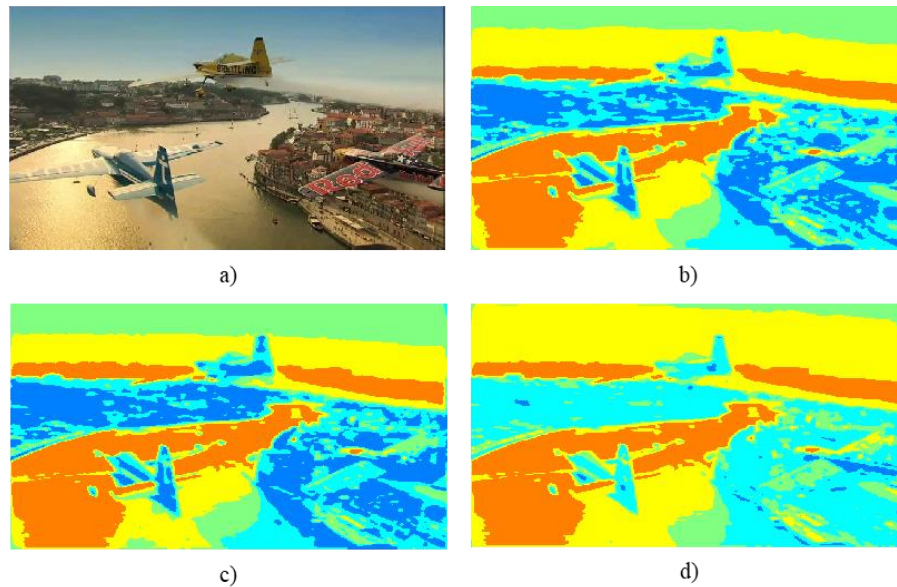


Figura 2.19. a) Imagen original $I_{RGB}(x, y, z)$, b) Imagen segmentada con K-means $I_{A_1}^t(x, y)$, b) Imagen segmentada con FCM $I_{A_2}^t(x, y)$, c) Imagen segmentada con modelado de mezclas gaussianas $I_{A_3}^t(x, y)$.

CAPÍTULO III

SEGMENTACIÓN NEUROINSPIRADA

Para desarrollar un método de segmentación de objetos neuroinspirado, se puede presentar no solo el esquema del capítulo anterior, también se puede abordar un esquema que parte de modelos computacionales basados en la corteza visual. Consecuentemente, en este capítulo se presenta el algoritmo denominado NISeg, que se basa en los modelos que explican el funcionamiento de cada una de las áreas corticales de los caminos dorsal y ventral. NISeg se muestra en el esquema de la Figura 3.1, donde el primer módulo captura un cuadro en el espacio de color RGB y se denomina retina. El siguiente módulo es el Núcleo Geniculado Lateral (NGL) y su función es encontrar la luminancia del cuadro y los componentes de color oponentes (azul-amarillo, rojo-verde). Posteriormente, se encuentran los módulos de las áreas V1/V2, los que procesan bordes, color y movimiento. El módulo de bordes detecta los contornos de los objetos mediante una red neuronal celular con plantillas basadas en filtros de Gabor. El de color analiza la crominancia en cada cuadro mediante un método no paramétrico y el de flujo óptico determina los patrones de movimiento contenidos en el escenario. El siguiente módulo es el área MT o V5 y se encarga de generar un modelo de atención que descarta patrones de movimiento que no son relevantes. Finalmente, el último módulo definido por V4 obtiene la segmentación de objetos, y a diferencia del enfoque clásico de la Figura 1.7, en el esquema de NISeg V4 retroalimenta a las áreas V1/V2, basándose en [64].

En este capítulo se presenta cada parte de NISeg de la siguiente manera: en la sección 3.1 se describen la retina y los NGL. Se presentan en la sección 3.2 las áreas visuales V1/V2 encargadas del procesamiento de características de bajo nivel. Se describe en la sección 3.3 el área V5 o MT, donde se realiza el reconocimiento de movimiento y en la sección 3.4 se presenta el área V4 encargada del reconocimiento de los objetos.

Cada sección presenta primeramente los detalles de los modelos computacionales que describen la parte de la corteza visual y posteriormente, se describe el algoritmo implementado en cada módulo de NISeg.

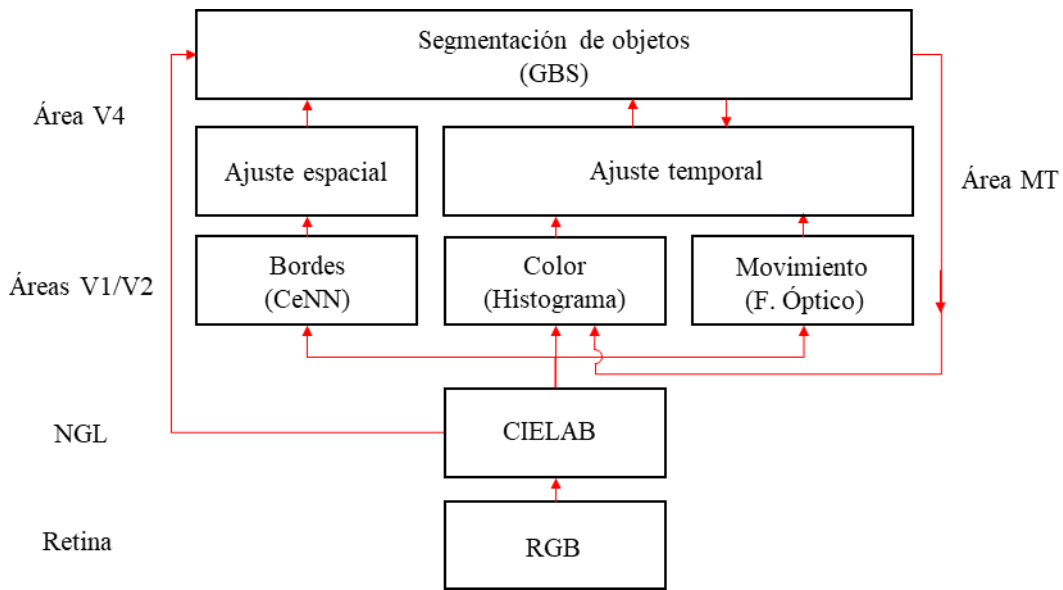


Figura 3.1. Diagrama propuesto de la percepción visual humana.

3.1. Retina y NGL

3.1.1. Modelos neurocomputacionales de la retina y los NGL

Biológicamente, en la retina se proyectan las imágenes de los escenarios observados, siendo esta la entrada al sistema de percepción visual y como se ve en la Figura 1.2, ésta contiene dos tipos de fotorreceptores: los bastones que perciben cambios de iluminación y los conos que perciben los colores. Estos últimos se dividen en 3 tipos: *Short* (S), que son sensibles al color azul, *Medium* (M) que miden el color verde y *Long* (L) que son sensibles al rojo [65].

La información obtenida por los fotorreceptores de la retina pasa a las células ganglionares y los NGL, cuyas neuronas son CR que alteran su respuesta en la tasa de impulsos electroquímicos cuando se presentan estímulos visuales. Los CR de color y luminancia tienen dos tipos de canales: ON-centro/OFF-envolvente y OFF-centro/ON-envolvente, tal como se menciona en la sección 1.1.

Enroth-Cugell y Robson propusieron en [66] que los CR se pueden representar matemáticamente como una función de diferencias de Gaussianas (DoG). La gaussiana con mayor desviación estándar corresponde a la envolvente y la más pequeña corresponde al centro, como se muestra la Figura 3.2 b) (vista transversal del CR mostrado en la Figura 3.2 a)). Este

modelo predice la capacidad de respuesta de un campo receptivo a un estímulo único y se expresa como:

$$CR(x_r, y_r) = \frac{\exp\left(\frac{-(x_r - x_c)^2 + (y_r - y_c)^2}{\sigma_c^2}\right)}{\sum_{u_r, v_r} \exp\left(\frac{-(u_r - x_c)^2 + (v_r - y_c)^2}{\sigma_c^2}\right)} - \frac{\exp\left(\frac{-(x_r - x_c)^2 + (y_r - y_c)^2}{\sigma_s^2}\right)}{\sum_{u_r, v_r} \exp\left(\frac{-(u_r - x_c)^2 + (v_r - y_c)^2}{\sigma_s^2}\right)} \quad (3.1)$$

donde (x_r, y_r) representa los receptores con centro (x_c, y_c) de un CR tipo ON-centro/OFF-envolvente y (u_r, v_r) es el índice de la respuesta del campo receptivo. La ecuación 3.1 define 2 gaussianas, la del centro con un ancho determinado por σ_c y la de envolvente con ancho definido por σ_s . En otro trabajo, Krantz modificó la función DoG en [67] incluyendo números imaginarios para representar los campos receptivos como estructuras bidimensionales, ya que es más cercano a la naturaleza física de los campos receptivos. El modelo DoG estándar con las modificaciones de Krantz es capaz de analizar una variedad de estímulos y producir salidas como un campo receptivo.

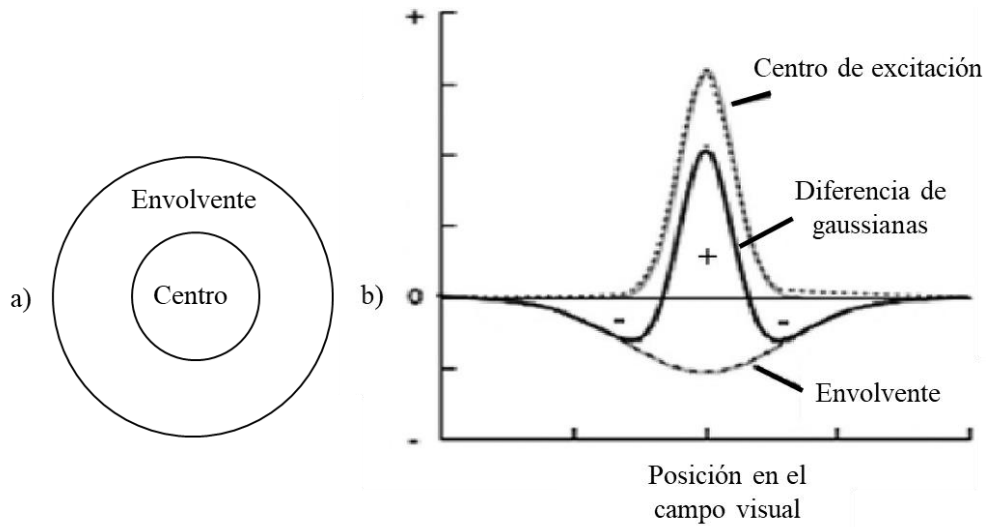


Figura 3.2. a) Envlovente y centro de un CR, b) DoG que simula un CR mostrado transversalmente.

3.1.2. Modelo de la retina y los NGL en NISeg

Computacionalmente, la retina se representa en NISeg con la entrada de un cuadro de video en el espacio de color RGB, imitando los tipos de conos rojos (L), verdes (M) y azules (S). Por lo tanto, cada cuadro de video se representa en el módulo retina por $I_{RGB}^t(x, y, z)$, donde $z = 1$ se refiere al color rojo, $z = 2$ al color verde y $z = 3$ al color azul.

Para los NGL, se propuso utilizar el espacio de color CIELAB para representar los CR. Este espacio de color se diseñó para que fuera uniforme respecto a la visión de color humana, de tal manera que la cantidad de valor numérico en este espacio corresponde a la cantidad de cambio visual percibido. A pesar de no manejar el esquema de DoG, CIELAB si maneja la luminancia y la teoría de colores oponentes, ya que tiene componentes que impiden que haya colores rojo y verde o azul y amarillo en un mismo pixel. Para realizar el cambio del espacio de color RGB a CIELAB se realizan los siguientes pasos:

Paso 1: convertir el cuadro $I_{RGB}^t(x, y, z)$ al espacio de color XYZ. Este espacio tiene las siguientes componentes: luminancia (Y) y el plano XZ, el cual contiene todas las cromaticidades posibles. Para realizar la conversión se utiliza la matriz de transformación de [68] dada por:

$$M_{xyz} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix}. \quad (3.2)$$

M_{xyz} se multiplica por las componentes del cuadro $I_{RGB}^t(x, y, z)$ de la siguiente forma:

$$I_{XYZ}^t(x, y, w) = M_{xyz} * I_{RGB}^t(x, y, z) \quad (3.3)$$

donde w corresponde al índice que representa cada una de las componentes del espacio XYZ.

Paso 2: $I_{XYZ}^t(x, y, w)$ se normaliza para el punto blanco D65 que se define como el centro en el diagrama espacial cromático del espacio de color estándar RGB [69]. Esto se logra aplicando las ecuaciones (3.4) a (3.6).

$$x_r(x, y) = \frac{I_{XYZ}(x, y, 1)}{0.9504} \quad (3.4)$$

$$y_r(x, y) = I_{XYZ}(x, y, 2) \quad (3.5)$$

$$z_r(x, y) = \frac{I_{XYZ}(x, y, 3)}{1.0888} \quad (3.6)$$

donde $x_r(x, y)$ es una mezcla de la curva de sensibilidad del rojo al verde, $y_r(x, y)$ significa luminosidad y z es aproximadamente igual al estímulo de azul.

Paso 3: para obtener las componentes de CIELAB primero se obtienen las matrices $f_x(x, y)$, $f_y(x, y)$, $f_z(x, y)$ a través de las ecuaciones (3.7) a (3.9).

$$f_x(x, y) = \begin{cases} \sqrt[3]{x_r(x, y)} & \text{si } x_r(x, y) > 0.008856 \\ \frac{903.3 x_r(x, y) + 16}{116} & \text{de otra manera} \end{cases} \quad (3.7)$$

$$f_y(x, y) = \begin{cases} \sqrt[3]{y_r(x, y)} & \text{si } y_r(x, y) > 0.008856 \\ \frac{903.3 y_r(x, y) + 16}{116} & \text{de otra manera} \end{cases} \quad (3.8)$$

$$f_z(x, y) = \begin{cases} \sqrt[3]{z_r(x, y)} & \text{si } z_r(x, y) > 0.008856 \\ \frac{903.3 z_r(x, y) + 16}{116} & \text{de otra manera} \end{cases} \quad (3.9)$$

Paso 4: para finalizar, se aplican las ecuaciones (3.10) a (3.12) para obtener las componentes de luminancia, a y b del espacio de color CIELAB.

$$I_L^t(x, y) = 116f_y(x, y) - 16 \quad (3.10)$$

$$I_a^t(x, y) = 500 \left(f_x(x, y) - f_y(x, y) \right) \quad (3.11)$$

$$I_b^t(x, y) = 200 \left(f_y(x, y) - f_z(x, y) \right). \quad (3.12)$$

En la Figura 3.3 se muestra una imagen RGB y como es codificada en las diferentes componentes de CIELAB, donde en b) se presenta la luminancia $I_L^t(x, y)$, en c) la componente $I_a^t(x, y)$ para rojo y en d) para verde, en e) la componente $I_b^t(x, y)$ para amarillo y en f) para azul.

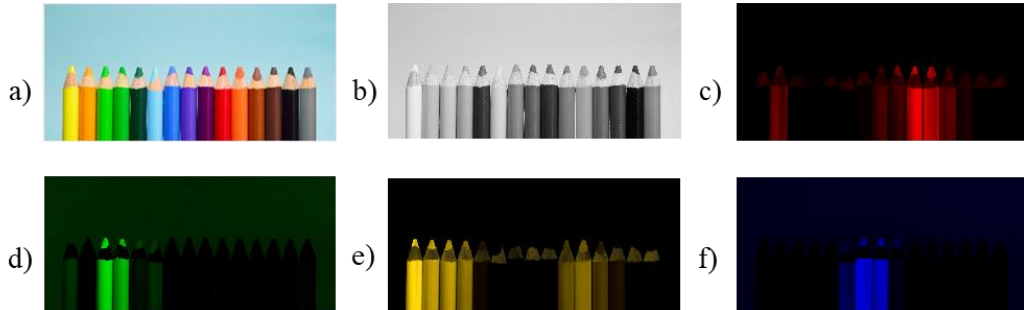


Figura 3.3. a) Codificación de un cuadro $I_{RGB}^t(x, y, z)$ en componentes de CIELab, b) $I_L^t(x, y)$ luminancia, c) $I_a^t(x, y)$ para rojo, d) $I_a^t(x, y)$ para verde, e) $I_b^t(x, y)$ para amarillo f) $I_b^t(x, y)$ para azul.

3.2. Corteza visual primaria (V1) y secundaria (V2)

Una vez que la información de cada cuadro se procesa en el módulo NGL, el siguiente paso es procesar esta información en V1 y V2. Debido a que ambas cortezas procesan las mismas características del estímulo visual de entrada, en NISeg se consideran un nivel de procesamiento denominado V1/V2 y tiene tres módulos: color, forma (bordes) y movimiento [17], los cuales se describirán a continuación:

3.2.1. Color

3.2.1.1. Modelos neurocomputacionales de color

Empezando por la parte biológica, V1/V2 tienen capas corticales que se componen de neuronas que en su conjunto se denominan mapas de selección de color, donde cada neurona responde a un color en particular. El proceso en el que se genera la selectividad de color en estos mapas se llama autoorganización y se trata de un mecanismo de aprendizaje en el que un conjunto de células se organizan a través del tiempo con base en los patrones de los estímulos de entrada que aparecen [70]. Para ilustrar lo anterior, en la Figura 3.4 se muestra una simulación de un mapa de color en V1/V2 en color falso, que indica la selectividad de las neuronas a distintos colores y como se puede observar, las neuronas están organizadas en conjuntos discontinuos y dispersos. Esta forma en la que se organizaron estas neuronas se basó en los colores que regularmente se presentaban en la retina a través del tiempo.

En los mapas de color, existen tres tipos de conjuntos de neuronas: selectivos al rojo, al verde y al azul y como su nombre sugiere, cada conjunto responde cuando se presenta un color en la retina. Por ejemplo, cuando se presenta un color amarillo, reaccionan los conjuntos verde y rojo y cuando se presenta un color cian o morado, reacciona el conjunto azul.

De esta manera, estas neuronas forman un mapa de color que funciona como una combinación de porciones selectivas de color para los 6 colores rojo, amarillo, verde, cian, azul o morado. Esto significa que V1/V2 tiene regiones corticales que construyen la percepción del

color midiendo cuales neuronas se estimulan en el mapa de color con base en los estímulos de entrada.

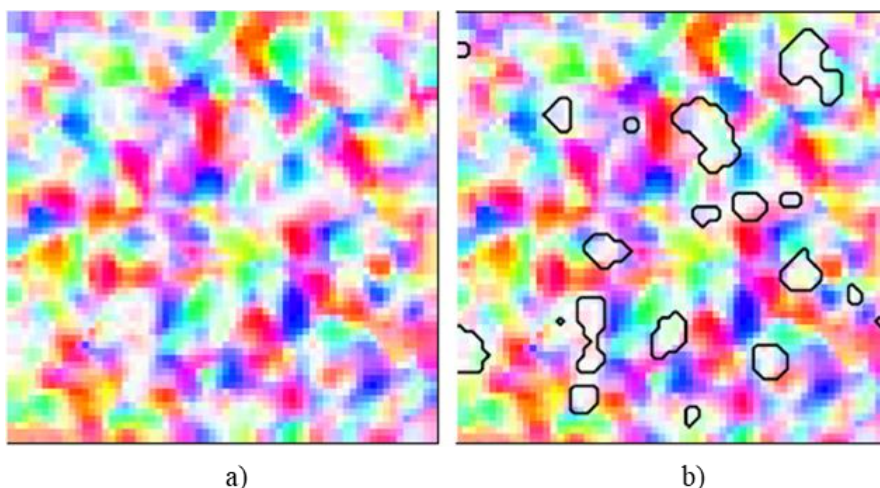


Figura 3.4. Organización de un mapa de color. a) Mapa de color selectivo, b) Contornos de color.

Para explicar los campos receptivos y su conectividad con V1/V2, se muestran las Figura 3.5 y Figura 3.6. La Figura 3.5, muestra las respuestas de los CR de color ante un estímulo rojo, donde los CR Verde/Rojo ON/OFF y Verde/Rojo OFF/ON tienen una mayor activación; estos 2 tipos se conectan a la neurona señalada dentro de un cuadro blanco en el mapa de color de V1/V2 en la Figura 3.6. Las neuronas laterales en este mapa se muestran oscuras cuando hay una fuerte conexión, grises cuando la conexión es débil y blancas cuando la conexión no existe. Las neuronas que muestran color se deben a la variedad de matices del estímulo observado. En la Figura 3.6 b) se muestra un histograma donde los *bins* representan las conexiones laterales de la neurona de la Figura 3.6 con otras con la misma preferencia al color [71].

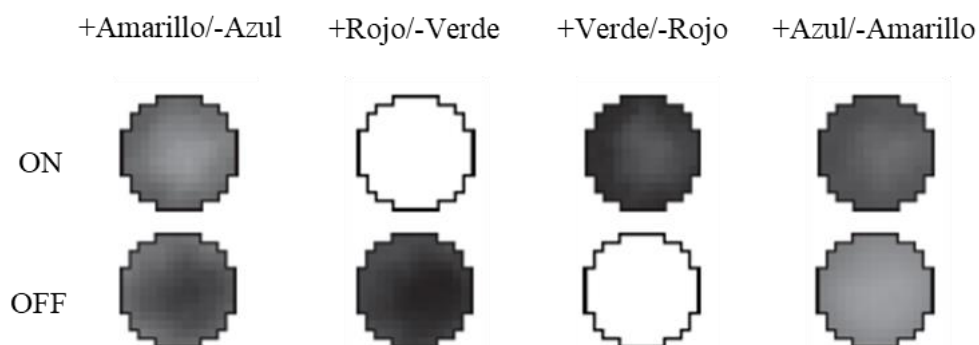


Figura 3.5. Conexión de los campos receptivos ante el estímulo rojo.

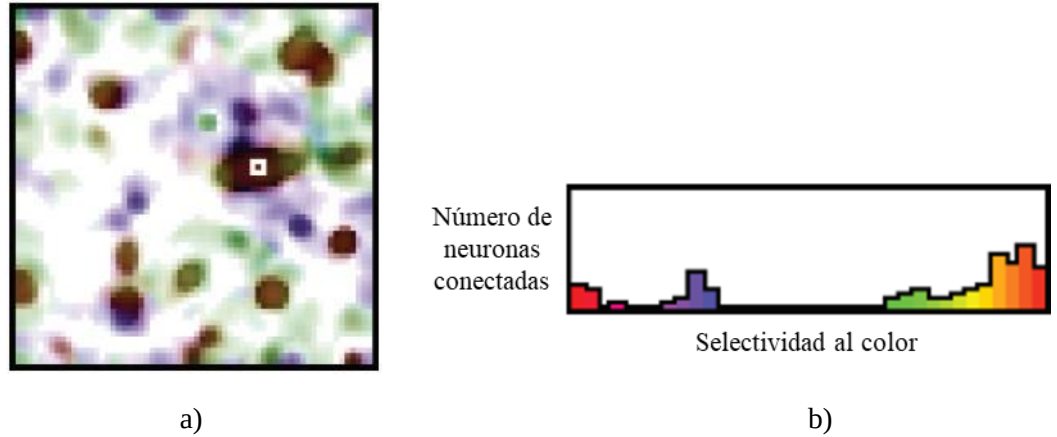


Figura 3.6. Conexiones laterales de una neurona de selectividad al rojo. a) Mapa, b) Histograma [71].

3.2.1.2. Modelo de color en NISeg

De manera computacional, se desarrolló un algoritmo basado en el funcionamiento del procesamiento de color para el área V1/V2 de NISeg. Este algoritmo se basó en un análisis de distribución no paramétrica con los histogramas de los componentes $I_a^{(t)}(x, y)$ e $I_b^{(t)}(x, y)$. No obstante, este análisis no partió directamente de estos componentes, sino que se realizó desde una segmentación preliminar que genera V4, ya que V1/V2 se retroalimentan con V4 como se menciona en la descripción de la Figura 3.1. Consecuentemente, para entender la forma en que se estructura el modelo no paramétrico de color para V1/V2, es necesario presentar el modelo de GBS de V4, con el que se obtienen regiones preliminares que son utilizadas para generar los modelos no paramétricos.

GBS parte del algoritmo de Felzenszwalb y Huttenlocher [72][73][74] y su objetivo es agrupar los píxeles que tengan una apariencia similar, donde la similitud se basa en la variación de color dentro de una región. Por ejemplo, las regiones de píxeles homogéneos en color se deben fusionar, aun cuando las texturas varíen. Consecuentemente, se introduce el concepto de variación interna de una región, donde las regiones se fusionan solo si su variación de color es menor que la diferencia interna dentro de cada región.

A continuación, se procede a formalizar los elementos del algoritmo de GBS de V4. Suponga que $x_{GB} = (V_{GB}, E_{GB})$ es un grafo con vértices $v_{GB} \in V_{GB}$, y se refiere al conjunto de elementos que serán segmentados, y E_{GB} son los bordes definidos por $(v_{GB}^{i_{GB}}, v_{GB}^{j_{GB}}) \in E_{GB}$ que

corresponden a los pares de vértices vecinos 4, donde $i_{GB} \neq j_{GB}$ son índices para diferenciar a 2 distintos vértices vecinos. Cada borde $(v_{GB}^{i_{GB}}, v_{GB}^{j_{GB}})$ tiene un peso correspondiente $w_{GB}(v_{GB}^{i_{GB}}, v_{GB}^{j_{GB}})$, y se define como una medida no negativa de la diferencia entre elementos vecinos $v_{GB}^{i_{GB}}$ y $v_{GB}^{j_{GB}}$. GBS utiliza los árboles de expansión mínima (MST por sus siglas en ingles) para definir un camino de vértices adyacentes, cuya suma de los pesos de sus bordes sea la menor todos los caminos posibles, un ejemplo de esto se muestra en la Figura 3.7 donde se tienen dos árboles de expansión, en a) la suma de pesos entre los bordes es de 11 y en b) de 7, por lo que este último se considera el MST. La finalidad de este camino es para definir la diferencia interna de una región que se utiliza para obtener la diferencia interna mínima y la diferencia entre regiones para determinar si juntar o separar regiones.

Para el algoritmo NISeg, los elementos de V_{GB} son pixeles cuyos valores son definidos por las 3 componentes de CIELAB $I_L^t(x, y)$, $I_a^t(x, y)$ e $I_b^t(x, y)$. Los pesos w_{GB} de los bordes E_{GB} son una medida de diferencia entre 2 pixeles conectados por un borde, y se calcula mediante distancia euclidiana de sus 3 valores, de la siguiente manera,

$$w_{GB} = \sqrt{\left(v_{GB_1}^{i_{GB}} - v_{GB_1}^{j_{GB}}\right)^2 + \left(v_{GB_2}^{i_{GB}} - v_{GB_2}^{j_{GB}}\right)^2 + \left(v_{GB_3}^{i_{GB}} - v_{GB_3}^{j_{GB}}\right)^2}. \quad (3.13)$$

El resultado de GBS es el cuadro segmentado $I_{GB}^t(x, y)$, y se refiere a la partición de elementos de V_{GB} en componentes (regiones) $R \in I_{GB}^t(x, y)$ tales que cada una corresponde a una región conectada en un grafo $x_{GB_1} = (V_{GB}, E_{GB_1})$, donde $E_{GB_1} \subseteq E_{GB}$.

Para NISeg, la diferencia interna de una región se define como el mayor peso en el árbol de expansión mínima $MST(R, E_{GB})$ y se calcula con la ecuación (3.14).

$$Int(R) = \max_{e \in MST(R, E_{GB})} w_{GB}(e) \quad (3.14)$$

Para entender el funcionamiento de la ecuación (3.14) se muestra la Figura 3.7, donde en b) se tiene el MST cuya diferencia interna es igual a 4, que es el borde con mayor peso.

Para visualizar los MST de las regiones de una imagen se muestra la Figura 3.8, donde se tiene la imagen de un árbol en el que cada cuadro corresponde a un pixel, de manera que el MST recorre cada región según la similitud de la característica de color. Los histogramas de color se generan por la información de color de cada región.

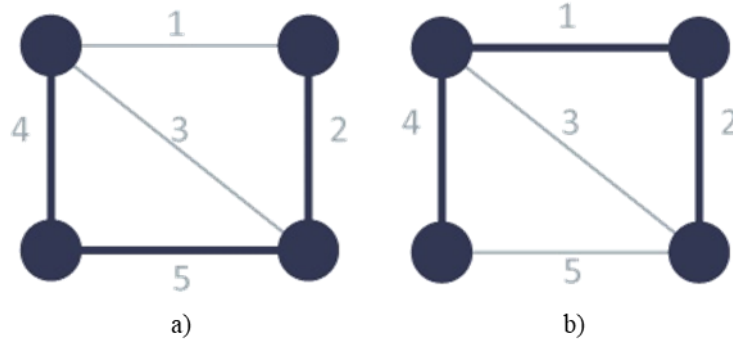


Figura 3.7. a) Árbol de expansión con suma de pesos de 11 y b) MST con suma de pesos de 7.

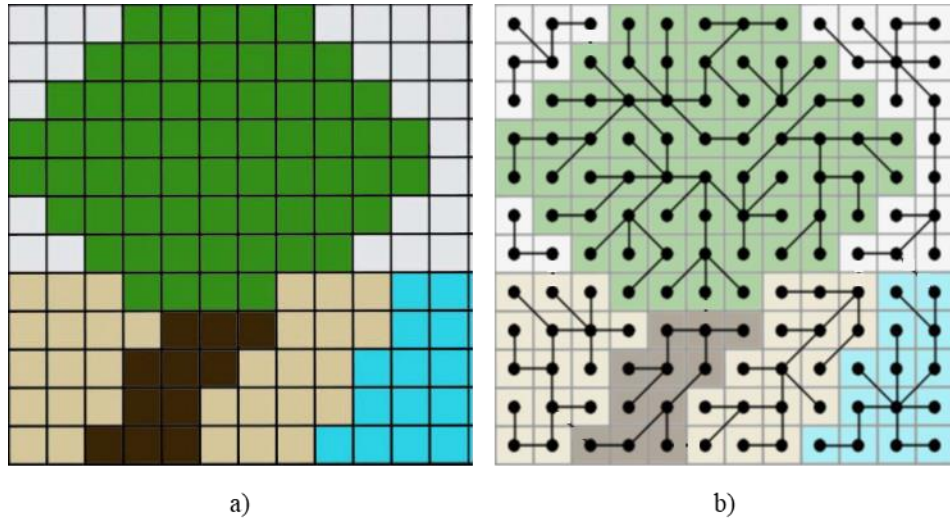


Figura 3.8. a) Imagen original dividida en píxeles y b) MST correspondiente [75].

Después de definir la diferencia interna de una región, se procede a definir la diferencia entre dos regiones $R_1, R_2 \subseteq V_{GB}$, que está dada por el peso mínimo que conecta a ambas regiones de la siguiente manera:

$$Dif(R_1, R_2) = \min_{v_{GB}^{i_{GB}} \in R_1, v_{GB}^{j_{GB}} \in R_2, (v_{GB}^{i_{GB}}, v_{GB}^{j_{GB}}) \in E_{GB}} w_{GB}(v_{GB}^{i_{GB}}, v_{GB}^{j_{GB}}). \quad (3.15)$$

Para generar resultados más robustos en la segmentación, se utiliza una función umbral para controlar el grado al cual la diferencia entre las regiones debe ser menor o igual a la diferencia interna mínima para unir las regiones. La diferencia interna mínima se define con la ecuación siguiente:

$$MInt(R_1, R_2) = \min(Int(R_1) + \tau_{GB}(k_{GB}, R_1), Int(R_2) + \tau_{GB}(k_{GB}, R_2)) \quad (3.16)$$

donde la función umbral es $\tau_{GB}(\cdot)$ y se define como:

$$\tau_{GB}(k_{GB}, R) = \frac{k_{GB}}{|R|} \quad (3.17)$$

donde $|R|$ denota el tamaño de la región R en pixeles y k_{GB} es un parámetro constante, que no debe tener valores menores a 150 para no generar regiones que sean menores en comparación con el tamaño de los objetos.

Para que GBS represente la parte de V4 que procesa color y luminancia, NISeg se propone el siguiente método:

Condiciones de inicio: La entrada es un grafo $x_{GB} = (\{I_L^t(x, y), I_a^t(x, y), I_b^t(x, y)\}, E_{GB})$ con vértices n_{GB} y bordes m_{GB} .

Salida: La salida es una segmentación $I_{GB}^t(x, y)$ que contiene las regiones R_{r, n_R}^t , donde $r = 1, \dots, N_r$ indica una región dentro, N_r el número total de regiones, $n_r = 1, \dots, N_r$ indica un pixel dentro de una región y N_R el número total de pixeles de esta región.

A continuación, se muestran los pasos a seguir para aplicar GBS de NISeg. Considere a q en $I_{GB}^{t,q}(x, y)$, como el cálculo número q de la segmentación que va de 0 a m_{gb} :

Paso 1: se toma cada pixel del cuadro $I_L^t(x, y)$ como una región para obtener la primera segmentación $I_{GB}^{t,0}(x, y)$.

Paso 2: se genera el conjunto $\varphi = (E_{GB_1}, \dots, E_{GB_m})$ que parte de ordenar de manera ascendente cada par de pixeles con base en los pesos de sus bordes $(v_{GB}^{i_{GB}}, v_{GB}^{j_{GB}})$.

Paso 3: se construye $I_{GB}^{t,q}(x, y)$, dado $I_{GB}^{t,q-1}(x, y)$ como sigue. Suponga que $v_{GB}^{i_{GB}}$ y $v_{GB}^{j_{GB}}$ denotan pixeles conectados por el q -ésimo borde $E_{GB_q} = (v_{GB}^{i_{GB}}, v_{GB}^{j_{GB}})$. Si $v_{GB}^{i_{GB}}$ y $v_{GB}^{j_{GB}}$ están en regiones diferentes en $I_{GB}^{t,q-1}(x, y)$ y $w_{GB}(E_{GB_q})$ (calculado con la ecuación (3.13)) es pequeño comparado con la diferencia interna de ambas regiones (obtenida con la ecuación (3.14)), entonces se fusionan las 2 regiones, de otra manera no se hace nada.

Paso 4: el paso 3 se repetirá para $q = 1, \dots, m_{gb}$, donde m_{gb} son todos los bordes existentes iniciales en $I_{GB}^{t,0}(x, y)$.

Paso 5: Finalmente, V4 genera la salida $I_{GB}^{t,m_{GB}}(x, y)$ cuando q alcanza su valor máximo m_{gb} . Debido a que q indica solamente un número de operaciones, se iguala $I_{GB}^t(x, y)$ a $I_{GB}^{t,m_{GB}}(x, y)$.

Una vez calculado el cuadro $I_{GB}^t(x, y)$, este se regresa a V1/V2 para realizar la extracción de las regiones de los componentes $I_a^t(x, y)$ e $I_b^t(x, y)$ que se encuentran en la misma posición que las regiones de $I_{GB}^t(x, y)$. Para ello, se definen R_{a,r,n_R,I_a}^t y R_{b,r,n_R,I_b}^t , que son regiones con los mismos subíndices que R_{r,n_R}^t , incluyendo también la información de los componentes de color.

Finalmente, se obtiene el histograma de las regiones R_{a,r,n_R,I_a}^t y R_{b,r,n_R,I_b}^t de los componentes a y b . Para esto, se definen los *bins* del histograma partiendo de los valores máximos y mínimos de los píxeles de los componentes $I_a^t(x, y)$ e $I_b^t(x, y)$ respectivamente. Estos valores se obtuvieron variando los niveles de color del espacio estándar RGB con punto blanco D65. El rango de valores que pueden tomar los píxeles de $I_a^t(x, y)$ es de $[-86.18, 98.24]$ y para los píxeles de $I_b^t(x, y)$ es de $[-107.86, 94.48]$. Con base en estos rangos, se definieron un total de 32 *bins* para el histograma definidos como N_I . De las regiones de R_{a,r,n_R,I_a}^t y R_{b,r,n_R,I_b}^t se generaron 2 histogramas por cada objeto segmentado dentro de cada cuadro, dados por:

$$h_a^{\Omega,r,t} = \frac{1}{N_p} \sum_{i_p=1}^{N_p} \delta \left[f \left(R_{a,r,n_R,I_a}^t - \Omega \right) \right] \quad \Omega = 1, \dots, N_I/2 \quad (3.18)$$

$$h_b^{\Omega,r,t} = \frac{1}{N_p} \sum_{i_p=1}^{N_p} \delta \left[f \left(R_{b,r,n_R,I_b}^t - \Omega \right) \right] \quad \Omega = 1, \dots, N_I/2 \quad (3.19)$$

donde $f(\cdot)$ es la función que asigna los valores de color de los píxeles de R_{a,r,n_R,I_a}^t y R_{b,r,n_R,I_b}^t al intervalo correspondiente en la distribución del histograma. Posteriormente se concatenan ambos histogramas con la siguiente ecuación:

$$h_T^{\Omega,r,t} = \{h_a^{\Omega,r,t}, h_b^{\Omega,r,t}\} \quad \Omega_T = 1, \dots, N_I \quad (3.20)$$

En la Figura 3.9 se muestra de manera ilustrativa cómo se obtienen los histogramas, en a) se muestra el cuadro original en b) una segmentación previa con GBS proveniente del área V4, en c) se muestra un objeto extraído del cuadro original, en d) se muestra el histograma de la componente a y en e) la de la b y en f) se juntan los histogramas ambas componentes.

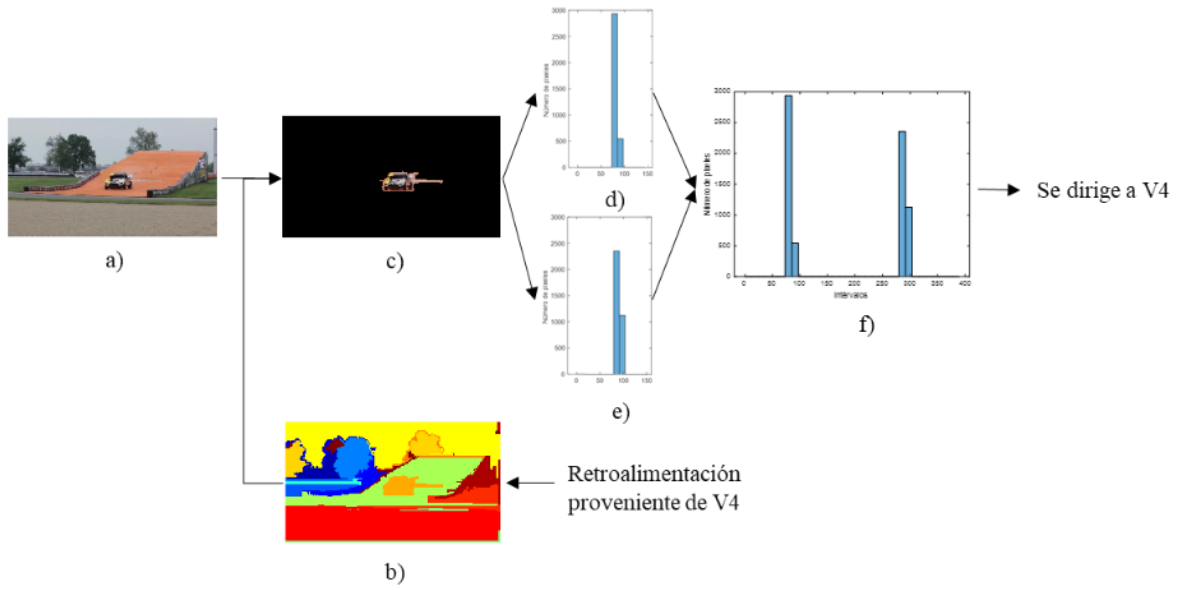


Figura 3.9. a) Imagen original $I_{RGB}^t(x, y)$, b) Imagen segmentada $I_{GB}^t(x, y)$, c) Extracción del objeto

$I_{GB}^{t,mGB}(x, y)$, d) histograma $h_a^{\Omega_T, r, t}$, e) histograma $h_b^{\Omega_T, r, t}$ y f) histograma total $h_T^{\Omega_T, r, t}$.

3.2.2. Orientación

3.2.2.1. Modelos neurocomputacionales de orientación

Biológicamente, las áreas V1/V2 de los mamíferos contienen CR bien definidos y restringidos a pequeñas regiones de espacio que se componen de neuronas selectivas a la orientación de los patrones contenidos en los estímulos visuales y se denominan células simples y complejas [76].

Las células simples responden a la orientación de formas y frecuencia espacial. A partir de estos conocimientos, surgen los filtros de Gabor [77], los cuales simulan el comportamiento de células simples sensibles a los patrones de orientación de estímulos visuales [76], y realizan un proceso como un sistema lineal calculando sumas ponderadas de la información proveniente de los NGL [40], que se puede definir con la ecuación (3.22) por medio de convolución entre los filtros de Gabor de la ecuación (3.21) y el cuadro de luminancia $I_L^t(x, y)$.

$$CS_{\theta}(x, y) = CRO_{\theta}(x_B, y_B) * I_L^t(x, y) \quad (3.21)$$

$$CRO_{\theta}(x_B, y_B) = \exp\left(-\frac{x'^2}{2\sigma_x^2} + \frac{y'^2}{2\sigma_y^2}\right) \cos\left(2\pi \frac{x'}{\lambda_B}\right) \quad (3.22)$$

$$+ \exp \left(-\frac{x'^2}{2\sigma_x^2} + \frac{y'^2}{2\sigma_y^2} \right) \sin \left(2\pi \frac{x'}{\lambda_B} \right) i_m$$

donde $x_B = 1, \dots, N_B$ y $y_B = 1, \dots, N_B$ definen la posición dentro del filtro, N_B es el tamaño en receptores del filtro, $x' = x \cos \theta + y \sin \theta$ y $y' = -x \sin \theta + y \cos \theta$, σ_x^2 y σ_y^2 son las varianzas, del eje vertical y horizontal respectivamente, λ determina la longitud de onda a la que es sensible y θ representa la orientación del filtro.

Por otro lado, las células complejas, que tienen como entrada la respuesta de las células simples, se simulan con una operación no lineal que suma las salidas de las convoluciones de los filtros de Gabor. En la ecuación (3.23) se define el funcionamiento de una célula compleja según [78].

$$\begin{aligned} CC_{\theta,di}(x,y) = & -20CC_{\theta,di}(x,y) \\ & + \left(7 - CC_{\theta,di}(x,y) \right) \left(CS_{\theta}(x,y) \left(1 + \left[\sum_{v_s} H_{Ec_{\theta,di}}(x,y) - H_{Ic_{\theta,di}}(x,y) \right] \right) \right. \\ & \left. + 0.5[CC_{\theta,di}(x,y) - 0.03] \right) - \left(1 + CC_{\theta,di}(x,y) \right) \left(C_{P_{\theta,di}}(x,y) + C_{S_{\theta,di}}(x,y) \right), \end{aligned} \quad (3.23)$$

donde el termino $H_{Ec_{\theta,di}}(x,y)$ describe una entrada excitatoria de conexiones de largo alcance en V1 a una célula compleja en la posición (x,y) , con orientación θ y disparidad di , mientras que $H_{Ic_{\theta,di}}(x,y)$ es la entrada inhibitoria. El índice v significa el lado ON u OFF que proviene la entrada de una célula bipolar. El termino $\sum_{v_s} H_{Ec_{\theta,di}}(x,y)$ suma la entrada de ambos lados $v_s = 1,2$ de la célula. $C_{P_{\theta,di}}(x,y)$ es una entrada en la misma posición y disparidad que otras células bipolares con diferentes orientaciones. El término $C_{S_{\theta,di}}(x,y)$ es una entrada de una competición espacial entre posición y orientación dentro de la disparidad.

3.2.2.2. Modelo de orientación en NISeg

Computacionalmente, NISeg implementa las células simples con la convolución de un banco de filtros de Gabor sobre un cuadro de video $I_L^{(t)}(x,y)$ para enfatizar bordes con una orientación definida. Estos filtros son diseñados a través de ondas senoidales multiplicadas por una función

Gaussiana compleja tal como se muestra en la Figura 3.10. Estos filtros estand dados por la ecuación (3.22).

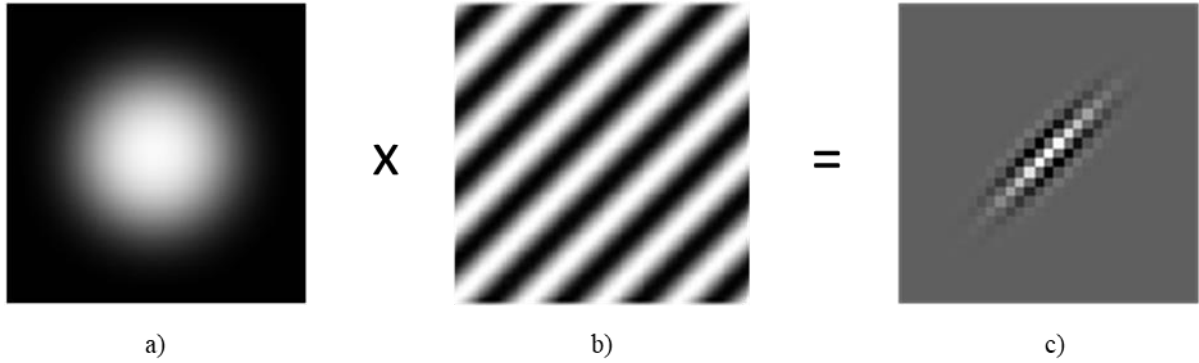


Figura 3.10. a) Señal Gaussiana, b) Señal senoidal, c) Filtro de Gabor.

donde $x_B = 1, \dots, N_B$ y $y_B = 1, \dots, N_B$ definen la posición dentro del filtro, N_B es el tamaño en pixeles del filtro, σ_x^2 y σ_y^2 son las varianzas, para en el eje vertical y horizontal respectivamente. Estas varianzas permiten a la función gaussiana tener una forma elíptica. λ_B es la longitud de onda de la señal senoidal y θ representa la orientación de la normal a las franjas paralelas del filtro de Gabor.

Con base en la ecuación (3.21), se diseñaron 4 filtros de Gabor cuyos valores de sus parámetros son: $\sigma_x^2 = 4$, $\sigma_y^2 = 36$ y $\lambda_B = 3$ y la dimensión de estos filtros es de $N_B = 31$. Con estos parámetros, los filtros son selectivos a orientaciones de bordes definidos con valores de θ de 0° , 45° , 90° y 135° . Esta cantidad de filtros fue seleccionada para obtener la mayor cantidad de información de bordes sin comprometer el costo computacional del procesamiento, estos filtros se muestran en la Figura 3.11.

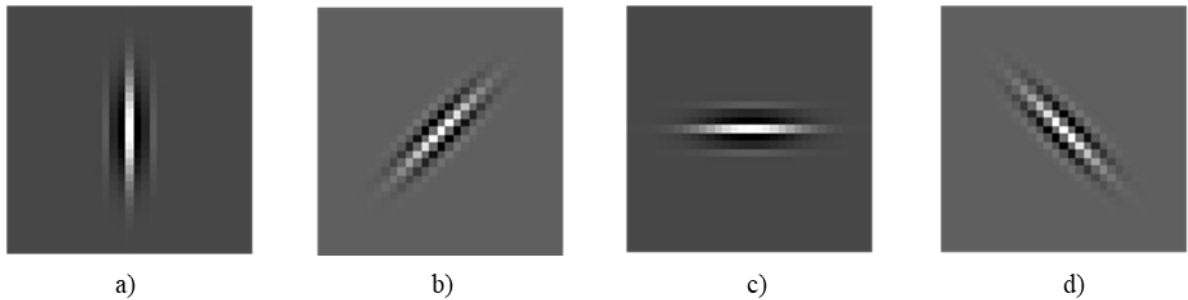


Figura 3.11. Filtros de Gabor de orientación a) de 0° b) de 45° c) de 90° y d) de 135° .

Para simular las células complejas en NISeg, se propuso una red CeNN para juntar las características de orientación y bordes obtenidas de cada uno de los 4 filtros de Gabor. Una red CeNN consiste en un arreglo rectangular $M_{CN} \times N_{CN}$ de elementos con coordenadas cartesianas (i_{CN}, j_{CN}) , donde $i_{CN} = 1, 2, \dots, M_{CN}$, y $j_{CN} = 1, 2, \dots, N_{CN}$. Cada elemento está definido por:

$$\begin{aligned} \dot{x}_{CN}^{i_{CN}, j_{CN}} = & -x_{CN}^{i_{CN}, j_{CN}} + \sum_{i_{CN}, j_{CN}; k_{CN}, l_{CN}} A^{i_{CN}, j_{CN}; k_{CN}, l_{CN}} y_{CN}^{k_{CN}, l_{CN}} \\ & + \sum_{i_{CN}, j_{CN}; k_{CN}, l_{CN}} B^{i_{CN}, j_{CN}; k_{CN}, l_{CN}} u_{CN}^{k_{CN}, l_{CN}} + z_{CN}^{i_{CN}, j_{CN}} \end{aligned} \quad (3.24)$$

donde $x_{CN}^{i_{CN}, j_{CN}}$, $y_{CN}^{k_{CN}, l_{CN}}$, $u_{CN}^{k_{CN}, l_{CN}}$ y $z_{CN}^{i_{CN}, j_{CN}}$ son su estado, salida, entrada y umbral respectivamente, $A^{k_{CN}, l_{CN}}$ y $B^{k_{CN}, l_{CN}}$ son las plantillas de retroalimentación y de entrada sináptica. Las sumatorias representan operaciones de convolución que simulan las células simples. La ecuación de salida estándar de la CeNN se muestra a continuación,

$$y_{CN}^{i_{CN}, j_{CN}} = f_{CN}(x_{CN}^{i_{CN}, j_{CN}}) = \frac{1}{2} |x_{CN}^{i_{CN}, j_{CN}} + 1| - \frac{1}{2} |x_{CN}^{i_{CN}, j_{CN}} - 1| \quad (3.25)$$

a esta ecuación también se le conoce como función de saturación. La condición de estado inicial se define a través de la ecuación (3.26).

$$x_{CN}^{i_{CN}, j_{CN}} = 0. \quad (3.26)$$

Para NISeg se utiliza la red CeNN de retroalimentación cero o *feedforward*, cuya particularidad estriba en que todos los elementos de la plantilla $A^{k_{CN}, l_{CN}}$ son cero. De manera que la ecuación (3.24) queda reducida a la que se muestra en (3.27), mostrando la convolución con el operador $*$, y también sustituyendo la plantilla $B^{k_{CN}, l_{CN}}$ con el filtro de Gabor que se calculó en la ecuación (3.22) y la entrada $u_{CN}^{i_{CN}, j_{CN}}$ con el cuadro de luminancia $I_L^t(x, y)$.

$$\dot{x}_{CN}^{x_B, y_B} = -x_{CN}^{x_B, y_B} + CRO_{\theta}(x_B, y_B) * I_L^t(x, y) + z_{CN}^{x_B, y_B}. \quad (3.27)$$

los índices i_{CN}, j_{CN} y k_{CN}, l_{CN} se cambiaron por de la plantilla de los filtros de Gabor x_B, y_B y del cuadro de video (x, y) , respectivamente.

La Figura 3.12 muestra el cálculo de la CeNN *feedforward*, donde se tiene la entrada $I_L^t(x, y)$, el estado $x_{CN}^{x_B, y_B}$, la función de activación $f_{CN}(\cdot)$ y la salida $y_{CN}^{x_B, y_B}$.

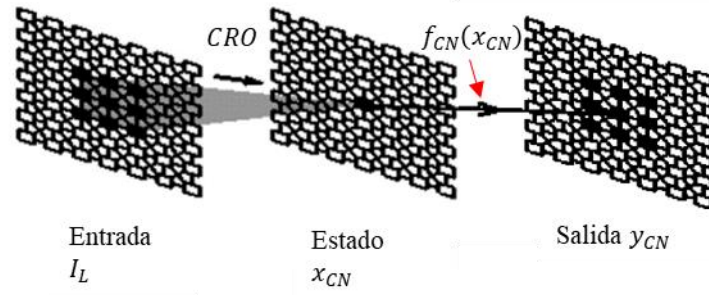


Figura 3.12. Proceso de la CeNN *feedforward* implementada.

El esquema del sistema de CeNN *feedforward* se representa en el diagrama de bloques que se muestra en la Figura 3.13, donde no hay retroalimentación de la señal de salida.

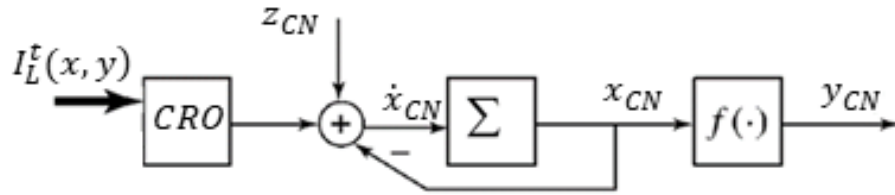


Figura 3.13. Estructura del sistema de una CeNN con retroalimentación cero [79].

La forma en que se utilizó la red CeNN en conjunto con los filtros de Gabor se explica en los pasos siguientes:

Paso 1: la entrada al algoritmo es el cuadro de luminancia $I_L^t(x, y)$. Se seleccionan los parámetros del filtro de Gabor para definir los valores de θ , en el siguiente orden: 0° , 45° , 90° y 135° y el umbral $z_{CN} = 0$.

Paso 2: con el valor de orientación definido en el paso 1 se utiliza la ecuación (3.22) para calcular la plantilla del filtro de Gabor $CRO_\theta(x_B, y_B)$.

Paso 3: se calcula el estado $\dot{x}_{CN\theta}^{x_B, y_B}$ con la ecuación (3.27).

Paso 4: se calcula la salida $y_{CN\theta}^{x_B, y_B}$ con la ecuación (3.25).

Paso 5: se van sumando las salidas conforme se van calculando en el paso 4 y una vez calculadas para las 4 diferentes orientaciones se obtiene $y_{CN}^{i_{CN}, j_{CN}}$ y se avanza al paso 6, si no, se regresa al 1.

Paso 6: para finalizar se elimina la bipolaridad de $yt_{CN}^{i_{CN},j_{CN}}$ debida a la función de saturación de la ecuación (3.25), resultando en el cuadro $I_{CN}^t(x, y)$, el cual contiene la información de bordes del cuadro $I_L^t(x, y)$.

Un ejemplo del resultado de este algoritmo se muestra en la Figura 3.14 donde se puede observar en c) que con las 4 orientaciones escogidas de los filtros de Gabor se consigue obtener la mayor cantidad de información de bordes sin comprometer el tiempo computacional.

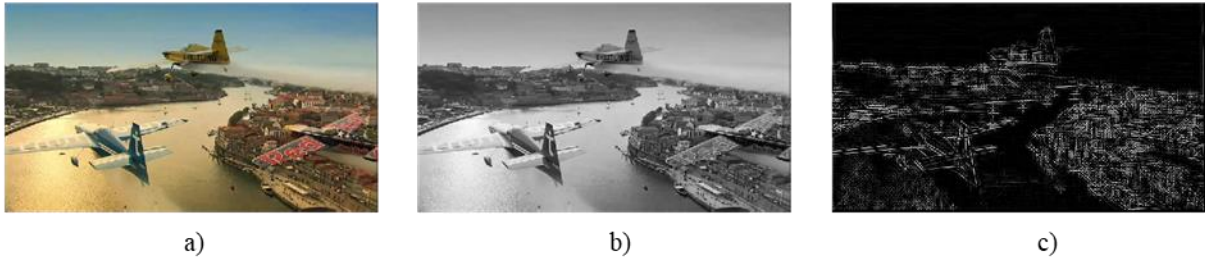


Figura 3.14. a) Imagen original $I_{RGB}^t(x, y, z)$, b) Imagen de luminancia $I_L^t(x, y)$, c) Imagen resultante $I_{CN}^t(x, y)$.

3.2.3. Movimiento

3.2.3.1. Modelos neurocomputacionales de movimiento

Biológicamente, la percepción de movimiento inicia en V1 mediante capas corticales de neuronas que son selectivas a la dirección ya que responden cuando un estímulo se mueve en una dirección particular dentro del CR [17]. Estas células son generalmente sintonizadas en orientación, pero responden únicamente a bordes de una orientación o inclinación particular mientras estos estímulos se mueven a través de sus CR [80]. A partir de estas neuronas, V2 logra codificar la rapidez y dirección de movimiento en varios puntos del escenario. Las células selectivas a dirección se concentran en V1/V2 y se proyectan principalmente a capas corticales encargadas del procesamiento de movimiento en MT.

3.2.3.2. Modelo de movimiento en NISeg

Computacionalmente, el proceso de percepción de rapidez y dirección del movimiento en NISeg se consigue con el flujo óptico, un concepto que se refiere a los desplazamientos entre los patrones de intensidad de los objetos observados. Esta definición se origina en una descripción fisiológica de la percepción visual del mundo a través de la formación de imágenes

en la retina. En ese sentido, el flujo óptico es causado por el movimiento relativo entre el observador y los objetos de la escena observada y representa el movimiento de las intensidades en el plano de la imagen, pero, no necesariamente explica el movimiento 3D real en la escena física. Esta definición de flujo óptico asume que la intensidad de los píxeles durante el movimiento permanece constante y la principal dificultad es hacer frente a la incertidumbre debido a la mala postura del problema llamado problema de apertura [81]. Para ejemplificar el flujo óptico, se muestra la Figura 3.15, donde los vectores representan el patrón de dirección de movimiento relativo entre el observador y el escenario. El observador se encuentra encima de un avión que está aterrizando.

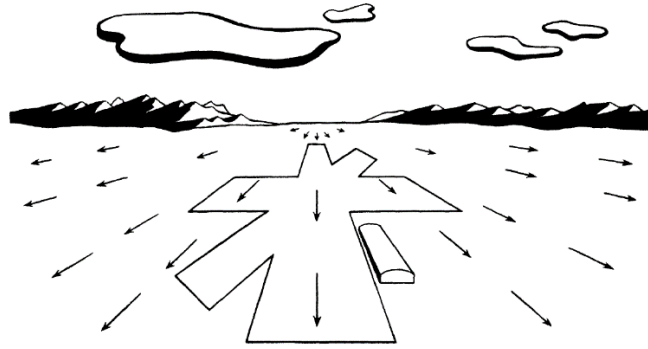


Figura 3.15. Ejemplo de expansión radial del flujo óptico [82].

Para obtener el flujo óptico, es necesario calcularlo con dos cuadros consecutivos para conocer sus diferencias en el tiempo y el espacio. Para este cálculo se utiliza el algoritmo de Lucas-Kanade [83], el cual resuelve la ecuación (3.28),

$$I_X(x, y)u_{LK}(x, y) + I_Y(x, y)v_{LK}(x, y) + I_T(x, y) = 0 \quad (3.28)$$

donde $I_X(x, y)$, $I_Y(x, y)$, e $I_T(x, y)$ son las derivadas del cuadro de luminancia espaciotemporal, $u_{LK}(x, y)$ es el flujo óptico horizontal y $v_{LK}(x, y)$ es el flujo óptico vertical.

Para resolver la ecuación (3.28) para $u_{LK}(x, y)$ y $v_{LK}(x, y)$, el método Lucas-Kanade divide la imagen original en pequeñas secciones y asume una velocidad constante en cada sección. Después, se realiza un ajuste de mínimos cuadrados ponderados de la ecuación (3.28) a un modelo constante para $[u_{LK}(x, y) \ v_{LK}(x, y)]'$ en cada sección definida como Ω_{LK} . El método consigue este ajuste minimizando la ecuación (3.29)

$$\sum_{x_{LK} \in \Omega_{LK}} W^2(x_{LK}) [I_X(x_{LK})u_{LK}(x_{LK}) + I_Y(x_{LK})v_{LK}(x_{LK}) + I_T(x_{LK})]^2 \quad (3.29)$$

donde $x_{LK} = (x_1, y_1)$ es un pixel dentro de una sección Ω_{LK} y $W(x_{LK})$ denota una función matricial que otorga más peso a los pixeles en el centro de cada sección. La solución para el problema de minimización se consigue a través de la ecuación (3.30).

$$\begin{aligned} & \begin{bmatrix} \sum_{x_{LK} \in \Omega_{LK}} W^2(x_{LK}) I_X^2(x_{LK}) & \sum_{x_{LK} \in \Omega_{LK}} W^2(x_{LK}) I_X(x_{LK}) I_Y(x_{LK}) \\ \sum_{x_{LK} \in \Omega_{LK}} W^2(x_{LK}) I_Y(x_{LK}) I_X(x_{LK}) & \sum_{x_{LK} \in \Omega_{LK}} W^2(x_{LK}) I_Y^2(x_{LK}) \end{bmatrix} \begin{bmatrix} u_{LK}(x, y) \\ v_{LK}(x, y) \end{bmatrix} \\ &= - \begin{bmatrix} \sum_{x_{LK} \in \Omega_{LK}} W^2(x_{LK}) I_Y(x_{LK}) I_T(x_{LK}) \\ \sum_{x_{LK} \in \Omega_{LK}} W^2(x_{LK}) I_X(x_{LK}) I_T(x_{LK}) \end{bmatrix}. \end{aligned} \quad (3.30)$$

Para obtener los valores de $u_{LK}(x, y)$ y $v_{LK}(x, y)$ se realizan los siguientes pasos, basándose en [84] y [85]:

Paso 1: calcular $I_{X_a}(x_{LK})$ e $I_{Y_a}(x_{LK})$ usando la plantilla $[-1 \ 8 \ 0 \ -8 \ 1]/12$ propuesta en [84] para $I_{X_a}(x_{LK})$ y su traspuesta para $I_{Y_a}(x_{LK})$, a través de la operación de convolución como se muestra en las ecuaciones (3.31) y (3.32). Estas plantillas son operadores diferenciales con los que se obtendrán los gradientes en el eje horizontal y vertical del cuadro actual.

$$I_{X_a}(x_{LK}) = \frac{[-1 \ 8 \ 0 \ -8 \ 1]}{12} * I_L^t(x, y) \quad (3.31)$$

$$I_{Y_a}(x_{LK}) = \frac{[-1 \ 8 \ 0 \ -8 \ 1]^{T_r}}{12} * I_L^t(x, y) \quad (3.32)$$

donde T_r indica la traspuesta de la matriz.

Paso 2: calcular $I_{T_a}(x_{LK})$ entre los cuadros $I_L^{t-1}(x, y)$ e $I_L^t(x, y)$ usando el filtro de diferencia $[-1 \ 1]$, cuya operación es restarle el cuadro anterior al cuadro actual, esto se realiza con la ecuación siguiente,

$$I_{T_a}(x_{LK}) = [-1 \ 1] \begin{bmatrix} I_L^{t-1}(x, y) \\ I_L^t(x, y) \end{bmatrix}. \quad (3.33)$$

Paso 3: Suavizar los componentes de los gradientes, $I_{X_a}(x_{LK})$, $I_{Y_a}(x_{LK})$, e $I_{T_a}(x_{LK})$, usando una plantilla separable e isotrópica de 5 x 5 cuyos coeficientes en una dimensión son $[1 \ 4 \ 6 \ 4 \ 1]/16$, esto se realiza con la ecuación siguiente,

$$I_X(x_{LK}) = I_{X_a}(x_{LK}) * \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \cdot \frac{1}{256}. \quad (3.34)$$

Lo mismo se aplica para $I_{Y_a}(x_{LK})$ e $I_{T_a}(x_{LK})$, resultando los cuadros $I_Y(x_{LK})$ e $I_T(x_{LK})$ respectivamente.

Paso 4: Resolver las ecuaciones para cada pixel usando el siguiente método

$$\text{Si } A_{LK} = \begin{bmatrix} a_{LK} & b_{LK} \\ b_{LK} & c_{LK} \end{bmatrix} = \begin{bmatrix} \sum W^2(x_{LK}) I_X^2(x_{LK}) & \sum W^2(x_{LK}) I_X(x_{LK}) I_Y(x_{LK}) \\ \sum W^2(x_{LK}) I_Y(x_{LK}) I_X(x_{LK}) & \sum W^2(x_{LK}) I_Y^2(x_{LK}) \end{bmatrix}, \text{ entonces}$$

$$\text{los eigenvalores de } A_{LK} \text{ son } \lambda_{i_{LK}} = \frac{a_{LK} + c_{LK}}{2} \pm \frac{\sqrt{4b_{LK}^2 + (a_{LK} - c_{LK})^2}}{2}; i_{LK} = 1, 2$$

Los eigenvalores son comparados al umbral, $\tau_{LK} = 0.0009$, que corresponde al valor de reducción de ruido. El resultado cae dentro de alguno de los siguientes casos:

Caso 1: $\lambda_{1LK} \geq \tau_{LK}$ y $\lambda_{2LK} \geq \tau_{LK}$. A_{LK} no es singular, el sistema de ecuaciones (3.30) se resuelve.

Caso 2: $\lambda_{1LK} \geq \tau_{LK}$ y $\lambda_{2LK} < \tau_{LK}$. A_{LK} es singular, los gradientes son normalizados para resolver el sistema de ecuaciones de (3.30).

Caso 3: $\lambda_{1LK} < \tau_{LK}$ y $\lambda_{2LK} < \tau_{LK}$. El flujo óptico de $u_{LK}(x, y)$ y $v_{LK}(x, y)$ es 0.

Paso 5: $u_{LK}(x, y)$ y $v_{LK}(x, y)$ se suman para obtener un flujo óptico total y a través de la ecuación (3.35) se obtiene el cuadro de magnitud $I_{FO}^t(x, y)$, cuyo valor se utilizará en posteriores procesamientos.

$$I_{FO}^t(x, y) = \|u_{LK}(x, y) + v_{LK}(x, y)\| \quad (3.35)$$

En la Figura 3.16 se muestra en a) y b) los cuadros $I_L^t(x, y)$ e $I_L^{t+1}(x, y)$ a los que se les aplicó el método de Lucas-Kanade para obtener su respectivo flujo óptico. En c) se muestra el resultado de aplicar el filtro de diferencia y en d) se muestra el cuadro de magnitud de flujo óptico. Los cuadros de c) y d) tienen cierta similitud a un cuadro de bordes ya que de las porciones homogéneas en luminancia no se obtiene información de movimiento.

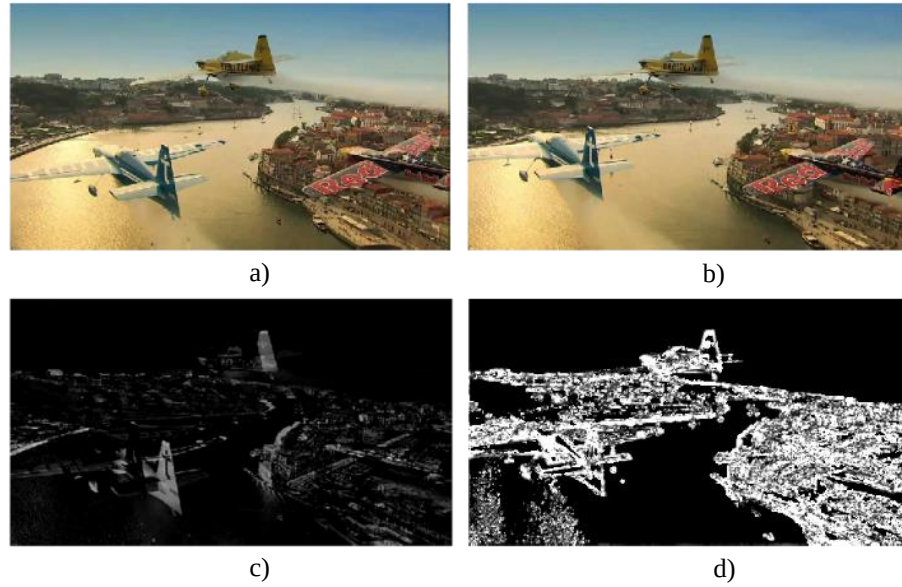


Figura 3.16. a) Primer cuadro $I_L^t(x, y)$, b) Segundo cuadro $I_L^{t+1}(x, y)$, c) Cuadro de diferencia temporal $I_T(x_{LK})$, d) Magnitud de flujo óptico resultante $I_{FO}^t(x, y)$.

3.3. Área MT

3.3.1. Modelo neurocomputacional del área MT

Biológicamente, el área MT se compone de neuronas que forman CR más amplios que los de V1/V2 y crecen conforme avanza la información de nivel jerárquico en el proceso visual. El crecimiento en el tamaño de los CR hace que sus propiedades de respuesta cambien de análisis locales y simples a análisis más complejos y globales. MT provee una siguiente etapa de análisis de movimiento de las señales provenientes de V1/V2 y se realiza lo siguiente:

Resolver el problema de apertura. El problema de apertura se refiere al hecho de que el movimiento de una estructura espacial como una barra o borde, no puede determinarse sin ambigüedades si se ve a través de una pequeña apertura, de modo que los extremos del estímulo no sean visibles, esto ejemplifica en la Figura 3.17 donde la dirección detectada del borde es diferente a la dirección real del movimiento. En el contexto de lo biológico, este problema lo enfrentan neuronas individuales con CR que solo toman muestras de una región espacialmente restringida de la imagen retiniana [86]. Para resolver esto la población de neuronas en MT

integra las componentes ortogonales de movimiento de varios bordes de un objeto y codifica la verdadera dirección de movimiento del objeto [80].

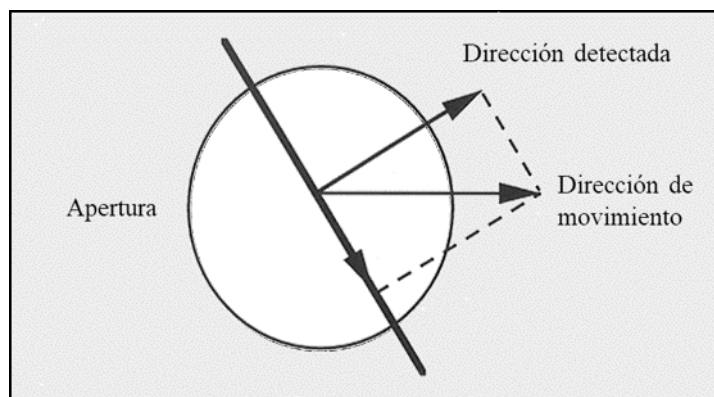


Figura 3.17. Ejemplo del problema de apertura

Supresión de movimientos oponentes. Mientras que los CR de V1/V2 responden solo a la dirección a la que son sensibles, las neuronas de MT, realizan los patrones de movimiento en común e inhiben aquellos patrones de movimiento que no son relevantes [80].

La supresión de movimientos oponentes es importante para remover ruido y construir superficies de regiones. Para entender un estímulo considerado como ruido por MT, en [80] se propone como ejemplo un resplandor, el cual puede ser considerado como movimiento en todas las direcciones a la vez y las neuronas en V1/V2 responden a estas diferentes direcciones. Mientras tanto, las neuronas del área MT responden muy poco o nada al resplandor debido a que suprimen su actividad ante las diferentes direcciones de movimiento. Esto muestra que MT está más vinculado a la percepción de movimiento que V1/V2. Otro procesamiento que realiza el área MT es segmentar objetos en movimiento. En [17] se menciona que existe evidencia que sugiere que el área MT desempeña un rol especial en la combinación de señales de movimiento en el espacio para representar superficies en movimiento.

3.3.2. Modelo de MT en NISeg para coherencia temporal

Para simular el procesamiento que realiza MT computacionalmente, en NISeg se propuso un algoritmo cuya función es conseguir coherencia temporal en la segmentación de objetos en secuencias de video, que se apoya de la información temporal y de las regiones formadas en V4 para resolver el problema de apertura y los movimientos oponentes.

La coherencia temporal sugiere que dos cuadros consecutivos posiblemente contienen objetos similares y por lo tanto, las clases no cambian, solo la posición de algunos objetos. Es posible que cada objeto en el video también este sujeto a pequeñas transformaciones, como la traslación, rotación o deformación en los cuadros vecinos [87].

Para lograr la coherencia temporal, se diseñaron dos algoritmos, uno para los objetos en movimiento y otro para los estáticos. El primero consiste en mantener un objeto que se encuentra en movimiento en la misma clase a lo largo de la secuencia de video. Para esto, se usa la información obtenida del flujo óptico y de una previa segmentación de objetos proveniente del área V4. En la Figura 3.18 se muestra un esquema del proceso que se sigue para conseguir la coherencia temporal y posteriormente se describen los pasos a seguir.

Paso 1: primeramente, se generan las máscaras binarias $MA_r(x, y)$ con las regiones de $I_{GB}^t(x, y)$ a través de la ecuación (3.36), después se multiplican con el cuadro $I_{FO}^t(x, y)$ para extraer las regiones $R_{FO_{r,n_R}}^t$, como se muestra en la ecuación (3.37).

$$MA_r(x, y) = \begin{cases} 1 & \text{si } I_{GB}^t(x, y) = r \\ 0 & \text{de otra forma} \end{cases} \quad (3.36)$$

$$R_{FO_{r,n_R}}^t = MA_r(x, y) \cdot I_{FO}^t(x, y) \quad (3.37)$$

Paso 2: una vez obtenidas las regiones $R_{FO_{r,n_R}}^t$, se realiza una binarización para conocer los pixeles con un movimiento considerable y descartar los que tengan movimientos irrelevantes, esto se consigue a través de la siguiente ecuación:

$$R_{B_{r,n_R}}^t = \begin{cases} 1 & \text{si } R_{FO_{r,n_R}}^t \geq \tau_B \\ 0 & \text{de otra forma} \end{cases} \quad (3.38)$$

donde τ_B es un umbral definido con un valor de 0.01, el cual se ajustó de manera experimental.

Paso 3: se calcula la cantidad de movimiento de las regiones de $R_{B_{r,n_R}}^t$ para conocer si el movimiento a nivel región es relevante, utilizando la siguiente ecuación:

$$P_B^{t,r} = \frac{1}{N_R} \sum_{n_R}^{N_R} R_{B_{r,n_R}}^t \quad (3.39)$$

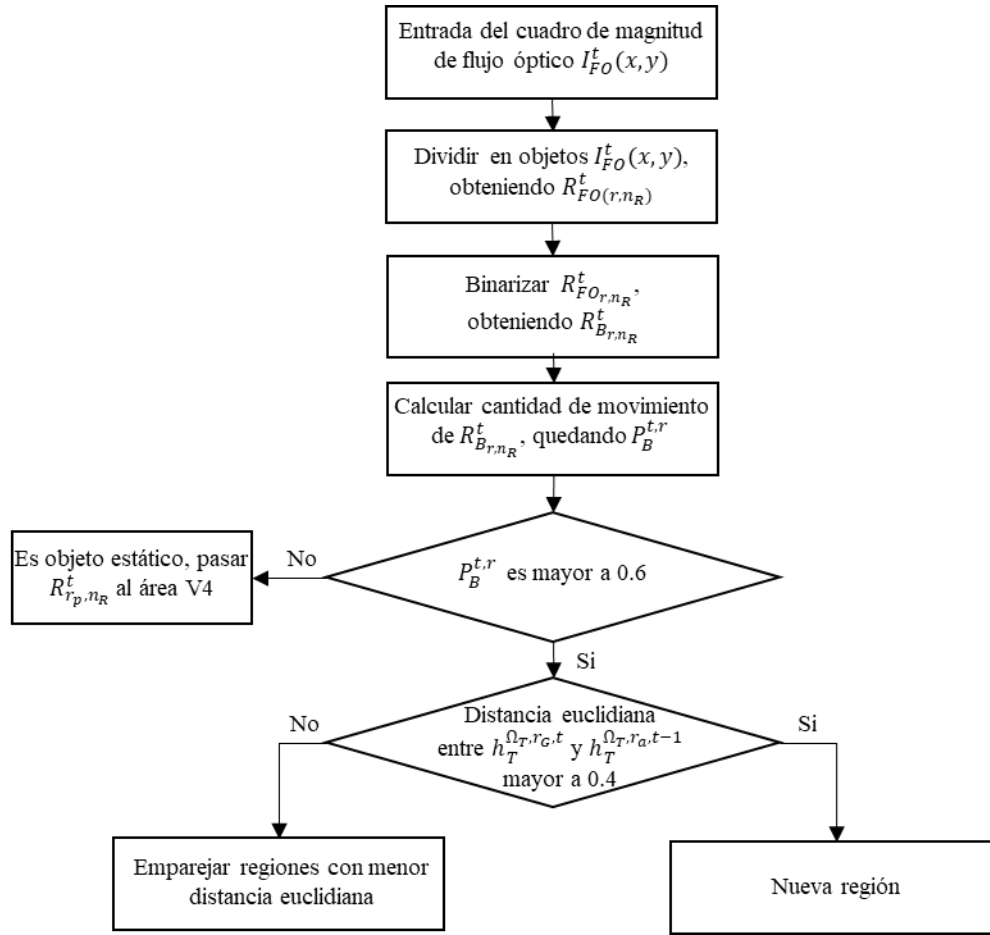


Figura 3.18. Proceso para obtener la coherencia temporal.

Paso 4: se evalúa el $P_B^{t, r}$ de cada región para conocer si son mayores o menores a un umbral establecido como 0.6. Si son menores, entonces el movimiento de estos objetos es pequeño, por lo que se toman como objetos estáticos. Los índices de estas regiones vienen dados por la ecuación siguiente:

$$r_p = \{r | P_B^{t, r} < 0.6\} \quad (3.40)$$

Entonces, el nuevo conjunto R_{r_p, n_R}^t se manda al algoritmo del área visual V4 como se muestra en el algoritmo de la Figura 3.18. Si las regiones son mayores al umbral, quiere decir que existe un movimiento considerable por lo que se toman como objetos en movimiento. En la ecuación (3.41) se extraen los índices de los objetos en movimiento para tenerlos definidos y utilizarlos más adelante.

$$r_G = \{r | P_B^{t,r} \geq 0.6\} \quad (3.41)$$

Paso 5: se obtienen las distancias euclidianas de los histogramas $h_T^{\Omega_T, r_G, t}$ de las regiones R_{r_G, n_R}^t y $h_T^{\Omega_T, r_a, t-1}$ de R_{r_G, n_R}^{t-1} (calculados con la ecuación (3.20)), donde $r_a = 1, \dots, N_a$ es el índice de las regiones anteriores y N_a el total de estas. Lo anterior se realiza con la ecuación siguiente:

$$d^{r_G, r_a, t} = \sqrt{\sum_{\Omega_T=1}^{N_I} (h_T^{\Omega_T, r_G, t} - h_T^{\Omega_T, r_a, t-1})^2} \quad (3.42)$$

Paso 6: se evalúa si existen distancias euclidianas $d^{r_G, r_a, t}$ menores a un umbral definido como 0.4, como se ilustra en la Figura 3.18. Se buscan todas estas distancias menores al umbral y se definen clases a las regiones actuales dependiendo de las regiones del cuadro anterior donde exista mayor traslape espacial. Si todas las distancias $d^{r_G, r_a, t}$ son mayores al umbral entonces se define una nueva región. Esto se realiza para generar regiones nuevas cuando entra un objeto nuevo a la escena y se obtienen con la siguiente ecuación:

$$r_{F1} = \begin{cases} r_G & d^{r_G, r_a, t} < 0.4 \\ N_a + 1 & d^{r_G, r_a, t} \geq 0.4 \end{cases} \quad (3.43)$$

quedando un conjunto de regiones definido como R_{r_{F1}, n_R}^t .

En el paso 6 se busca conservar en la misma clase a una misma región que se presenta a lo largo de una secuencia de video, comparando histogramas de color con distancias euclidianas, de esta manera se busca conseguir coherencia temporal entre regiones. En la Figura 3.19 se ilustra el proceso hasta el paso de la extracción de movimiento de una región, se obtiene la magnitud del flujo óptico en (b) y se realiza una binarización de sus valores en (c), también se realiza una segmentación utilizando GBS en (d), y se extrae una máscara para cada uno de los objetos en (e), en f) se obtiene una región que muestra en pixeles blancos la cantidad de movimiento del mismo.

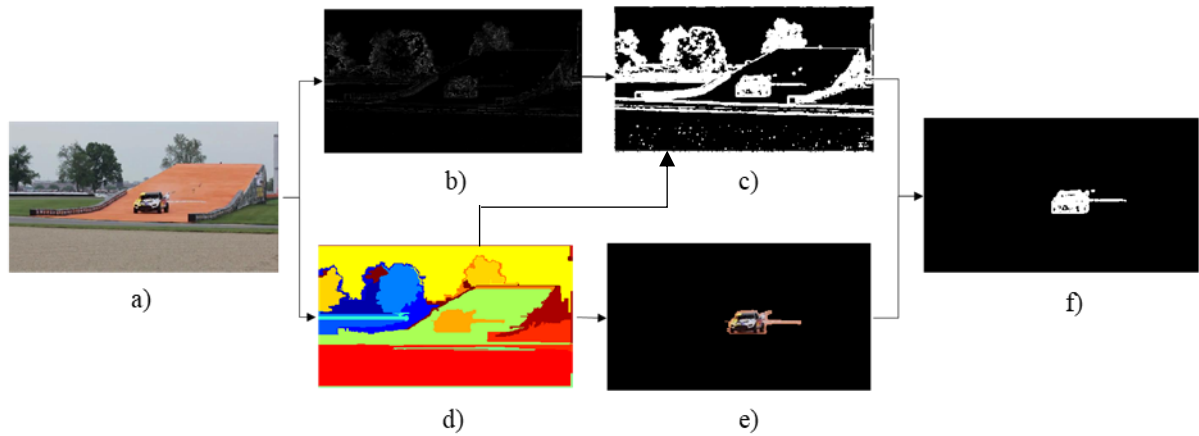


Figura 3.19. Proceso para la obtención la de movimiento de los objetos, a) Cuadro original $I_{RGB}^t(x, y, z)$, b) Magnitud de FO $I_{FO}^t(x, y)$, c) Binarización de magnitud del FO $R_B^t(r, n_R)$, d) Regiones de los objetos por GBS $I_{GB}^t(x, y)$, e) Extracción de uno de los objetos R_{Br1, n_R}^t , f) Movimiento del objeto.

3.4. Área V4

3.4.1. Modelo neurocomputacional del área V4

Biológicamente, V4 es un área del cerebro que se encuentra en la vía visual ventral y se especializa en el reconocimiento de objetos. Las neuronas de V4 responden selectivamente a orientación, color, disparidad y formas simples. Continúan el proceso de integrar características de bajo en nivel en otras de más alto nivel como, por ejemplo, contornos definidos por diferencias en rapidez y/o dirección de movimiento. También se menciona en [17] que V4 retroalimenta estas características a las áreas V1/V2.

Otras características que detecta V4, pero no las áreas V1/V2, son: selectividad de curvatura de contornos y la codificación invariante de la luminancia del tono. En lo referente a la curvatura de contornos, los psicólogos de la Gestalt han planteado la hipótesis de que las escenas visuales se agrupan perceptualmente en objetos, que son detectados y reconocidos a través de sus contornos, basándose en los principios de: similitud, proximidad, continuidad, destino común, simetría, convexidad, etc. Respecto a la codificación invariante de la luminancia del matiz, las neuronas selectivas al color en V4 se diferencian de las de V1/V2 en que codifican en base al matiz, y no en base a la oponencia de color. También, muchas de estas neuronas muestran una afinidad a la orientación [17][88][89].

En los trabajos [90][91] se menciona que V4 tiene neuronas que son selectivas a la forma de los objetos a través de sus contornos locales a lo largo del límite del objeto y también que las respuestas de la población de neuronas selectivas a texturas pueden explicar la discriminación y categorización humana de la textura.

3.4.2. Modelo de V4 en NISeg para coherencia espacial

Computacionalmente, para imitar algunas funciones de V4 en NISeg, se propone un método para conseguir coherencia espacial entre objetos, que se define como la calidad de la segmentación de un cuadro donde los píxeles de las regiones segmentadas pertenecen a su correspondiente objeto. Para conseguir esto, se desarrollaron dos algoritmos, que son: GBS (el que se presentó en la sección 3.2.1) que utiliza color y luminancia para realizar una agrupación de píxeles tomando en cuenta su vecindario y el otro es un mejoramiento de coherencia espacial tomando información de los bordes de los cuadros para ajustar los parámetros de GBS.

Con lo mencionado en la parte biológica, GBS se relaciona con el área V4 puesto que basa su funcionamiento en medir la evidencia de que existe un borde entre 2 regiones, comparando 2 cantidades: una basada en diferencias de color y luminancia a través del borde y la otra basada en las diferencias entre píxeles vecinos dentro de cada región, estas diferencias se calculan con la ecuación (3.13). La forma en que realiza el agrupamiento de píxeles permite categorizar los objetos con textura dentro de la misma clase, lo que beneficia la coherencia espacial de la segmentación.

El segundo algoritmo realiza un mejoramiento de la coherencia espacial sintonizando la función umbral $\tau_{GB}(R)$ de la ecuación (3.17) del algoritmo GBS a través de los parámetros $|R|$ y k_{GB} . Luego, se aplica un filtro de mejora de coherencia gaussiano con desviación estándar σ_c al cuadro $I_L^{(t)}(x, y)$. Para realizar este algoritmo se siguen los siguientes pasos:

Paso 1: se aplica un filtro de mejora de coherencia utilizando una función gaussiana bidimensional como la que se utilizó en la ecuación (2.3) para remover detalles irrelevantes y ruido de un cuadro. Basándose en los resultados de [72], se escogió una desviación estándar $D_o = 0.8$. El filtro se aplica como sigue:

$$G(u, v): I_L^t(x, y) \rightarrow I_c^t(x, y) \quad (3.44)$$

Paso 2: para obtener el umbral $\tau_{GB}(R)$ se define un valor de $|R| = 600$ valor obtenido experimentalmente.

Paso 3: el valor de k_{GB} se calcula utilizando la cantidad de bordes del cuadro $I_{CN}^t(x, y)$, siendo inversamente proporcional a este. Se propone el cálculo de k_{GB} con la ecuación (3.45), porque una gran cantidad de bordes indica que hay muchos objetos en el cuadro, al igual que lo indica una k_{GB} pequeña. La ponderación 0.8 se definió experimentalmente.

$$k_{GB} = -0.8 \left(\frac{2}{N} \sum_{(x,y)} I_{CN}^t(x, y) - 1 \right) \quad (3.45)$$

Paso 4: se calcula el valor del $\tau_{GB}(R)$ con los valores de $|R|$ y k_{GB} , obtenidos en los pasos 2 y 3, y se realiza una segmentación por GBS que se manda a las áreas V1/V2 para la obtención de los histogramas de color.

La operación que falta realizar es otorgar un índice a las regiones R_{r_p, n_R}^t que se tomaron como objetos estáticos en el ajuste temporal. Para esto se utiliza la ecuación (3.46) que otorga el índice de las región anterior R_{r_F, n_R}^{t-1} con mayor intersección en la posición de la región R_{r_p, n_R}^t

$$r_{F2} = \underset{r_F}{\operatorname{argmax}} \left(R_{r_p, n_R}^t \cap R_{r_F, n_R}^t \right) \quad (3.46)$$

donde r_F es un conjunto de números enteros que representan cada objeto del escenario. De la ecuación (3.46) se obtienen las regiones resultantes R_{r_{F2}, n_R}^t .

Para finalizar el algoritmo, se unen los conjuntos de regiones R_{r_{F1}, n_R}^t (calculadas en el módulo MT de movimiento) y R_{r_{F2}, n_R}^t quedando un conjunto R_{r_F, n_R}^t , como se muestra en la siguiente ecuación:

$$R_{r_F, n_R}^t = R_{r_{F1}, n_R}^t \cup R_{r_{F2}, n_R}^t. \quad (3.47)$$

Posteriormente, se transforma R_{r_F, n_R}^t a un cuadro con índices (x, y) .

Así, se obtiene el cuadro final $I_S^t(x, y)$ que contiene todos los objetos de un cuadro de video. En la Figura 3.20 se presenta un ejemplo de la implementación de método neuroinspirado propuesto a un cuadro de video, en a) está el cuadro original, y de manera sobrepuesta a este cuadro se muestra en b) el *ground truth* correspondiente y en c) la segmentación final $I_S^t(x, y)$.

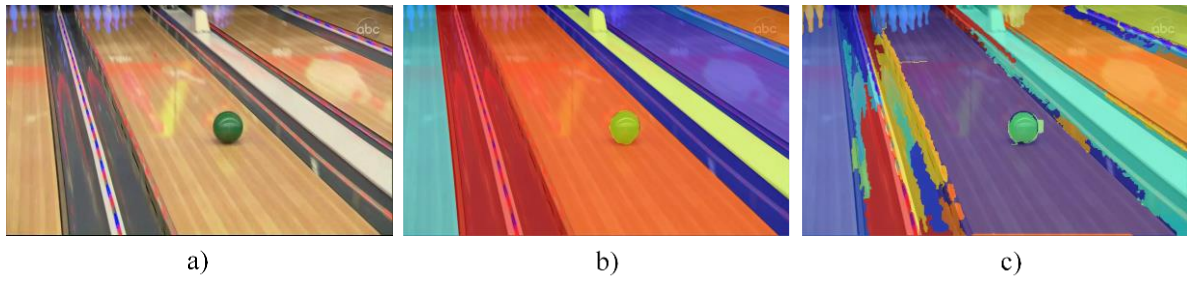


Figura 3.20. a) Cuadro original $I_{RGB}^t(x, y, z)$, b) *Ground truth*, c) Cuadro $I_S^t(x, y)$.

CAPÍTULO IV

RESULTADOS Y CONCLUSIONES

Este capítulo presenta un análisis de los resultados del desempeño de los algoritmos DObjA y NISeg. Para realizar dicho análisis, se hizo una búsqueda de diversas bases de datos de segmentación de objetos y se encontraron Cityscapes, Mapillary, CamVid y VSB100. Cityscapes, Mapillary y CamVid se diseñaron para segmentación semántica, contienen diversos videos de paisajes urbanos y la métrica de estas bases de datos mide coherencia espacial. Por otro lado, VSB100 tiene 100 videos con diferentes tipos de escenarios, toma en consideración todos los objetos del escenario y utiliza dos métricas que proporcionan una evaluación objetiva sobre la calidad de la segmentación de objetos en el espacio y el tiempo. Por estas razones, VSB100 se eligió para evaluar el funcionamiento de los métodos propuestos en esta tesis.

Para presentar los resultados, este capítulo se organiza de la siguiente forma: la sección 4.1 presenta la base de datos utilizada para la evaluación de los algoritmos. La sección 4.2 muestra los resultados obtenidos utilizando las dos métricas de evaluación sobre la coherencia espacial y temporal. Finalmente, la sección 4.3 reporta las conclusiones obtenidas de los resultados y del trabajo realizado.

4.1. Base de datos VSB100

La base de datos *Video Segmentation Benchmark* (VSB100) proporciona un conjunto de secuencias de video con su correspondiente *ground truth* y dos métricas de evaluación que pueden generalizar sobre subproblemas de segmentación de video, como:

- Objetos en movimiento/estáticos: son aquellos que pueden presentar o no un cambio de posición en el espacio a través del tiempo.
- Objetos rígidos/no rígidos: son aquellos que pueden presentar o no una deformación a través del tiempo.
- Videos tomados con una cámara en movimiento/estática: Son aquellos videos que pueden presentar o no movimiento generado por la cámara.

Esta base de datos contiene las secuencias de video de [92], pero VSB100 se extiende la segmentación a múltiples cuadros y a múltiples anotaciones humanas por cuadro. Esto permite dos propiedades importantes en las métricas de evaluación:

- Penalización de las segmentaciones temporalmente inconsistentes: segmentaciones que no cumplen con la definición de coherencia temporal.
- Tomar en cuenta la ambigüedad de segmentaciones correctas: la ambigüedad se refiere a la escala de segmentación entre elementos de escena, objetos y sus partes. Por ejemplo, cuando se segmenta una persona el término de segmentación se puede referir a la persona como un todo o a cada una de sus partes. Esto se aborda mediante múltiples anotaciones humanas para medir el nivel natural de ambigüedad.

Además de las métricas, las secuencias de video en VSB100 tienen las siguientes propiedades:

- El número de cuadros por video está limitado a un máximo de 121.
- La resolución de los cuadros es de 1280x720.
- Son 100 secuencias de video distribuidas en 40 de entrenamiento y 60 de prueba.
- Las anotaciones de video son precisas en los contornos de los objetos (coherencia espacial) y también en la consistencia temporal (coherencia temporal).
- No hay un tema específico para los tipos de escenario en los videos.

4.1.1. Métricas de evaluación

A continuación, se describen las 2 métricas a utilizar para evaluar el desempeño de los algoritmos desarrollados. Una de las métricas se llama *Boundary Precision-Recall* (BPR) orientada a medir la coherencia espacial y la otra se llama *Volume Precision-Recall* (VPR) enfocada a medir la coherencia temporal.

Métrica BPR

La métrica de evaluación BPR se basa en los contornos de los objetos, que son los límites que representan un cambio en los píxeles de un objeto a otro. Esta métrica evalúa la curva

precision-recall, que es una curva paramétrica que captura el equilibrio entre precisión y ruido a medida que varía el umbral de un detector de contornos.

La precisión (traducción de *precision* al español) es la fracción de detecciones que son verdaderos positivos en lugar de falsos positivos y está definida por:

$$PR = \frac{|SE \cap (\cup_{n_t=1}^{N_t} Gt_{n_t})|}{|SE|} \quad (4.1)$$

donde SE es el conjunto de segmentaciones generadas por los tres algoritmos de DObjA o NISeg, $\{Gt_{n_t}\}_{n_t=1}^{N_t}$ es el conjunto de anotaciones de bordes y $n_t = 1, \dots, N_t$ el índice de estos conjuntos.

La sensibilidad (traducción de *recall* al español) es la fracción de verdaderos positivos que se detectan en lugar de perderse y se define como:

$$RE = \frac{\sum_{n_t=1}^{N_t} |SE \cap Gt_{n_t}|}{\sum_{n_t=1}^{N_t} |Gt_{n_t}|} \quad (4.2)$$

Con las ecuaciones (4.1) y (4.2) se describe la calidad de los contornos de los objetos. Para tener una medida de coherencia espacial, se combina la precisión y la sensibilidad para obtener la medida llamada *f-measure*, la cual se muestra en la ecuación siguiente:

$$Fm = \frac{2(PR \times RE)}{RE + PR} \quad (4.3)$$

El Fm calculado con las ecuaciones (4.1), (4.2) y (4.3) se le conoce como BPR. Esta métrica es útil para medir la calidad de la segmentación para cada cuadro de video de manera independiente, por lo que solo evalúa la coherencia espacial. La evaluación de BPR se basa en los contornos de los objetos que definen su forma geométrica. Para mostrar un ejemplo de cómo evalúa las segmentaciones esta métrica se muestra la Figura 4.1, teniendo en a) el cuadro original y en b) y c) 2 diferentes segmentaciones. En b) se tiene un valor de BPR más alto que c) por mostrar una mejor coherencia espacial.

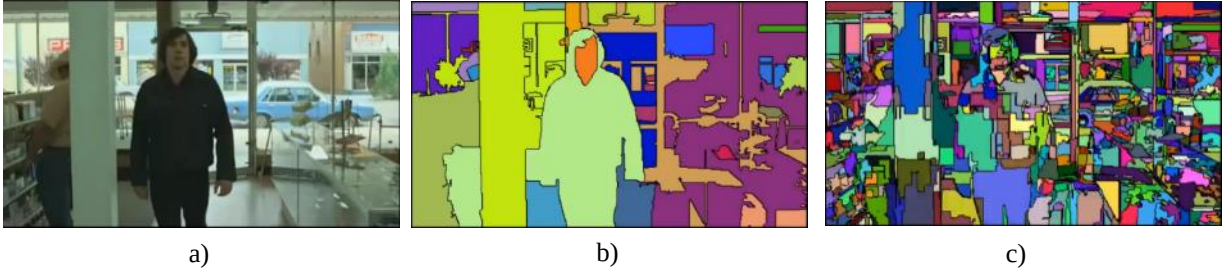


Figura 4.1. a) Cuadro original $I_{RGB}^t(x, y, z)$, b) Segmentación, c) Sobresegmentación.

Métrica VPR

La métrica VPR utiliza los volúmenes (segmentaciones espacio-temporales) de las segmentaciones generadas SE y las N_t anotaciones de las segmentaciones humanas Gt_{n_t} para dar una medición del traslape de los pixeles que pertenezcan a la misma clase entre ambos volúmenes.

Un problema que se tendría por usar las ecuaciones (4.1) y (4.2) para obtener VPR es el que generan las segmentaciones degeneradas (aquellas que toman un solo segmento en un cuadro o que cada pixel sea un segmento separado para todo el tiempo del video) que consiguen un puntaje más alto del que debería. Por lo cual en [93] enfrentan esto con una correcta normalización donde los volúmenes conseguidos por segmentaciones degeneradas son sustraídos del puntaje de traslape de la siguiente manera:

$$PR = \frac{\sum_{n_t=1}^{N_t} \left[\left\{ \sum_{se \in SE} \max_{gt \in Gt_{n_t}} |se \cap gt| \right\} - \max_{gt \in Gt_{n_t}} |gt| \right]}{N_t |R| - \sum_{n_t=1}^{N_t} \max_{gt \in Gt_{n_t}} |gt|} \quad (4.4)$$

$$RE = \frac{\sum_{n_t=1}^{N_t} \sum_{gt \in Gt_{n_t}} \left\{ \max_{se \in SE} |se \cap gt| - 1 \right\}}{\sum_{n_t=1}^{N_t} \left\{ |Gt_{n_t}| - \Gamma_{Gt_{n_t}} \right\}} \quad (4.5)$$

donde SE es el conjunto de segmentaciones espacio-temporales, $\{Gt_{n_t}\}_{n_t=1}^{N_t}$ es el conjunto de *ground truths* y $\Gamma_{Gt_{n_t}}$ es el número de volúmenes en Gt_{n_t} . Finalmente, VPR se obtiene con la ecuación (4.3) utilizando PR de la ecuación (4.4) y RE de la ecuación (4.5).

Para ilustrar el funcionamiento de VPR se muestra la Figura 4.2 donde los cuadros a) y d) pertenecen a una misma secuencia de video, pero son de diferente tiempo. b) y e) muestran una correcta segmentación donde las clases de los objetos se mantienen a través del tiempo

mostrados con el mismo color. En c) y f) se muestra una falta de coherencia temporal donde las segmentaciones se realizan de manera independiente en cada cuadro lo que ocasiona clases inconsistentes. Con lo comentado, los cuadros b) y e) tendrían un valor más alto de VPR que c) y f).

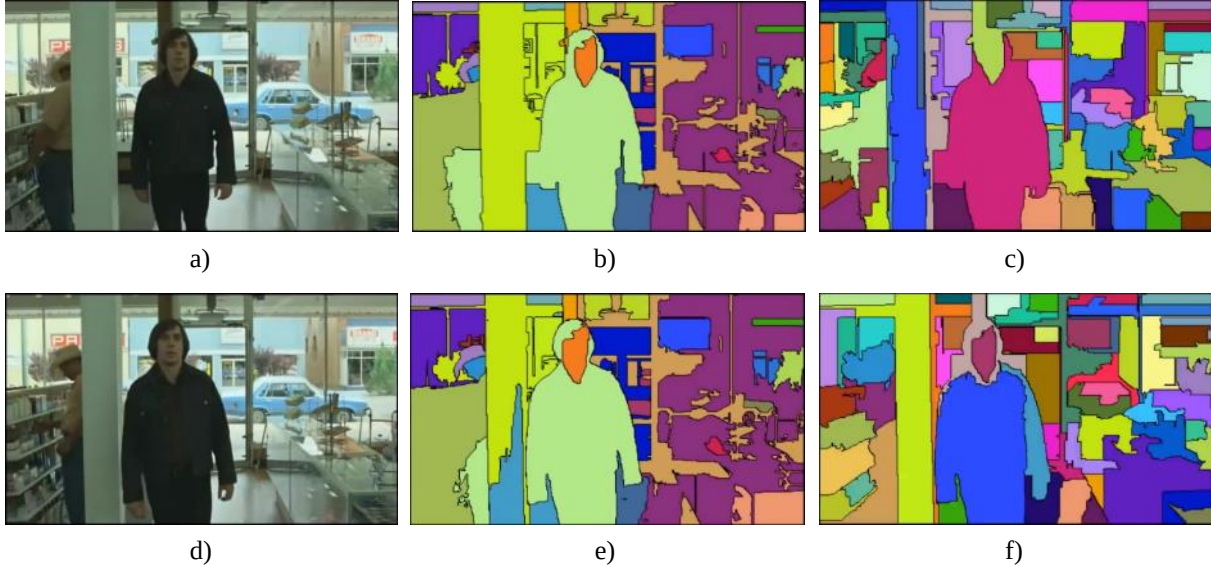


Figura 4.2. VPR explicada por coherencia temporal con los cuadros. a) $I_{RGB}^t(x, y, z)$, b) Segmentación deseada en t , c) Segmentación incorrecta en t , d) $I_{RGB}^{t+1}(x, y, z)$, e) Segmentación deseada en $t + 1$, f) Segmentación incorrecta en $t + 1$.

Propiedades de las métricas BPR y VPR

En [93] definen las propiedades que tienen las métricas BPR y VPR que cualquier métrica de evaluación de coherencia espacial y temporal debería cumplir:

- **No degeneración:** las medidas son bajas para segmentaciones degeneradas.
- **No suponen nada sobre la generación de datos:** las métricas no asumen un cierto número de clases y, por lo tanto, se aplican a casos en los que el número de etiquetas calculado es diferente del *ground truth*.
- **Múltiples anotaciones humanas:** se define como la propiedad de las métricas de poder utilizar múltiples anotaciones humanas para hacer frente a la inconsistencia al decidir el número de clases.

- **Acomodación adaptativa de la escala de segmentación:** las segmentaciones que abordan diferentes escalas no se penalizan, especialmente si esto se refleja en comparación con los *ground truths*. Esta propiedad se basa directamente en los humanos, que perciben psicológicamente las mismas escenas con un nivel de detalles diferente.
- **Sobresegmentación y enfoque en objetos:** en el enfoque de sobresegmentación, se tiene una alta precisión en BPR y una alta sensibilidad en VPR mientras que para el enfoque en objetos (métodos de segmentación que tienden a producir pocos volúmenes de objetos) se tiene una alta precisión en VPR y una alta sensibilidad en BPR. Ambos casos se pueden comparar de manera justa a través de la medida *f-measure*.
- **Medidas compatibles:** ambas métricas permiten comparar los resultados del mismo algoritmo en diferentes videos y el resultado de diferentes algoritmos en el mismo conjunto de videos.
- **Consistencia temporal:** las etiquetas de los objetos que no sean consistentes a lo largo del tiempo son penalizados por la métrica VPR.

4.2. Resultados con VSB100

En la Tabla 4.1 se muestran valores de BPR y VPR obtenidos de algunos algoritmos de la literatura, que abarcan varios enfoques de trabajo partiendo de una metodología no supervisada. Los algoritmos en la tabla se ordenaron según su desempeño en la métrica BPR. A continuación, se describen brevemente los métodos.

Keuper *et al.* en [94] proponen un método basado en información mutua puntal (IMP) de voxels espacio-temporales que mide su asociación estadística. Se establece la consistencia temporal mediante una evaluación de las afinidades puntuales basadas en IMP en el dominio espectral espacio-temporal. Por lo tanto, el método propuesto es independiente de cualquier cálculo de flujo óptico o modelos de movimiento previamente aprendidos. Xu *et al.* en [95] desarrollan un método llamado superpíxeles de correlación mejorada para la segmentación de video bajo el esquema de descomposición de bajo rango, que consiste en la exploración de afinidad entre superpíxeles. El algoritmo se divide principalmente en 2 pasos, análisis de características para establecer la afinidad inicial entre los superpíxeles, seguido de la construcción de un superpíxel de correlación mejorada que se consigue a través de una

combinación lineal. Zhao *et al.* en [96] extienden la técnica de agrupación espectral basada en optimización iterativa (SCIO por sus siglas en inglés) al aprendizaje multitarea (mSCIO) que es un aprendizaje en el que se unen características de bajo nivel para realizar el entrenamiento. También se desarrolla otro método que proporciona una solución para la agrupación multitarea mediante la extensión del método Nystrom (NysNcut) que es un método que consiste en la aproximación numérica de vectores propios de una matriz de afinidad ponderada construida con pares de píxeles de una imagen. Ochs y Brox proponen en [97] un método variacional para obtener segmentaciones de grupos de trayectoria dispersos. La información se propaga a través de un proceso de difusión jerárquico y no lineal que se ejecuta en el dominio continuo, tomando en cuenta superpíxeles. Corso *et al.* proponen en [98] un método nuevo para la segmentación automática de imágenes heterogéneas a través de una formulación bayesiana que genera un modelo que se integra al método de segmentación mediante un algoritmo de agregación ponderado, y se aplica a la tarea de detectar y segmentar tumores cerebrales y edemas en volúmenes de resonancia magnética multicanal. Galasso *et al.* desarrollan en [99] un método de segmentación de superpíxeles basado en cuadros combinando algunas características de movimiento y apariencias. También, se muestra un segundo algoritmo de segmentación jerárquica de imágenes con reconocimiento de movimiento para mejorar su enfoque en secuencias de video. Xu *et al.* abordan en [100], el problema del tiempo computacional que se requiere para procesar todos los voxels de una secuencia de video a través de una segmentación jerárquica de video utilizando GBS, donde cada cuadro de video se procesa una sola vez sin cambiar la segmentación de cuadros anteriores. Arbeláez *et al.* presentan en [101] un detector de contornos que combina múltiples características locales basándose en agrupación espectral, para después utilizarlo en un algoritmo de segmentación de imágenes que consiste en transformar la salida del detector de contornos en un árbol de región jerárquica. De esta forma, se reduce el problema de segmentación de imagen al de detección de contornos. Grundmann *et al.* presentan en [102] una técnica escalable para segmentación espaciotemporal de secuencias de video, utilizando un algoritmo jerárquico basado en gráficos. La técnica comienza por segmentar en exceso un grafo de video volumétrico en regiones espaciotemporales, luego se construye un gráfico sobre las regiones obtenidas y repiten este proceso múltiples veces para crear árboles de segmentaciones espaciotemporales.

Tabla 4.1. Desempeño de diferentes algoritmos de segmentación de objetos

Algoritmo	Métricas	
	BPR	VPR
Criterio humano	0.81	0.83
Oracle - Arbelaez et al. [101]	0.62	0.65
Arbelaez et al. [101]	0.62	0.26
MS TW [94]	0.56	0.64
ESP [95]	0.54	0.55
NISeg	0.51	0.47
Galasso et al. [99]	0.51	0.45
DObjA_FCM	0.47	0.25
Grundmann et al. [102]	0.47	0.52
Corso et al. [98]	0.47	0.51
DObjA_K-means	0.43	0.25
Xu et al. [100]	0.38	0.45
DObjA_EM	0.36	0.21
mSCIO [96]	0.31	0.39
NysNcut [96]	0.29	0.35
Ochs and Brox [97]	0.17	0.25

4.2.1. Análisis de los resultados

En la Tabla 4.1 se puede notar que en la primera posición se encuentra el criterio humano y para el caso de la métrica BPR, se refleja la capacidad humana de localizar los contornos de los objetos del escenario y VPR muestra la habilidad de identificar y agrupar objetos consistentemente en el tiempo.

Los 2 algoritmos de Arbelaez et al. [101] realizan la segmentación utilizando un árbol de región jerárquica partiendo de una detección de contornos obtenidos por agrupamiento espectral. El desempeño en BPR es muy bueno pero en VPR para conseguir un buen valor recurren a utilizar un predictor al que llaman Oracle para conectar las segmentaciones espaciales a través de volúmenes a lo largo del video. Este predictor tiene la gran desventaja de que necesita contar con los *ground truths* para funcionar, inconveniente que no se presenta en NISeg.

El tercero, MS TW, utiliza la información IMP entre voxeles en el dominio espectral para realizar la segmentación sin utilizar algún cálculo de flujo óptico o de algún modelo de movimiento previamente aprendido, también es necesario definir manualmente la cantidad de valores propios a calcular según la estructura de los videos de la base de datos a utilizar. NISeg, por el contrario, si utiliza la información obtenida del flujo óptico, lo que permite tener un mejor entendimiento de las características utilizadas para la segmentación espacio-temporal y también tiene la ventaja de que sus parámetros se ajustan de manera automática.

El cuarto, ESP, realiza una segmentación basándose en la afinidad entre superpíxeles, lo que reduce el costo computacional comparado con otros modelos basados en afinidad a nivel pixel. La desventaja de este algoritmo es que es necesario ajustar manualmente el parámetro encargado de controlar la magnitud de la combinación lineal entre superpíxeles para evitar problemas de baja o sobre segmentación.

En quinto lugar, queda NISeg que sigue un proceso neuroinspirado para extraer y utilizar características de bordes color y movimiento para realizar la segmentación, a diferencia de cualquier otro algoritmo.

El sexto, Galasso et al. [99], tiene resultados muy similares a NISeg en BPR y en VPR. Esto puede deberse a que se utilizan las mismas características que NISeg para realizar la segmentación, que son: características de apariencia (color, brillo y textura) y características de movimiento. La diferencia es que se usó agrupamiento espectral para realizar la segmentación.

Posteriormente siguen 3 algoritmos con mismo puntaje en BPR: FCM de DObjA, después sigue [102] y [98]. DObjA con FCM utiliza la información de luminancia para agrupar, y bordes para determinar el número de grupos. El de [65] queda bien en VPR a comparación con los demás algoritmos por utilizar un GBS de 3 dimensiones (tomando el tiempo como tercera dimensión), pero no tiene buen BPR y requiere un mayor procesamiento y más cantidad de memoria debido a que debe guardar la información de varios cuadros a la vez, ya que el GBS es espacio-temporal. El algoritmo de Corso et al. [98] obtuvo un valor alto en VPR por utilizar la información de voxeles individuales e información local a diferentes escalas. Tanto [98] como [102], tienen mal desempeño en BPR porque debido a que se deben ajustar las escalas de segmentación al cambiar de video.

En los últimos 6 lugares se encuentra *K-means* y EM de DObjA, que utilizan la misma información de FCM para agrupar. Se tiene Xu et al. [100] que utiliza un agrupamiento basado en GBS. También se tienen 2 algoritmos de agrupación espectral mSCIO y NysNcut. En último lugar se localiza el algoritmo de Ochs and Brox [97] que uso un método variacional.

En lo referente a las tres versiones de DObjA, FCM tuvo los mejores resultados, esto es consistente con [103], donde se compara el desempeño de FCM y *K-means* utilizando la base de datos de segmentación Berkeley. De acuerdo a los resultados en esa comparación, FCM es mejor que *K-means* en términos de producir mejores segmentos de color en imágenes con ruido FCM. El bajo rendimiento de EM puede deberse principalmente a la inicialización aleatoria de sus parámetros y a que el módulo determinación de número de grupos de DObjA le detecto menos grupos en promedio.

En la Figura 4.3 se muestra un ejemplo del funcionamiento de los algoritmos propuestos comparándose con su *ground truth* correspondiente. Se puede apreciar que Dobja y NISeg resultaron tener buenos desempeños en regiones homogéneas como en la duela, mientras que NISeg también se desempeñó bien en regiones como en la bola de boliche, a diferencia de DObjA que segmentó varias regiones dentro de la misma. Esto último influyo en la diferencia que existe en la métrica de BPR entre ambos algoritmos.

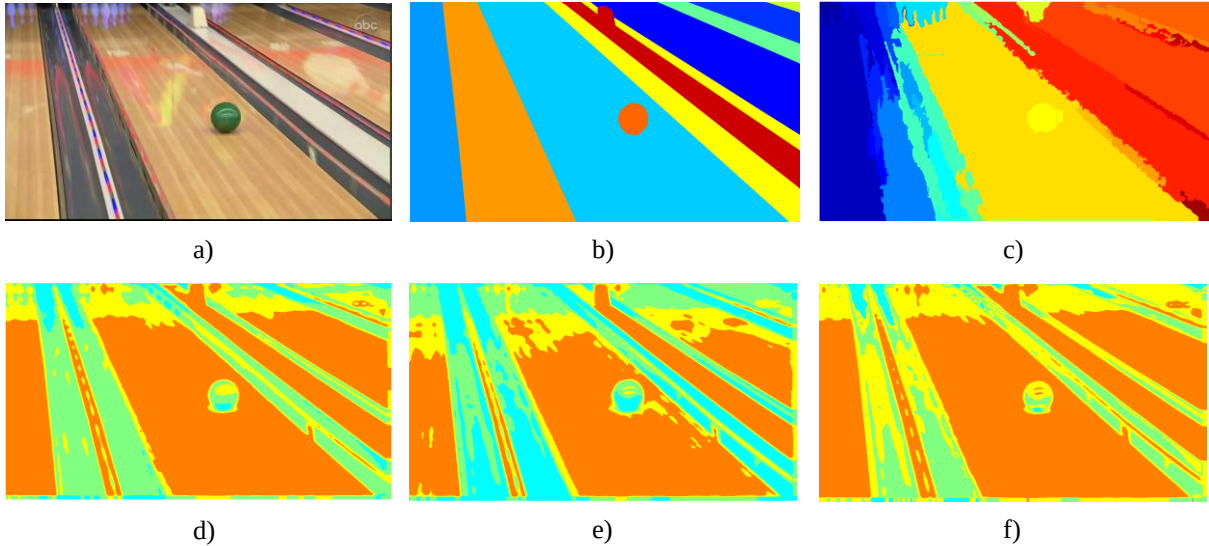


Figura 4.3. a) Cuadro original $I_{RGB}^t(x, y, z)$, b) *Ground truth*, c) Segmentación con NISeg $I_S^t(x, y)$, d) *K-means* $I_{A_1}^t(x, y)$, e) FCM $I_{A_2}^t(x, y)$ y f) EM $I_{A_3}^t(x, y)$.

Otra ventaja de NISeg sobre DObjA se da por el similar número de regiones con respecto al *ground truth*, debido a que hay menos ruido dentro de cada región. Esto se consiguió porque se utiliza más información para realizar la segmentación, como: color, bordes y movimiento, también por considerar la información espacial de los cuadros y no solo a nivel pixel, y finalmente por definir diferentes clases a las regiones que no están conectadas. Los algoritmos de DObjA salieron bajos en VPR básicamente por no tomar información temporal en consideración.

4.3. Conclusiones

Se desarrollaron dos algoritmos para la segmentación de objetos, DObjA y NISeg. DObjA se basó en los resultados obtenidos de experimentos con señales EEG y se divide en 4 etapas: preprocesamiento, extracción de características, agrupamiento y una etapa de determinación de número de grupos. Además, tiene 3 versiones basadas en la forma de realizar el agrupamiento: *K-means*, FCM y EM. Por otro lado, NISeg se basó en las teorías de la percepción visual que se encuentran en la literatura. Este se dividió en varios niveles jerárquicos que representan o simulan el sistema visual de los humanos, que son: Retina, NLG, Áreas V1/V2, Área MT y Área V4.

Los resultados de ambos algoritmos se evaluaron con la base de datos VSB100 ya que contiene videos con escenarios complejos y se utilizan las métricas BPR y VPR para medir la calidad objetiva espacio-temporal de la segmentación de objetos. De acuerdo al estudio comparativo realizado, NISeg obtuvo mejor desempeño que DObjA, ya que maneja características como color, bordes, movimiento, considera información de cuadros anteriores y utiliza la información local y no solo de píxeles individuales.

Por otro lado, NISeg alcanzó un 5° lugar en la comparación con otros algoritmos de la literatura evaluados en VSB100. De acuerdo a los resultados del algoritmo de Galasso et al. [99] se observó un desempeño casi igual a NISeg en ambas métricas, lo que indica que utilizar las mismas características de color, bordes y movimiento genera desempeños similares. Otros algoritmos como los de Corso et al. [98] y Grundmann et al. [102] tienen buen valor en VPR pero bajos resultados en BPR ya que es necesario tener que ajustar el escalamiento de la segmentación al cambiar de videos.

Entre los algoritmos que consiguieron mejores desempeños que NISeg están ESP y MS TW, pero tienen la desventaja de tener que ajustar uno de sus parámetros de manera manual, estos son: la magnitud de la combinación lineal entre superpíxeles para ESP y el número de valores propios para MS TW. Con respecto al algoritmo de Arbelaez et al. [101], aunque este tuvo un buen desempeño en BPR, tiene la gran desventaja en relación a NISeg de que necesita de los *ground truths* para tener un buen desempeño en la métrica VPR.

A través de las ventajas y el desempeño que tuvo NISeg respecto a otros algoritmos existentes en la literatura evaluados en VSB100, se llegó a la conclusión de que es factible migrar los conocimientos de los modelos y procesos de la percepción visual humana a algoritmos computacionales de segmentación de objetos.

Como trabajo futuro, se plantea incluir un módulo de memoria en el diagrama de NISeg que representaría el funcionamiento de la corteza prefrontal, de manera que pueda servir para ajustar los algoritmos que extraen las características de más bajo nivel, como lo son: bordes, color y movimiento. Para esto se utilizaría una red neuronal que se entrenaría aprovechando el conjunto de videos de entrenamiento de VSB100, esto con la finalidad de mejorar ambas métricas de segmentación. También para mejorar un poco más VPR podría considerarse un GBS modificado, que tome al menos la información de un píxel de un cuadro anterior.

REFERENCIAS

- [1] D. K. Hu, J. P. Li, S. X. Yang, y S. Gregori, “A bio-inspired model for color image segmentation”, *2009 Int. Conf. Apperceiving Comput. Intell. Anal. ICACIA 2009*, pp. 317–320, 2009.
- [2] I. M. Avila-Mora y C. Castellanos-Sánchez, “Bio-inspired clustering of moving objects”, *Proc. 2009 WRI Glob. Congr. Intell. Syst. GCIS 2009*, vol. 4, pp. 58–62, 2009.
- [3] E. W. Contini, S. G. Wardle, y T. A. Carlson, “Decoding the time-course of object recognition in the human brain: From visual features to categorical decisions”, *Neuropsychologia*, vol. 105, pp. 165–176, 2017.
- [4] Y. Hwang, M.-H. Jeong, W. Cho, y S.-R. Oh, “Dual-layered oscillatory model for object detection and tracking”, *Electron. Lett.*, vol. 53, núm. 16, pp. 1092–1094, 2017.
- [5] G. Sanchez, J. Madrenas, y J. Cosp-vilella, “LEGION-based image segmentation by means of spiking neural networks using normalized synaptic weights implemented on a compact scalable neuromorphic architecture”, *Neurocomputing*, vol. 352, pp. 106–120, 2019.
- [6] D. S. Levine, *Introduction to Neural and Cognitive Modeling*, 3rd ed. New York and London: Routledge, 2018.
- [7] D. Zhou, H. Zhou, C. Gao, y Y. Guo, “Simplified parameters model of PCNN and its application to image segmentation”, *Pattern Anal. Appl.*, vol. 19, núm. 4, pp. 939–951, 2015.
- [8] T. Masquelier y S. J. Thorpe, “Learning to recognize objects using waves of spikes and spike timing-dependent plasticity”, *Proc. Int. Jt. Conf. Neural Networks*, 2010.
- [9] R. J. Alberto y M. Chacon, “Artificial Neural Image Processing Applications: A Survey”, núm. August 2016, 2012.
- [10] J. G. Yu, G. S. Xia, C. Gao, y A. Samal, “A Computational Model for Object-Based Visual Saliency: Spreading Attention Along Gestalt Cues”, *IEEE Trans. Multimed.*, vol. 18, núm. 2, pp. 273–286, 2016.
- [11] K. Grill-Spector y R. Malach, “The Human Visual Cortex”, *Annu. Rev. Neurosci.*, vol.

- 27, núm. 1, pp. 649–677, 2004.
- [12] A. Crossman y D. Neary, *Neuroanatomía*, 3a ed. Bracelona: ELSEVIER MASSON, 2007.
 - [13] J. H. Martin, *Neuroanatomy text and atlas*, FOURTH EDI. McGraw-Hill Education, 2012.
 - [14] M. Mayeli, “Neurophysiology of Visual Perception”, pp. 13–26, 2019.
 - [15] X. Xu, J. M. Ichida, J. D. Allison, J. D. Boyd, A. B. Bonds, y V. A. Casagrande, “A comparison of koniocellular, magnocellular and parvocellular receptive field properties in the lateral geniculate nucleus of the owl monkey (*Aotus trivirgatus*)”, *J. Physiol.*, vol. 531, núm. 1, pp. 203–218, 2001.
 - [16] T. Poscoliero y M. Girelli, “Electrophysiological Modulation in an Effort to Complete Illusory Figures: Configuration, Illusory Contour and Closure Effects”, *Brain Topogr.*, vol. 31, núm. 2, pp. 202–217, 2018.
 - [17] N. Kruger *et al.*, “Deep hierarchies in the primate visual cortex: What can we learn for computer vision?”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, núm. 8, pp. 1847–1871, 2013.
 - [18] N. V. K. Medathati, H. Neumann, G. S. Masson, y P. Kornprobst, “Bio-inspired computer vision: Towards a synergistic approach of artificial and biological vision”, *Comput. Vis. Image Underst.*, vol. 150, pp. 1–30, 2016.
 - [19] Y. Li y Z. Gao, “An Automatic Fuzzy Clustering Segmentation Algorithm with Aid of Set Partitioning”.
 - [20] P. Wang *et al.*, “Understanding Convolution for Semantic Segmentation”, *WACV-2018*, 2018.
 - [21] Y. Guo, Y. Liu, T. Georgiou, y M. S. Lew, “A review of semantic segmentation using deep neural networks”, *Int. J. Multimed. Inf. Retr.*, vol. 7, núm. 2, pp. 1–7, 2017.
 - [22] H. Caesar, J. Uijlings, y V. Ferrari, “Region-based semantic segmentation with end-to-end training”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905 LNCS, pp. 381–397, 2016.
 - [23] Wa. Rawat y Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review”, *Neural Comput.*, vol. 48, núm. 1–2, pp.

- 181–211, 2017.
- [24] J. Long, E. Shelhamer, y T. Darrel, “Fully convolutional networks for semantic segmentation”, 2015.
 - [25] X. Fu y H. Qu, “Research on Semantic Segmentation of High-resolution Remote Sensing Image Based on Full Convolutional Neural Network”, *2018 12th Int. Symp. Antennas, Propag. EM Theory, ISAPE 2018 - Proc.*, pp. 1–4, 2019.
 - [26] A. Khoreva, R. Benenson, J. Hosang, M. Hein, y B. Schiele, “Simple does It: Weakly supervised instance and semantic segmentation”, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1665–1674, 2017.
 - [27] Y. Liu, J. Han, Q. Zhang, y L. Wang, “Salient Object Detection Via Two-Stage Graphs”, vol. 1, núm. c, pp. 1–15, 2018.
 - [28] W. Wang, J. Shen, R. Yang, y F. Porikli, “A Unified Spatiotemporal Prior based on Geodesic Distance for Video Object Segmentation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, núm. 1, pp. 20–33, 2018.
 - [29] A. Borji, “What is a salient object? A dataset and a baseline model for salient object detection”, *IEEE Trans. Image Process.*, vol. 24, núm. 2, pp. 742–756, 2015.
 - [30] S. S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, y R. V. Babu, “Saliency Unified: A Deep Architecture for simultaneous Eye Fixation Prediction and Salient Object Segmentation”, pp. 5781–5790, 2016.
 - [31] G. Singh y M. A. Ansari, “Efficient detection of brain tumor from MRIs using K-means segmentation and normalized histogram”, *India Int. Conf. Inf. Process. IICIP 2016 - Proc.*, vol. 1, núm. 2, 2017.
 - [32] X. Meng, W. Gu, Y. Chen, y J. Zhang, “Brain MR image segmentation based on Gaussian Filtering and Improved FCM”, *PLoS One*, vol. 12, núm. 8, 2017.
 - [33] A. Sorkine, F. Perazzi, y O. Wang, “Methods and systems of performing video object segmentation”, 2018.
 - [34] F. Galasso, N. S. Nagaraja, T. Jim, T. Brox, y B. Schiele, “A Unified Video Segmentation Benchmark : Annotation, Metrics and Analysis”, *2013 IEEE Int. Conf. Comput. Vis.*, 2013.
 - [35] I. Essa, V. Kwatra, y M. Grundmann, “Vector representation for video segmentation”,

- 2018.
- [36] A. Khoreva, F. Galasso, y M. Hein, “Classifier Based Graph Construction for Video Segmentation”, *2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.
 - [37] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, y M. Felsberg, “A Generative Appearance Model for End-to-end Video Object Segmentation”, 2018.
 - [38] A. Ng, M. Jordan, y Y. Weiss, “On Spectral Clustering: Analysis and an algorithm”, *Nips*, 2001.
 - [39] Z. Li y J. Chen, “Superpixel segmentation using Linear Spectral Clustering”, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1356–1363, 2015.
 - [40] A. González-Casillas *et al.*, “Towards a model of visual recognition based on neurosciences”, *Procedia Comput. Sci.*, vol. 145, pp. 214-231. ISSN 1877-0509, 2018.
 - [41] Z. Gao, J. Zeng, y H. Liu, “A biologically-inspired model for dynamic saliency detection”, *Process. 2014 Int. Conf. Multisens. Fusion Inf. Integr. Intell. Syst. MFI 2014*, 2014.
 - [42] J. Vacher, P. Mamassian, y R. Coen-Cagli, “Probabilistic Model of Visual Segmentation”, pp. 1–13, 2018.
 - [43] Z. Yang, J. Lian, S. Li, Y. Guo, Y. Qi, y Y. Ma, “Heterogeneous SPCNN and its application in image segmentation”, *Neurocomputing*, vol. 285, pp. 196–203, 2018.
 - [44] F. Lopes, “EEG: Origin and Measurement”, pp. 19–39, 2010.
 - [45] H. Hogendoorn y A. N. Burkitt, “Predictive coding of visual object position ahead of moving objects revealed by time-resolved EEG decoding”, *Neuroimage*, vol. 171, pp. 55–61, 2018.
 - [46] D. Rosenthal, P. Deguzman, L. C. Parra, y P. Sajda, “Evoked neural responses to events in video”, *IEEE J. Sel. Top. Signal Process.*, vol. 8, núm. 3, pp. 358–365, 2014.
 - [47] C. L. Larson, J. Aronoff, I. C. Sarinopoulos, y D. C. Zhu, “Recognizing Threat: A Simple Geometric Shape Activates Neural Circuitry for Threat Detection”, *J. Cogn. Neurosci.*, vol. 21, núm. 8, pp. 1523–1535, 2009.
 - [48] J. Bulthé, B. De Smedt, y H. P. Op de Beeck, “Decoding Dynamic Brain Patterns from Evoked Responses: A Tutorial on Multivariate Pattern Analysis Applied to Time Series

- Neuroimaging Data”, *J. Cogn. Neurosci.*, vol. 29, pp. 677–697, 2017.
- [49] F. Li, C. Yi, Y. Jiang, Y. Liao, Y. Si, y J. Dai, “Different Contexts in the Oddball Paradigm Induce Distinct Brain Networks in Generating the P300”, vol. 12, pp. 1–10, 2019.
- [50] J. Jin, H. Zhang, I. Daly, X. Wang, y A. Cichocki, “An improved P300 pattern in BCI to catch user’s attention”, *J. Neural Eng.*, vol. 14, 2017.
- [51] S. Vlad y N. M. Roman, *International Conference on Advancements of Medicine and Health Care through Technology*, vol. 71, núm. October. 2018.
- [52] J. V. Odom, M. Bach, M. Brigell, G. E. Holder, D. L. McCulloch, y A. P. Tormene, “ISCEV standard for clinical visual evoked potentials (2016 update)”, *Doc. Ophthalmol.*, vol. 120, núm. 1, pp. 111–119, 2016.
- [53] A. D. Corral-Sáenz, R. Rangel-González, y M. I. Chacón-Murguía, “Análisis del Desempeño de Sistema de Detección de Señal SSVEP Utilizando Clasificadores Árbol Simple y Máquina de Vectores de Soporte”, núm. 1, pp. 1–23, 2019.
- [54] C. Im y J. M. Seo, “A review of electrodes for the electrical brain signal recording”, *Biomed. Eng. Lett.*, vol. 6, núm. 3, pp. 104–112, 2016.
- [55] OpenBCI, “ULTRACORTEX MARK IV”, 2018. [En línea]. Disponible en: <https://docs.openbci.com/Headware/01-Ultracortex-Mark-IV>.
- [56] H. Marzbani, H. R. Marateb, y M. Mansourian, “A comprehensive review on system design, methodology and clinical applications”, *Basic Clin. Neurosci.*, vol. 7, pp. 143–158, 2016.
- [57] J. M. Macías-Macías, J. A. Ramírez-Quintana, A. D. Corral-Saénz, y M. I. Chacón-Murguía, “Detección de la onda p300 con red neuronal convolucional implementada en sistema embebido portable”, *40° Congr. Int. Ing. EN Electron. ELECTRO 2018*, vol. 40, pp. 175–180, 2018.
- [58] P. Roy y P. Kumar, “A parallel LEGION algorithm and cell-based architecture for real time split and merge video segmentation”, *J. Real-Time Image Process.*, 2015.
- [59] L. Zhao, F. A. Breve, M. G. Quiles, R. A. F. Romero, C. Postal, y C. Sp, “Visual Selection and Shifting Mechanisms Based on a Network of Chaotic Wilson-Cowan Oscillators”, *Int. Conf. Nat. Comput.*, núm. Icnc, 2007.

- [60] S. Saraswathi y A. Allirani, “Survey on image segmentation via clustering”, *Inf. Commun. ...*, 2013.
- [61] D. Xu y Y. Tian, “A Comprehensive Survey of Clustering Algorithms”, *Ann. Data Sci.*, vol. 2, núm. 2, pp. 165–193, 2015.
- [62] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1a ed. Springer, 2006.
- [63] J. L. Diez, “Tecnicas de Agrupamiento para el Analisis de Datos Cuantitativos y Cualitativos”, núm. September, 2005.
- [64] O. Jensen, M. Bonnefond, T. R. Marshall, y P. Tiesinga, “Oscillatory mechanisms of feedforward and feedback visual processing”, *Trends Neurosci.*, vol. 38, núm. 4, pp. 192–194, 2015.
- [65] J. . Bowmaker y H. J. Dartnall, “Visual pigments of rods and cones in a human retina”, pp. 501–511, 1980.
- [66] “Functional characteristics and diversity of cat retinal ganglion cells. Basic characteristics and quantitative description”, *Investig. Ophthalmol. Vis. Sci.*, vol. 25, núm. 3, pp. 250–267, 1984.
- [67] S. N. Blythe y J. H. Krantz, “A Mathematical Model of Retinal Receptive Fields Capable of Form & Color Analysis”, vol. 1, núm. June, 2004.
- [68] B. J. Lindbloom, “RGB/XYZ Matrices”, 2017. [En línea]. Disponible en: http://www.brucelindbloom.com/index.html?Eqn_RGB_XYZ_Matrix.html. [Consultado: 03-may-2019].
- [69] MathWorks, “rgb2lab”, 2019. .
- [70] J. A. Bednar y S. P. Wilson, “Cortical Maps”, *Neurosci.*, 2015.
- [71] J. B. De Paula, R. Miikkulainen, J. Bednar, B. Kuipers, B. Porter, y E. Seidemann, “Modeling the self-organization of Color-selective neurons in the visual cortex”, núm. August, pp. 1–29, 2007.
- [72] P. F. Felzenszwalb y D. P. Huttenlocher, “Efficient graph-based image segmentation”, *Int. J. Comput. Vis.*, vol. 59, núm. 2, pp. 167–181, 2004.
- [73] Y. Ben-Shabat, T. Avraham, M. Lindenbaum, y A. Fischer, “Graph based over-segmentation methods for 3D point clouds”, *Comput. Vis. Image Underst.*, vol. 174, núm. March, pp. 12–23, 2018.

- [74] V. Murino y E. Puppo, “Fast Superpixel-Based Hierarchical Approach to Image Segmentation”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9279, pp. 364–374, 2015.
- [75] A. Saglam y N. A. Baykan, “Effects of color spaces and distance norms on graph-based image segmentation”, *2017 3rd Int. Conf. Front. Signal Process. ICFSP 2017*, pp. 130–135, 2017.
- [76] A. Palmer, “An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex”, vol. 58, núm. 6, 1987.
- [77] S. L. Liu, Z. D. Niu, G. Sun, y Z. P. Chen, “Gabor filter-based edge detection: A note”, *Optik (Stuttg.)*, vol. 125, núm. 15, pp. 4120–4123, 2014.
- [78] Y. Cao y S. Grossberg, *A Laminar Cortical Model of Stereopsis and 3D Surface Perception : Closure and da Vinci Stereopsis*. 2004.
- [79] L. Chua y T. Roska, *Cellular neural networks and visual computing*. Cambridge, United Kingdom: CAMBRIDGE UNIVERSITY PRESS.
- [80] R. A. Andersen, “Neural mechanisms of visual motion perception in primates”, *Neuron*, vol. 18, núm. 6, pp. 865–872, 1997.
- [81] D. Fortun, P. Bouthemy, y C. Kervrann, “Optical flow modeling and computation: A survey”, *Comput. Vis. Image Underst.*, vol. 134, pp. 1–21, 2015.
- [82] W. H. Warren, “The state of flow”, *High-level motion Process.*, pp. 315–358, 1998.
- [83] A. Plyer, G. Le Besnerais, y F. Champagnat, “Massively parallel Lucas Kanade optical flow for real-time video processing applications”, *J. Real-Time Image Process.*, vol. 11, núm. 4, pp. 713–730, 2014.
- [84] J. L. Barron, D. J. Fleet, y S. S. Beauchemin, “Performance of Optical Flow Techniques”, *Philos. Behind Phys.*, vol. 12, pp. 43–77, 1994.
- [85] Mathworks, “opticalFlowLK”, 2019. [En línea]. Disponible en: <https://la.mathworks.com/help/vision/ref/opticalflowlk.html>.
- [86] L. R. Squire, *Encyclopedia of Neuroscience*. Academic Press, 2009.
- [87] H. Mobahi, R. Collobert, y J. Weston, “Deep learning from temporal coherence in video”, pp. 1–8, 2009.
- [88] T. Kim, W. Bair, y A. Pasupathy, “Neural coding for shape and texture in macaque

- area V4”, *J. Neurosci.*, pp. 3073–18, 2019.
- [89] A. Pasupathy, “The neural basis of image segmentation in the primate brain”, *Neuroscience*, vol. 296, núm. October, pp. 101–109, 2015.
 - [90] Y. El-Shamayleh y A. Pasupathy, “Contour Curvature As an Invariant Code for Objects in Visual Area V4”, *J. Neurosci.*, vol. 36, núm. 20, pp. 5532–5543, 2016.
 - [91] G. Okazawa, S. Tajima, y H. Komatsu, “Image statistics underlying natural texture selectivity of neurons in macaque V4”, *Proc. Natl. Acad. Sci.*, vol. 112, núm. 4, pp. E351–E360, 2015.
 - [92] P. Sundberg, T. Brox, M. Maire, P. Arbel, y J. Malik, “Occlusion Boundary Detection and Figure / Ground Assignment from Optical Flow”.
 - [93] F. Galasso, N. S. Nagaraja, T. J. Cardenas, T. Brox, y B. Schiele, “A unified video segmentation benchmark: Annotation, metrics and analysis”, *Proc. IEEE Int. Conf. Comput. Vis.*, núm. April, pp. 3527–3534, 2013.
 - [94] M. Keuper y T. Brox, “Point-Wise Mutual Information-Based Video Segmentation with High Temporal Consistency”, pp. 789–803, 2016.
 - [95] H. Xu, E. R. Hancock, y W. Zhou, “The low-rank decomposition of correlation-enhanced superpixels for video segmentation”, *Soft Comput.*, vol. 23, núm. 24, pp. 13055–13065, 2019.
 - [96] Y. Zhao, Y. Yuan, F. Nie, y Q. Wang, “Spectral clustering based on iterative optimization for large-scale and high-dimensional data”, *Neurocomputing*, vol. 318, pp. 227–235, 2018.
 - [97] P. Ochs y T. Brox, “Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions”, *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1583–1590, 2011.
 - [98] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, y A. Yuille, “Efficient multilevel brain tumor segmentation with integrated bayesian model classification”, *IEEE Trans. Med. Imaging*, vol. 27, pp. 629–640, 2008.
 - [99] F. Galasso, R. Cipolla, y B. Schiele, “Video Segmentation with Superpixels”, *Comput. Vis. -- ACCV 2012*, vol. 7724, pp. 760–774, 2013.
 - [100] C. Xu, C. Xiong, y J. J. Corso, “Streaming hierarchical video segmentation”, *Lect.*

- Notes Comput. Sci.*, pp. 626–639, 2012.
- [101] P. Arbel, M. Maire, C. Fowlkes, y J. Malik, “Contour detection and hierarchical image segmentation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 898–916, 2011.
- [102] M. Grundmann, M. Han, y I. Essa, “Efficient Hierarchical Graph-Based Video Segmentation”, *2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010.
- [103] D. J. Bora, “Performance Comparison of K-Means Algorithm and FCM Algorithm with Respect to Color Image Segmentation”, *Int. J. Emerg. Technol. Adv. Eng.*, vol. 7, núm. 8, pp. 460–470, 2017.