



Escuela
Politécnica
Superior

Inferencia automática de modelos bioinspirados de visión para neuroprótesis visuales



Máster Universitario en Ingeniería de
Telecomunicación

Trabajo Fin de Máster

Autor:

Rubén Crespo Cano

Tutor/es:

Antonio Martínez Álvarez

Sergio Cuenca Asensi

Eduardo Fernández Jover

Noviembre 2015



Universitat d'Alacant
Universidad de Alicante

UNIVERSIDAD DE ALICANTE

ESCUELA POLITÉCNICA SUPERIOR

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN



Universitat d'Alacant
Universidad de Alicante

TRABAJO FIN DE MÁSTER:

*I*NFERENCIA *A*UTOMÁTICA DE *M*ODELOS
*B*IOINSPIRADOS DE *V*ISIÓN PARA
*N*EUROPRÓTESIS *V*ISUALES

Autor:

RUBÉN CRESPO CANO

Tutores:

DR. ANTONIO MARTÍNEZ ÁLVAREZ

DR. SERGIO CUENCA ASENSI

DR. EDUARDO FERNÁNDEZ JOVER

NOVIEMBRE DE 2015

Agradecimientos

La realización del presente Trabajo Final de Máster se fundamenta en el trabajo realizado por el Dr. Antonio Martínez Álvarez, quien me ha guiado y dirigido durante el último año, aportando ideas y sugerencias, pero siempre confiando en mi criterio. Además, debo agradecerle la disponibilidad que me ha proporcionado, estando operativo los siete días de la semana. Igualmente quiero agradecerle que me haya convencido para realizar con él mi tesis doctoral.

Quiero transmitir mis agradecimientos a mis otros dos directores, los doctores Eduardo Fernández Jover y Sergio Cuenca Asensi, por todo el conocimiento que me han aportado. Asimismo, quiero dar las gracias a la investigadora y estudiante de doctorado Ariadna Díaz Tahoces por todo el trabajo que ha realizado, proporcionándome los registros biológicos y trasladándome el conocimiento biológico necesario, ya que sin su esfuerzo no habría sido posible desarrollar este trabajo.

No me quiero olvidar de mi padre y de mi madre, ya que han sido los que me han dado todo, y sin los cuales no podría haber realizado mis estudios.

Y por último, quiero agradecer a Ana que haya sido capaz de aguantarme todo este tiempo, apoyándome siempre en todo momento.

Dedicatoria

Dedicado a mi familia y a Ana.

“El propósito de la educación es mostrar a la gente cómo aprender por sí mismos. El otro concepto de la educación es adoctrinamiento.”

Noam Chomsky.

“Si quieres cambiar al mundo, cámbiate a ti mismo”

Mahatma Gandhi.

Índice general

1. Introducción	11
1.1. Motivación	11
1.2. Objetivos	12
1.3. Metodología	12
1.4. Contenido y estructura	12
1.5. Publicaciones	13
2. Estado del arte	15
2.1. Histología y funcionamiento de la retina	15
2.2. Modelos de retina	19
2.2.1. Modelos de bajo nivel	19
2.2.2. Modelos de alto nivel	21
2.3. Descripción del modelo de retina diseñado	22
2.3.1. Modelo neuronal de integración y disparo	24
3. Estrategia evolutiva multiobjetivo de inferencia de modelos de retina	27
3.1. Introducción	27
3.2. Algoritmos Genéticos	27
3.2.1. Elementos de un Algoritmo Genético	28
3.2.2. Algoritmo NSGA-II	29
3.2.3. Codificación de los cromosomas	30
4. Herramientas y métodos	33
4.1. Diseño del software de modelado y simulación	33
4.2. Tecnologías	36
4.2.1. Python	36
4.2.2. SciPy y NumPy	36
4.2.3. Biblioteca de primitivas de procesamiento de imagen	36
4.2.4. Machine Learning in Python: scikit-learn	36
4.2.5. Distributed Evolutionary Algorithms in Python	37
4.2.6. Vision Egg	37
4.2.7. Clúster de computación	37
5. Casos de estudio	39
5.1. Setup experimental	39
5.2. Registros electrofisiológicos	43
5.3. Campos receptivos: delimitación, localización y cálculo del área	47
5.3.1. Delimitación y localización	47
5.3.2. Cálculo del área del campo receptivo	48
5.4. Modelado de las células ganglionares <i>transient</i>	50
5.5. Métricas	50

5.5.1. Firing rate absolute difference (FRAD)	51
5.5.2. Peri-Stimulus Time Histogram (PSTH) - Kullback-Leibler divergence (KLD)	51
5.5.3. Interspike Interval Histogram (ISI) - Kullback-Leibler divergence (KLD)	52
5.5.4. Receptive field absolute difference (RFAD)	53
5.6. Caso de estudio 1: Estudio de la convergencia del método	53
5.7. Caso de estudio 2: Ajuste paramétrico multiobjetivo de un modelo de retina	57
6. Conclusión	61
7. Anexos	67
7.1. Anexo 1: Clase MoogaRetina.py	67

Índice de figuras

2.1. Sección de un ojo humano.	16
2.2. Interconexión de las células retinianas.	17
2.3. Función <i>sombrero mexicano</i> o <i>mexican hat</i>	18
2.4. Campos receptivos de los distintos tipos de células ganglionares.	19
2.5. Modelo de Retina propuesto por Misha Mahowald y Carver Mead.	21
2.6. Bloques funcionales del procedimiento del modelo de retina bio- inspirado bajo estudio.	23
2.7. Etapas del proceso de filtrado y modelado hasta la obtención de la salida del modelo neuronal de integración y disparo.	24
2.8. Ejemplo de salida del modelo neuronal de integración y disparo de una matriz de actividad 10×10	25
2.9. Descripción del modelo neuronal de integración y disparo.	25
2.10. Codificación frecuencial de la matriz de actividad.	26
3.1. Pasos de la estrategia de modelado multiobjetivo.	28
3.2. Proceso de selección del algoritmo NSGA-II.	30
3.3. Ejemplo de codificación de un individuo.	31
4.1. Estructura del patrón de diseño <i>Factory Method</i>	34
4.2. Diagrama de clases UML.	35
5.1. Estímulo utilizado para la clasificación de las células ganglionares.	40
5.2. Representación del estímulo utilizado para la clasificación de las células ganglionares.	40
5.3. Representación de los estímulos con barras blancas utilizados en los experimentos. Las flechas de color gris indican la dirección del movimiento de las barras, desplazándose en ambos sentidos.	41
5.4. Descripción de la generación y utilización de los estímulos vi- suales creados con <i>Vision Egg</i> así como el diagrama de flujo del procesamiento de los registros biológicos y sintéticos.	45
5.5. Procedimiento de adquisición de datos.	46
5.6. Puesta en fase de las respuestas $0^\circ - 180^\circ$, $45^\circ - 225^\circ$, $90^\circ - 270^\circ$ y $135^\circ - 315^\circ$	47
5.7. Muestra del cálculo del centroide de una célula ganglionar y con- junto de campos receptivos de distintas células ganglionares.	48
5.8. Modelo GMM de dos componentes.	49
5.9. Modelado de la respuesta impulsiva de una célula ganglionar <i>tran-</i> <i>sient</i>	50
5.10. Definición de la tasa de disparo media dada una ventana temporal.	51
5.11. Peri-Stimulus-Time Histogram (PSTH) tras varias ejecuciones del experimento.	52

5.12. A. Tren de impulsos con intervalos variables. B. Distribución del ISI.	53
5.13. Test de convergencia para 300 generaciones de las métricas PSTH-KLD (a), FRAD (b), ISI-KLD (c) y RFAD (d) en el proceso de comparación de los registros sintéticos y biológicos.	55
5.14. Test de convergencia para 2500 generaciones de las métricas PSTH-KLD (a), FRAD (b), ISI-KLD (c) y RFAD (d) en el proceso de comparación de los registros sintéticos y biológicos.	56
5.15. Gráficos multiobjetivo que enfrentan a cada par de métricas (300 generaciones).	58
5.16. Gráficos multiobjetivo que enfrentan a cada par de métricas (2500 generaciones).	59

Índice de tablas

5.1. Parámetros del Cromosoma	43
5.2. Etiquetado de las pruebas de convergencia	53

Capítulo 1

Introducción

El presente trabajo, titulado “*Inferencia automática de modelos bioinspirados de visión para neuroprótesis visuales*”, aborda la problemática del modelado a alto nivel de una retina biológica, y se enfoca tanto en la inferencia automática de un modelo de retina que integra distintas contribuciones funcionales, como en su optimización automática mediante un algoritmo genético y multiobjetivo que aproxima el comportamiento sintético y biológico. El trabajo se valida con un detallado caso de estudio que presenta distintas métricas para medir la bondad del ajuste paramétrico llevado a cabo en la etapa de experimentación.

El proyecto ha sido realizado gracias a la colaboración existente entre el departamento de *Tecnología Informática y Computación* de la *Universidad de Alicante* y el *Instituto de Bioingeniería* de la *Universidad Miguel Hernández*, donde se han llevado a cabo los experimentos de laboratorio necesarios para la obtención de los registros biológicos, imprescindibles para poder establecer métricas comparativas entre los modelos biológicos y sintéticos.

1.1. Motivación

El avance en los últimos años en áreas de conocimiento tales como la ingeniería de software, ingeniería electrónica, o la neurociencia, ha dado lugar a nuevas disciplinas de carácter interdisciplinar, tales como la *neuroingeniería*, con la finalidad del estudio y resolución de problemas mediante sistemas artificiales. La *neuroingeniería* aborda el estudio del sistema nervioso biológico y el desarrollo de interfaces de interacción y acceso.

El objetivo final del estudio sería el diseño de un chip capaz de emular el comportamiento de la retina humana, consiguiendo un sistema altamente configurable y parametrizable, que pudiera adaptarse adecuadamente a cada individuo. Los sistemas basados en software pueden ser una buena solución, atendiendo a la relación rendimiento/coste, pero penalizarían otros requisitos, como son la velocidad, consumo o rendimiento. Para poder llevar a cabo los distintos objetivos es necesario conocer cómo funcionan los distintos modelos de retina, con el fin de realizar una configuración paramétrica óptima, atendiendo a las necesidades de cada individuo, consiguiendo así hacer las veces de una retina biológica mediante una prótesis electrónica.

En resumen, uno de los principales objetivos del presente Trabajo Final de Máster consiste en integrar distintos conocimientos, propios de la *neuroingeniería*, la ingeniería de software, y el tratamiento digital de señales e imágenes, para modelar una retina cuyo comportamiento deberá ser sintetizado en hardware de forma conveniente.

1.2. Objetivos

A continuación se enumeran los objetivos de este trabajo:

- Realizar un estado del arte sobre los modelos de retina junto con su problemática.
- Realizar una introducción a las técnicas de computación evolutiva.
- Proponer una métrica que permita medir la bondad de un modelo sintético de retina integrado en una neuroprótesis visual.
- Diseñar una infraestructura de simulación de modelos de visión bioinspirada altamente parametrizable y orientada a la aproximación de modelos de simulación.
- Desarrollar un sistema automático de inferencia de modelos bioinspirados de retina en función de múltiples objetivos de interés.
- Estudiar y adaptar el algoritmo NSGA-II para su integración en la infraestructura de simulación.
- Desarrollar diversos casos de estudio que muestren la viabilidad del sistema propuesto.

1.3. Metodología

El trabajo está estructurado en dos fases diferenciadas: la primera fase consiste en el estudio y revisión de los modelos de retina actuales, junto con el estudio de los Algoritmos Genéticos en general y el algoritmo NSGA-II en particular. La segunda fase consiste en el estudio de las distintas métricas de calidad y el desarrollo de un sistema automático de inferencia de modelos de retina, junto con los casos de estudio propuestos para comprobar experimentalmente la concordancia entre el sistema propuesto y los registros biológicos obtenidos en el laboratorio.

1.4. Contenido y estructura

El presente trabajo se estructura en siete capítulos. El contenido de cada uno de ellos se describe a continuación:

1. **Introducción.** En este capítulo se presenta la motivación, los objetivos, la metodología, el contenido y la estructura del presente Trabajo Final de Máster. Además, se enumeran los distintos textos de carácter científico que han sido publicados a raíz del trabajo realizado.
2. **Estado del arte.** En este capítulo se presenta una introducción teórica al modelo de retina bajo estudio. También se da una perspectiva sobre qué es una retina humana y los esfuerzos actuales en imitar el comportamiento de ésta, mediante neuroprótesis destinadas a mejorar la visión en pacientes en los que se ha perdido o visto afectada por problemas retinianos.
3. **Estrategia evolutiva multiobjetivo de inferencia de modelos de retina.** En este capítulo se explica la estrategia evolutiva llevada a cabo para reducir el espacio de búsqueda en el proceso de ajuste paramétrico de los modelos de retina.

4. **Herramientas y métodos.** Capítulo en el que se presenta el diseño técnico del proyecto realizado, así como las herramientas que se han utilizado.
5. **Casos de estudio.** En este capítulo se presentan dos casos de estudio, realizados para cerciorar el comportamiento del sistema diseñado y generar conocimiento útil en el avance del proceso de estudio de la retina. También se presenta el modo en el que han sido adquiridos los datos biológicos necesarios para poder realizar el estudio comparativo entre las retinas biológicas y sintéticas, así como la forma en la que se han modelado las distintas métricas comparativas.
6. **Conclusiones.** El propósito de este apartado es el de sintetizar las principales conclusiones derivadas de este trabajo.
7. **Anexos.** Por último, para finalizar el trabajo, se adjuntan los diversos anexos necesarios para comprender globalmente el Trabajo Final de Máster, como pueden ser partes del código fuente desarrollado.

1.5. Publicaciones

Las publicaciones relacionadas con este trabajo consisten en una comunicación en el congreso internacional IWINAC [1] y una publicación científica. Destacar que el trabajo presentado en el congreso internacional fue seleccionado como uno de los mejores trabajos. En cuanto al artículo científico, se recibió una invitación para su publicación. Actualmente, dicha publicación se encuentra en el primer ciclo de revisión en un *journal* Q1 con factor de impacto en 2014 de 6,507 e indexado en el *Journal of Citation Reports* (JCR).

- Rubén Crespo-Cano, Antonio Martínez-Álvarez, Ariadna Díaz-Tahoces, Sergio Cuenca-Asensi, J. M. Ferrández, Eduardo Fernández, “On the Automatic Tuning of a Retina Model by Using a Multi-objective Optimization Genetic Algorithm,” *Artificial Computation in Biology and Medicine*, Volume 9107 of the series Lecture Notes in Computer Science pp 108-118, 2015 [2].
- Antonio Martínez-Álvarez, Rubén Crespo-Cano, Ariadna Díaz-Tahoces, Sergio Cuenca-Asensi, J. M. Ferrández, Eduardo Fernández, “Automatic Tuning of a Retina Model by Using a Multi-objective Optimization Genetic Algorithm,” *International Journal of Neural Systems*, [en revisión].

Capítulo 2

Estado del arte

El objetivo de este capítulo es dar una perspectiva del funcionamiento y estructura de la retina humana y explorar los principales modelos de retina publicados por la comunidad científica, los cuales han constituido la base del modelo final diseñado.

2.1. Histología y funcionamiento de la retina

En la parte posterior del globo ocular (de vertebrados y cefalópodos) se encuentra una delgada capa de células, denominada retina. El sistema visual tiene por función la transformación de los estímulos luminosos, que llegan en forma de ondas electromagnéticas en el espectro visible, en señales eléctricas que son interpretadas por el cerebro. La retina forma parte del sistema visual y se considera parte del sistema nervioso, concretamente del cerebro. La retina integra un conjunto rico de células especializadas y complejas estructuras neuronales, las cuales son sensibles al color, intensidad lumínica, movimiento, detección de bordes, y otras muchas características que componen el sentido de la vista. Su área en la especie humana es de unos $42mm$.

La luz se proyecta en la retina tras atravesar todo el globo ocular, entrando por una diminuta apertura de sólo unos milímetros de diámetro, la pupila. La información que se percibe se empieza a procesar en la retina, gracias a un conjunto de células especializadas en percibir el contraste, el color, el movimiento y otras cualidades dependientes de las especies. Es de importancia conocer que los sistemas visuales de los distintos seres vivos, se pueden diferenciar entre sí de forma muy notable, tanto por las propiedades fisiológicas como por la cantidad de células sensoriales presentes.

Mediante el uso de esas estructuras y circuitos neuronales, la retina realiza un procesamiento cromático y acromático (espacio-temporal) de la información visual; finalmente codifica esta información en trenes de impulsos que son entregados, vía el nervio óptico, a los centros de procesamiento visual. Este proceso debe ser inequívoco y rápido, a fin de garantizar el reconocimiento de objetos ante cada estímulo en unos pocos cientos de milisegundos. Por tanto, la cuestión de cómo es comprimida en la retina la información sobre el mundo exterior, y cómo es codificada en trenes de impulsos, es una de las cuestiones básicas en neurociencia computacional [3]. Este conocimiento es también de importancia fundamental para el desarrollo de prótesis corticales útiles, que sean capaces de proporcionar a personas profundamente ciegas percepción visual, a través de la estimulación directa de la corteza visual [4, 5].

La retina está formada por una fina capa de tejido nervioso situada en la

parte posterior del globo ocular. La figura 2.1 muestra la sección de un ojo humano, en la que están indicadas las distintas partes que componen dicho órgano. La luz atraviesa el ojo a través de la córnea y se desplaza por el *humor acuoso*, el *crystalino*, y el *humor vítreo*. Dichas secciones tienen índices de refracción similares, configurando un camino óptico que finaliza en la fovea (lugar donde se experimenta la mayor agudeza visual), y no en el eje óptico del ojo.

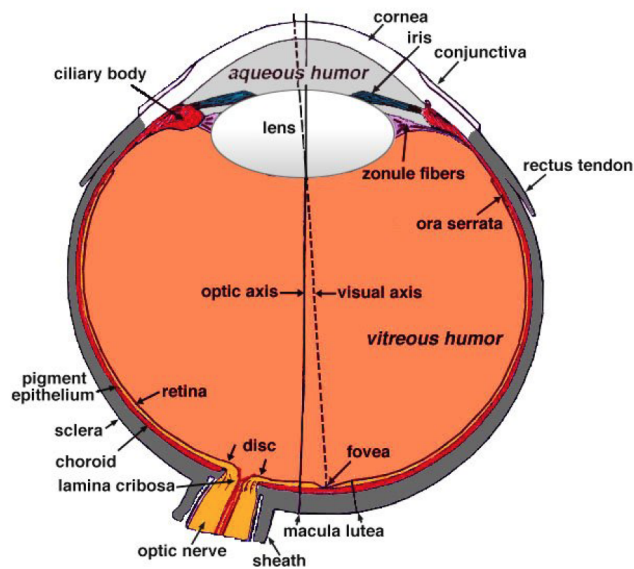


Figura 2.1: Sección de un ojo humano.

Las distintas capas que forman la retina están formadas por un grupo de cuerpos ordenados de células nerviosas. Entre cada una de ellas, se realiza la conexión sináptica –o transmisión eléctrica– que es la responsable del intercambio de información entre las células. La retina está formada por tres capas y por tanto, existen dos interfaces de interconexión. Estas dos capas de interconexión se denominan regiones plexiformes. La primera es la capa plexiforme externa, donde se produce la sinapsis entre los fotorreceptores y las dendritas de las células bipolares y horizontales. La segunda es la capa plexiforme interior, donde se conectan bipolares y amacrinas con la capa de células ganglionares. En la figura 2.2 se presenta un esquema con las células que se encuentran en la retina humana.

2.1. Histología y funcionamiento de la retina

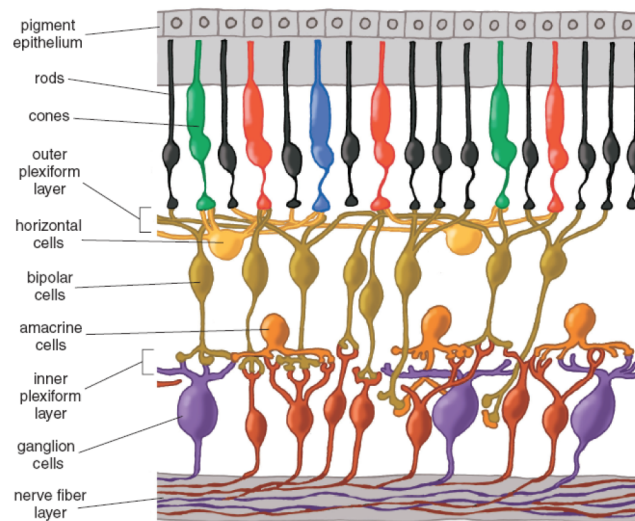
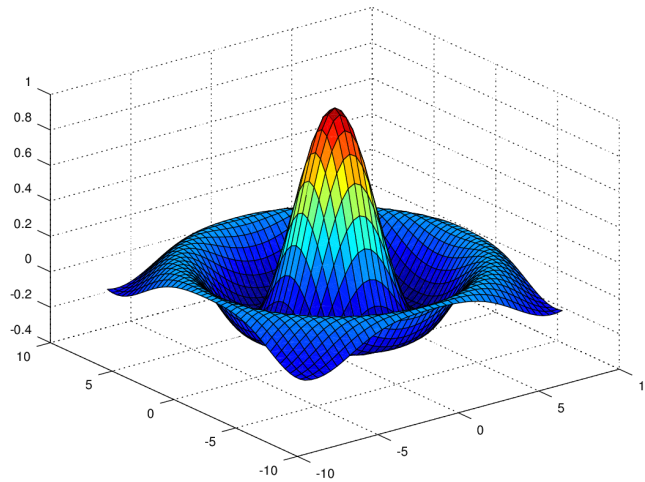


Figura 2.2: Interconexión de las células retinianas.

Dentro de la tipología de células que componen la retina se encuentran los *fotorreceptores*, encargados de transformar la luz en impulsos eléctricos. A su vez éstos se dividen en dos tipos: *conos* y *bastones*. Los bastones tienen forma alargada mientras que los conos tienen una estructura cónica. El ser humano tiene un único tipo de bastones y de dos a tres tipos de conos. Los conos son los encargados de trabajar con altas intensidades de luz y gestionar la distinción del color. Los bastones son los encargados de trabajar cuando la intensidad de luz es tenue. Los conos se sitúan en el área central de la retina en forma hexagonal y los bastones se sitúan en el perímetro.

La primera integración de la señal lumínica recibida se realiza en la *capa plexiforme externa*, donde se realiza un proceso de extracción de características y compresión de la información. Aquí es donde se realiza la comunicación eléctrica con las células horizontales y bipolares. Las células *bipolares* conectan los fotorreceptores con las células ganglionares. El ser humano tiene once tipos de células bipolares, de las cuales una trabaja con los bastones y las otras diez con los conos. Estas células a su vez se dividen en dos tipos: células despolarizantes y células polarizantes, permitiendo de esta forma transmitir señales positivas y negativas. Las células *horizontales* conectan las células bipolares entre sí, poniendo en contacto de forma lateral las neuronas de la capa plexiforme externa. Los detalles de los estímulos visuales se perciben gracias a las células horizontales. El campo receptivo es mayor que el de las células bipolares, por tanto, las células horizontales tienen un área de respuesta mayor. La respuesta de las bipolares es de tipo *ON* u *OFF*, mientras que la respuesta de las horizontales es dependiente de una señal oponente en la que intervienen los conceptos de centro y periferia de su campo receptivo. Por tanto, se forma una estructura en conjunto entre las células bipolares y horizontales en la que la información *ON* u *OFF* de las células bipolares se encuentra en el centro, y la señal oponente *OFF* u *ON* de las células horizontales se encuentra en la periferia. El filtrado que se realiza en esta estructura es similar a la función denominada *sombrero mexicano* o *mexican hat*, la cual se obtiene mediante la normalización de la segunda derivada de la función gaussiana. En la figura 2.3 se muestra un gráfico en el que se representa esta función.

La *capa plexiforme interna* es el lugar en el que se realizan las conexiones

Figura 2.3: Función *sombrero mexicano* o *mexican hat*.

sinápticas entre las células ganglionares y bipolares junto con las amacrinas. Es aquí donde se produce la segunda sinapsis de la vía vertical de la retina, entre los axones de las células bipolares y las dendritas de las células ganglionares (consultar la figura 2.2).

Las células *ganglionares* constituyen la última etapa del procesamiento que se realiza en la retina, siendo las encargadas de enviar la información codificada en forma de impulsos eléctricos a través del nervio óptico hacia el sistema nervioso central. Las células ganglionares, atendiendo al tipo de respuesta ante contrastes estacionarios introducidos en sus campos receptivos, pueden ser clasificadas en dos tipos [6]. Las células de tipo *sustained* responden con una descarga sostenida ante un contraste estacionario, mientras que las células de tipo *transient* responden decrementando la descarga ante el contraste permanente. En la figura 2.4 [7] se muestra la respuesta de la forma de onda resultante ante un contraste estacionario para dos tipos de células ganglionares: las células de tipo *Y* son células *transient* mientras que las células de tipo *X* son células *sustained*.

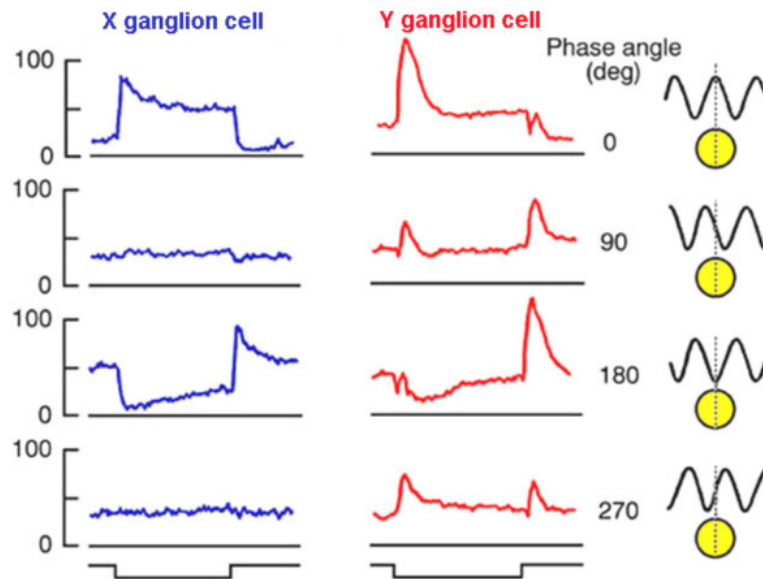


Figura 2.4: Campos receptivos de los distintos tipos de células ganglionares.

2.2. Modelos de retina

La retina es el punto de entrada al cerebro para la información sensorial visual. La información, en este caso fotones, llega a los fotorreceptores y se traduce en un mensaje eléctrico, que posteriormente se transforma en patrones de disparos que se transmiten a través del nervio óptico a varias regiones del cerebro. Tradicionalmente se ha recurrido a copiar el funcionamiento y la arquitectura biológica de una retina (y también de otros complementos del sistema visual) mediante estructuras electrónicas. Una cantidad considerable de investigadores ha estado trabajando en la generalización de las características comunes del procesamiento realizado por la retina y de los distintos modelos de retina biofísicos.

Los modelos de retina se clasifican en dos grandes grupos: los modelos de bajo nivel y los de alto nivel. Los modelos de bajo nivel (low level) describen con detalle no sólo el comportamiento de cada tipo de célula, sino su interconexión con otras células del mismo o distinto tipo, modelando así una respuesta. Los modelos de alto nivel (high level) describen de forma funcional el proceso realizado por la retina observado macroscópicamente (matemáticamente, mediante circuitos electrónicos, etc.).

A continuación se presentan, de forma categorizada, varios de los modelos más representativos publicados por la comunidad científica.

2.2.1. Modelos de bajo nivel

Los sistemas de simulación neuronal, se encargan de explotar y modelar las propiedades comunes de las neuronas para crear entornos de simulación que sean capaces de facilitar el modelado neuronal, así como proporcionar mecanismos de extensibilidad e interoperabilidad a los investigadores.

A continuación se detallan los proyectos más representativos del ámbito de la simulación neuronal aplicada al modelado de retinas.

NEST

NEST (The Neural Simulation Tool) es una herramienta para el diseño de modelos neuronales de impulsos y redes neuronales, con licencia *GNU Public License* [8]. El entorno NEST proporciona una interfaz de alto nivel para la creación de estructuras neuronales, a la vez que una gran cantidad de modelos neuronales y sinápticos [9]. Es capaz de soportar paralelización híbrida con procesos MPI (Message Passing Interface) y *multithreading*.

NEURON

NEURON proporciona un potente y flexible entorno para la implementación de modelos biológicos realistas –de señalización eléctrica y química– de neuronas y redes neuronales [10].

El procesamiento de información en el cerebro se produce a través de la difusión e interacción de señales eléctricas y químicas entre neuronas y dentro de ellas. Este proceso implica mecanismos no lineales, los cuales abarcan un ancho rango de escalas espaciales y temporales [11]. Consecuentemente, las ecuaciones que describen los mecanismos del cerebro generalmente no tienen soluciones analíticas, y por tanto, dificulta la comprensión del funcionamiento de las células y circuitos neuronales. Para acometer estos problemas, se ha diseñado la herramienta NEURON, que permite la creación de modelos biológicos cuantitativamente realistas de los mecanismos del cerebro, y además, posibilita la simulación de forma eficiente de cómo operan esos mecanismos. En este contexto, el término “biológicamente realista” no significa “infinitamente detallado”. En lugar de esto, significa que la elección de qué detalles se incluyen en el modelo y cuáles se omiten son decisión del investigador que construye el modelo, y por tanto, no son forzados por el programa de simulación. Para científicos e investigadores, una potente herramienta de simulación como NEURON, puede llegar a ser de ayuda indispensable para desarrollar las ideas teóricas y poder así profundizar en el conocimiento de los procesos químicos y eléctricos que se producen en el cerebro.

Brian Spiking Neural Network Simulator

Brian es un proyecto de software que permite el desarrollo de simulaciones de redes neuronales de impulsos [12]. El proyecto se distribuye mediante un paquete Python de código libre. El objetivo principal es la minimización del tiempo de desarrollo, mientras que la velocidad de ejecución es un objetivo secundario. Los investigadores deben especificar los modelos neuronales y sinápticos, en forma de ecuaciones matemáticas, creando grupos de neuronas y conectándolos mediante la sinapsis, ofreciendo además la posibilidad de añadir nuevos modelos que no están incluidos en el simulador.

GENESIS

GENESIS (The GEneral NEural Simulation System) es un entorno de simulación para la construcción de modelos realistas de sistemas neurobiológicos [13]. La simulación puede realizarse en varios niveles de abstracción: procesos subcelulares, neuronas individuales, redes neuronales y sistemas neuronales.

La primera versión se lanzó en julio de 1990, siendo uno de los primeros proyectos de código libre en el ámbito de la biología computacional.

2.2.2. Modelos de alto nivel

Los modelos de retina a alto nivel se encargan de describir el funcionamiento de la retina de forma matemática, mediante circuitos electrónicos, o mediante otra serie de procedimientos macroscópicos.

A continuación se detallan los proyectos más representativos de modelado de retinas a alto nivel.

La retina de Misha Mahowald

Misha Mahowald –junto con Carver Mead– diseñó un modelo electrónico de retina artificial que contribuyó a fundamentar y consolidar este área de conocimiento de manera notable [14]. En la figura 2.5 se presenta un elemento básico de dicho modelo de retina. La red de difusión está formada por la interconexión de un condensador y varias resistencias y se encarga de modelar el campo perceptivo que nutre una célula horizontal. La arquitectura propuesta por la investigadora Misha Mahowald se implementa sobre un chip *VLSI* (Very-large-scale integration) analógico.

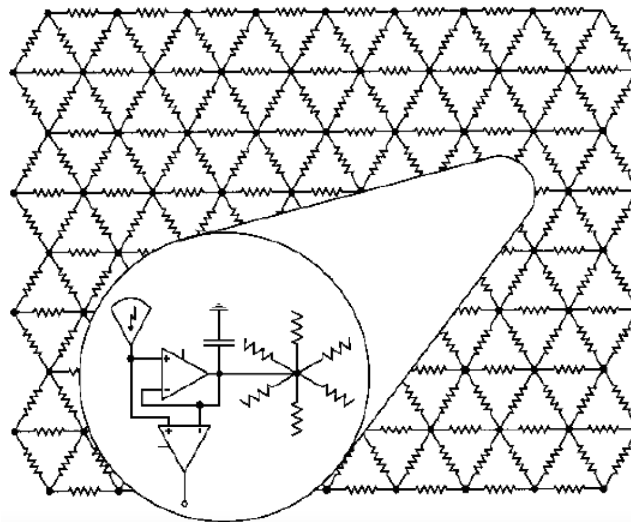


Figura 2.5: Modelo de Retina propuesto por Misha Mahowald y Carver Mead.

El modelo de Meister - Ammermuler

El modelo propuesto por Meister - Ammermuler [15, 16, 17, 18] modela la retina mediante un filtrado espacio-temporal de la entrada. En su trabajo se propone una arquitectura que consiste en un estímulo espacio-temporal como entrada, un conjunto de filtros paso banda espaciales y un filtro paso alto temporal, estando ponderados cada uno de ellos, como puede verse en la ecuación 2.1. La respuesta espacial $S(x)$ se define mediante un filtrado de tipo *Diferencia de Gaussianas* o *DoG*.

$$Retina(x, t) = S(x) \cdot T(t) \quad (2.1)$$

El modelo de visión de Itti

El modelo de atención selectiva desarrollado por Laurent Itti se basa en la aplicación de un conjunto de filtros espacio-temporales sobre una imagen de entrada, para posteriormente procesarlos mediante reglas de aprendizaje, determinando de esta forma las zonas más destacadas en cuanto al mecanismo biológico de atención [19]. Dentro del conjunto de filtros que se utilizan se encuentran detectores de sombra, detectores de movimiento, detectores de contorno, filtros sensibles a la orientación, filtros multicromáticos, etc.

Biological vision based approach

El proyecto presentado por Alexandre Benoit (junto con otros investigadores del *GIPSA Lab* [20]) pretende mostrar las ventajas del uso del modelado para el desarrollo de módulos bioinspirados, rápidos y eficientes, para el procesamiento de imágenes a bajo nivel [21]. Este modelo es denominado por ellos como “biological vision based approach”.

La idea básica es la imitación del sistema visual humano (HVS) mediante el modelado de alguna de sus partes, para el desarrollo de módulos de procesamiento de imagen a bajo nivel. Hasta ahora, las partes que mejor se conocen del sistema visual humano son la retina y la corteza visual primaria (V1), y es en ellas donde se centra su trabajo. La retina, considerada como un sistema de preprocesamiento, puede ser simulada mediante un análisis de alto nivel y por su parte, la corteza visual primaria puede ser simulada mediante un análisis a bajo nivel.

Virtual Retina

El simulador *Virtual Retina* está basado en el estado del arte conocido sobre el modelado y simulación de retinas, con la adición de la capacidad de simulación a gran escala [22]. El procesamiento de la retina expone un gran número de características que son fundamentales para la interpretación del área cortical, como el filtrado paso banda, control de ganancia, sincronismo de impulsos, etc [23]. En dichas implicaciones funcionales es donde se centra el simulador.

Virtual Retina es un modelo funcional simplificado con respecto a la complejidad fisiológica de la retina. Sin embargo, en oposición a los modelos lineales, el simulador incluye un mecanismo no lineal de control de ganancia de contraste. Además, mientras los modelos lineales se utilizan generalmente para ajustar las respuestas experimentales de unas pocas células experimentales, el modelo funcional propuesto es adecuado para simulación a gran escala.

2.3. Descripción del modelo de retina diseñado

Uno de los principales retos de este trabajo, es el diseño y desarrollo de una retina sintética que sea capaz de transformar el mundo visual externo frente a una persona ciega, en un conjunto de señales eléctricas que puedan utilizarse para estimular, en tiempo real, las neuronas de su corteza visual. Estas señales deben ser lo más similares posible a las señales de salida de la retina biológica. Estas tareas son comparables a los actuales desafíos que enfrentan las interfaces cerebro-ordenador y sus tecnologías inherentes [24]. La descripción completa de este problema ha sido discutida en otras publicaciones científicas [25, 26, 4, 27] y está fuera del alcance de este trabajo, pero la figura 2.6 resume los bloques de procesamiento básicos del modelo de retina bioinspirado del que se va a hacer uso en este trabajo, el cual se enmarca dentro del grupo de modelos de retina de

alto nivel. La razón por la cual se ha escogido un diseño de alto nivel es porque se desea llegar a producir un modelo de retina que sea fácilmente optimizable, que pueda correr en tiempo real (produciendo la salida del sistema en intervalos de tiempo biocompatibles) y que permita una implementación portable.

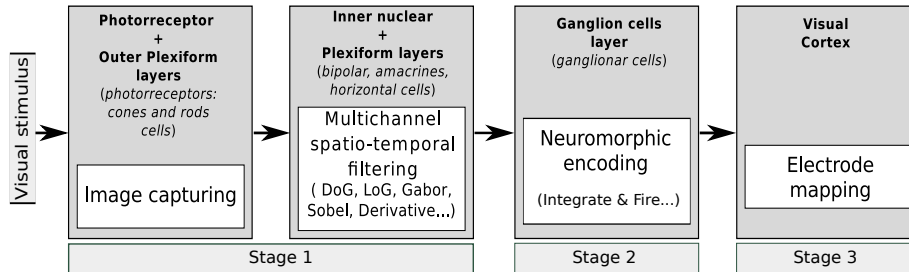


Figura 2.6: Bloques funcionales del procedimiento del modelo de retina bioinspirado bajo estudio.

Este modelo está basado en los registros electrofisiológicos de poblaciones de células ganglionares de retina, y es capaz de trabajar en tiempo real. Sin embargo, antes de proceder con la experimentación animal *in-vivo*, es necesario realizar una precisa afinación de los modelos de retina implicados. De esta forma, se habla de afinar un modelo de retina al proceso mediante el cual se realiza un ajuste paramétrico de aquellas funciones que definen el modelo de retina, y que mejor correspondencia consigue con los registros biológicos. Por tanto, se necesitan métricas de calidad adecuadas para evaluar el ajuste de los modelos sintéticos de retina.

Desde un punto de vista matemático, las operaciones de los dos primeros bloques de procesamiento de la figura 2.6, etiquetadas como *Stage 1*, pueden ser modeladas como una combinación ponderada de diferentes filtros de imágenes (ampliamente conocidos) convolutivos espacio-temporales, tales como filtros Gaussianos, Diferencia de Gaussianas (DoG), filtros de Gabor, operador Sobel, etc (ver *Stage₁* en la ecuación (3.1)). Esta combinación resulta en una matriz de actividad que alimenta al modelo *Integrate & Fire*, el cual calcula los patrones específicos de disparos que deben ser enviados a cada electrodo individual (ver *Stage₂* en la ecuación (3.2)). El bloque etiquetado como *Stage 3*, el cual representa la asignación nuevamente del potencial a cada electrodo, está fuera del alcance de este trabajo. Cada bloque de procesamiento del modelo de retina (*Stage1 + Stage2*) tiene muchos parámetros que deben ser afinados. Muchos de ellos se mueven en un rango continuo dinámico (por ejemplo, el parámetro σ de los filtros DoG) y otra parte de ellos pueden ser modelados como números naturales (por ejemplo, el tamaño del kernel ($N \times N$) de un filtro convolutivo). Con un espacio de búsqueda infinito que explorar, el proceso de ajuste paramétrico, representa un problema difícil de resolver. Para solucionar este problema, se propone el uso de una estrategia evolutiva, automática y multiobjetivo que aproveche las sinergias entre las redes neuronales y la computación evolutiva [28]. Esta estrategia permite explorar las ventajas y desventajas comparativas entre todos los objetivos, así como la generación de modelos de retina que mejor se aproximen a los registros biológicos reales, todo ello en un tiempo de cálculo factible.

Debido a la limitación del número de electrodos que forman el implante cerebral, es necesario realizar una reducción de la resolución de la imagen, divi-

diéndola en campos receptivos para cada electrodo, produciendo así una *matriz de actividad*. En la figura 2.7 [27] se observan las etapas del proceso descrito. La imagen de arriba a la izquierda muestra la entrada obtenida mediante una cámara de vídeo USB. La imagen se trata mediante el proceso de filtrado implementado por el modelo de retina, obteniéndose la imagen de arriba a la derecha. La imagen de abajo a la izquierda muestra la matriz de actividad de la entrada una vez filtrada. Por último, la imagen de abajo a la derecha muestra la salida del modelo neuronal de integración y disparo, explicado a continuación, de la matriz de actividad.

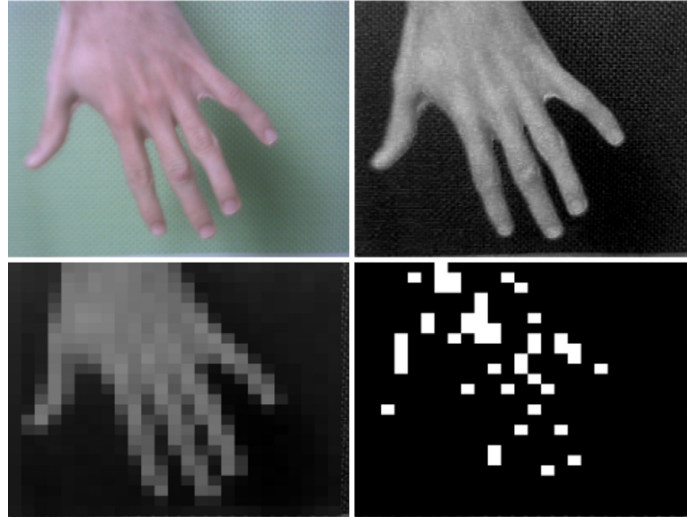


Figura 2.7: Etapas del proceso de filtrado y modelado hasta la obtención de la salida del modelo neuronal de integración y disparo.

En la figura 2.8 se presenta otro ejemplo de una posible salida del modelo neuronal de integración y disparo, en un instante de tiempo t , cuya matriz de actividad está formada por 10×10 componentes. Cada uno de los componentes estará asociado a un electrodo distinto, a través de los cuales se generarán las señales eléctricas.

El último paso del codificador de retina es la codificación neuromórfica. En el siguiente punto se detalla el modelo neuronal de integración y disparo y este proceso.

2.3.1. Modelo neuronal de integración y disparo

El modelo neuronal de integración y disparo (IaF) es uno de los modelos más ampliamente utilizados para el análisis del comportamiento de los sistemas neuronales. Describe el potencial de la membrana de una neurona en términos de activaciones sinápticas y la corriente inyectada que reciben [29]. El potencial de acción, llamado *spike*, se genera cuando el potencial de la membrana alcanza un valor umbral. El modelo IaF describe el potencial de acción de la membrana celular mediante un circuito eléctrico, formado por una resistencia y un condensador en paralelo, representando la capacitancia y la fuga en la membrana. En este modelo, el condensador de la membrana se carga hasta que alcanza un cierto valor umbral, en cuyo instante se descarga, produciendo un potencial de acción o *spike* y reiniciando el potencial. En la figura 2.9 [29] se

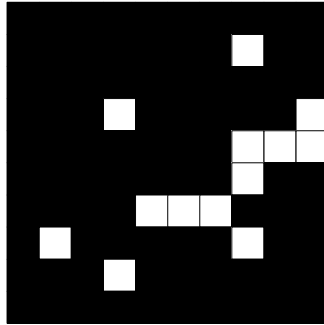


Figura 2.8: Ejemplo de salida del modelo neuronal de integración y disparo de una matriz de actividad 10×10 .

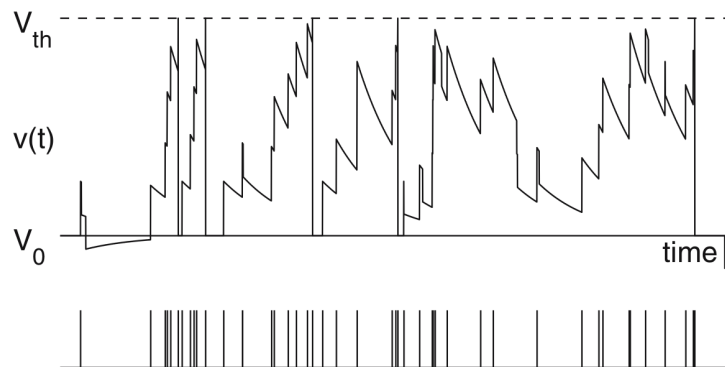


Figura 2.9: Descripción del modelo neuronal de integración y disparo.

describe la relación entre la entrada sináptica (imagen inferior), el potencial de la membrana (imagen intermedia) y los disparos de salida (imagen superior). La línea discontinua de la imagen intermedia representa el valor umbral.

En el proceso de la codificación neuromórfica, tras haber obtenido un mapa con las características relevantes de la imagen y haber reducido su resolución hasta una matriz de actividad, se traslada esta codificación de intensidad a una codificación frecuencial, acorde con las representaciones biológicas de la corteza visual. La figura 2.10 describe la operativa básica del módulo de integración y disparo ante un estímulo de entrada simple. El comportamiento del modelo es definido por el umbral para iniciar el disparo de un impulso, el potencial de reposo en el período refractario (por ejemplo, tras un reinicio de la señal) y el término de perdidas.

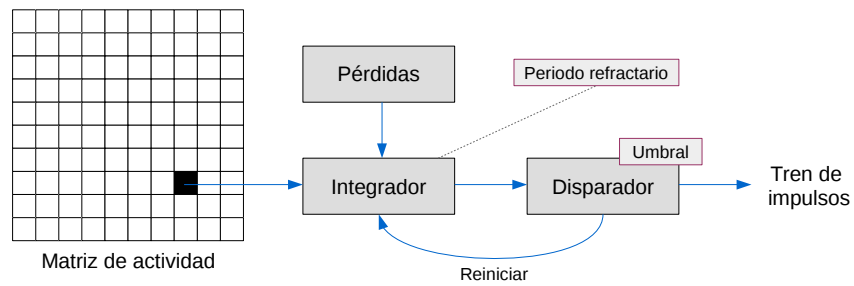


Figura 2.10: Codificación frecuencial de la matriz de actividad.

El modelo neuronal de integración y disparo se ha establecido como un modelo canónico para la descripción del potencial de acción de las neuronas, ya que puede ser analizado de forma matemática mientras que al mismo tiempo puede ser suficientemente complejo para modelar otras características esenciales del procesamiento neuronal.

Capítulo 3

Estrategia evolutiva multiobjetivo de inferencia de modelos de retina

3.1. Introducción

Como se ha introducido en la sección 2.3, considerando el vasto número de parámetros que deben ser tenidos en cuenta, el problema del modelado de una retina genética no puede ser llevado a cabo mediante la exploración exhaustiva de todo el espacio de soluciones. Además, no sólo uno, sino varios objetivos antagónicos entran en juego para asegurar la calidad del modelo de retina, y por tanto, el problema se convierte en multiobjetivo. Por ende, se propone utilizar un algoritmo genético para guiar la exploración del espacio de soluciones y un procedimiento multiobjetivo para evaluar cada solución alcanzada en el proceso de exploración. La bien conocida estrategia propuesta, se basa en la asunción de que la evolución, puede ser utilizada como una herramienta de optimización para un problema multiobjetivo. La idea es que una población de posibles soluciones candidatas evolucione, mediante procedimientos inspirados en la variación genética y la selección natural. En este caso, la población de soluciones candidatas se define como un conjunto de modelos de retina. Por tanto, la optimización multiobjetivo (MOO) basada en un Algoritmo Genético (MOOGA) será adoptada como estrategia de exploración del inmenso espacio de soluciones.

3.2. Algoritmos Genéticos

Los Algoritmos Genéticos (GAs) son métodos adaptativos, generalmente utilizados en problemas de búsqueda y optimización de parámetros. Pertenecen al grupo de métodos conocidos como Algoritmos Evolutivos (EAs), los cuales se basan en la imitación de los procesos evolutivos, tales como la selección natural, la recombinación o la mutación. Cada individuo de la población, que en nuestro proyecto se inicializa de forma aleatoria, representa una posible solución al problema. En consecuencia, los mejores individuos evolucionan a partir de las generaciones previas hasta obtener una solución óptima o pseudo-óptima.

Como se muestra en la figura 3.1, la población evoluciona a través de las operaciones de recombinación y mutación, y solamente aquellos individuos que representan las mejores soluciones, pueden ser extraídos de la población super-

viviente. El proceso de selección, elige de la población aquellos individuos que serán capaces de reproducirse, siendo los mejores individuos aquellos que maximicen o minimicen los objetivos, de acuerdo con su naturaleza. En este caso, varios objetivos son considerados al mismo tiempo, por tanto, se utiliza una estrategia MOO para clasificar a los individuos. La operación de recombinación intercambia subpartes de una pareja de individuos y los recombina, ya que ésta es la manera de imitar la recombinación biológica. Finalmente, la operación de mutación cambia, de forma aleatoria, los valores de algunos alelos del cromosoma.

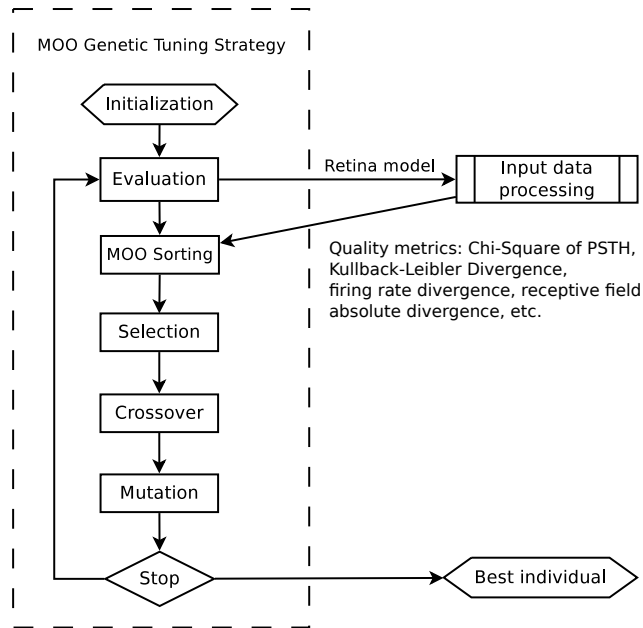


Figura 3.1: Pasos de la estrategia de modelado multiobjetivo.

3.2.1. Elementos de un Algoritmo Genético

A continuación se describen los principales elementos de un Algoritmo Genético, para que de esa forma el lector se familiarice con los conceptos que serán utilizados a lo largo del trabajo.

Población inicial

Todo algoritmo genético parte de una población inicial, la cual puede ser generada de diversas formas: generación aleatoria de cada individuo, utilización de un algoritmo voraz, generación aleatoria de una parte del individuo y posteriormente aplicación de una búsqueda local, etc.

Operador de mutación

Esta operación se encarga de generar nuevos individuos a partir de la modificación de uno o varios genes de una solución existente, produciendo así una mayor variedad dentro de los elementos de la población.

Operador de cruce

Se denomina cruce al proceso de recombinación realizado para producir la descendencia que se insertará en la siguiente generación. El cruce es una estrategia de reproducción sexual.

Función de evaluación

La función de evaluación es la encargada de medir la adaptación de cada uno de los individuos. Esto es equivalente a decir que se encarga de determinar cuán bueno es un individuo como solución al problema. Por tanto, es la función a optimizar, la cual dependerá de la información codificada en los genes del individuo.

Operador de selección

El operador de selección se encarga de que la información útil se transmita de generación en generación, conservando aquellos individuos mejor adaptados y que por tanto presentan un mejor resultado ante la función de evaluación. No obstante, también es necesario incluir un factor aleatorio que permita reproducirse a individuos que aunque estén peor adaptados, puedan contener alguna información útil para posteriores generaciones, consiguiendo de esta forma mayor diversidad en la población.

3.2.2. Algoritmo NSGA-II

Para resolver problemas de optimización multiobjetivo (MOO), se han propuesto diferentes métodos de resolución en los últimos años. El algoritmo NSGA-II (Non-dominated Sorting Genetic Algorithm II) ha sido reportado como uno de los mejores algoritmos MOO [30, 31], siendo incluido dentro del grupo de algoritmos elitistas que enfatizan la eficiencia computacional, de igual manera a como la realizan los algoritmos SPEA y SPEA2 [32]. El algoritmo NSGA-II establece una relación de orden entre los individuos de una población, basada principalmente en el concepto de no dominación o frentes de Pareto. Se dice que una solución X_i domina a otra solución X_j si el primer individuo es mejor o igual que el segundo individuo en cada objetivo único y, al menos, estrictamente mejor en uno de ellos. Es decir, los frentes de Pareto son definidos por aquellos puntos en los cuales no existe mejora en un único objetivo sin degradar el resto de objetivos.

El algoritmo NSGA-II primero agrupa los individuos en un primer frente que contiene todos los individuos no dominados, esto es, el frente de Pareto. Entonces, un segundo frente se construye seleccionando todos aquellos individuos que no son dominados en ausencia de los individuos del primer frente. Este proceso es repetido de forma iterativa hasta que todos los individuos son situados en algún frente. La función distancia de hacinamiento es usada para calcular la diversidad de una posible solución, y su propósito es mantener una buena distribución de las soluciones. Tras esto, los individuos del frente de Pareto son ordenados, en orden descendente, basándose en el valor de su distancia de hacinamiento. Como resultado, aquellas soluciones que tienen mayor diversidad son priorizadas.

En resumen, tras el resultado de la ejecución del algoritmo NSGA-II, la salida proporciona la población ordenada por frentes de Pareto no dominados y después, por la distancia de hacinamiento. La figura 3.2 muestra el proceso de selección mediante el uso del algoritmo NSGA-II.

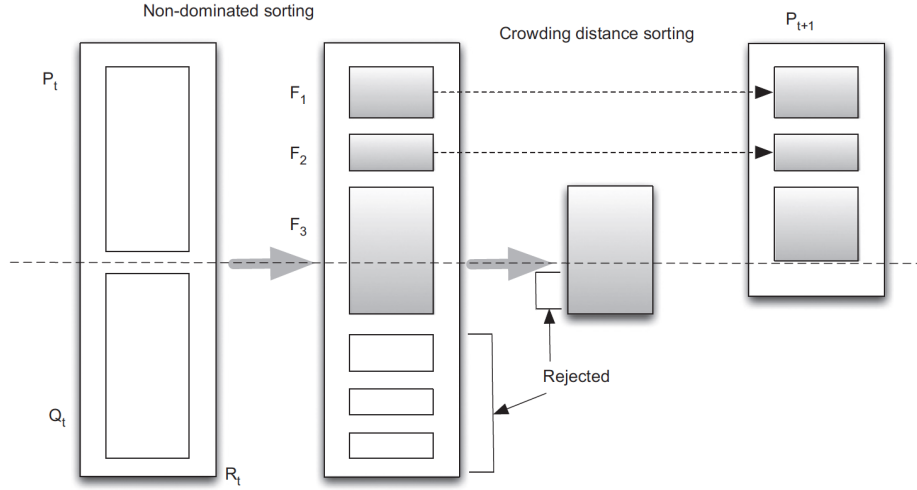


Figura 3.2: Proceso de selección del algoritmo NSGA-II.

3.2.3. Codificación de los cromosomas

En nuestro planteamiento, cada individuo representa un posible modelo de retina, y por tanto, cada gen codifica los valores de los parámetros de la función que modela la retina. Como ejemplo, las ecuaciones 3.1 y 3.2 representan un modelo de retina genérico, como se ha descrito en la sección 2.3. Los parámetros de cada función serán los candidatos a ser codificados como los genes que compondrán el cromosoma.

$$\begin{aligned}
 Stage_1 &= N^i \cdot f_{Gauss}^i(\sigma_i, \mu_i, \kappa_i) + \\
 &N^{i+1} \cdot f_{Gauss}^{i+1}(\sigma_{i+1}, \mu_{i+1}, \kappa_{i+1}) + \dots \\
 &+ N^{i+M} \cdot f_{DoG}^{i+M}(\sigma_1^{i+M}, \sigma_2^{i+M}, \mu_1^{i+M}, \mu_2^{i+M}, \kappa_1^{i+M}, \\
 &\kappa_2^{i+M}) + \dots + N^{i+K} \cdot f^{i+K}(\dots) + \dots; \quad (3.1)
 \end{aligned}$$

$$\begin{aligned}
 Stage_2 &= IaF(Stage_1; CellType, Threshold, \\
 &Leakage, RefractoryPeriod, \\
 &Persistence); \quad (3.2) \\
 &i, M, N, K \in \mathbb{N}
 \end{aligned}$$

La figura 3.3 muestra el modo mediante el cual cada gen es codificado y forma parte del cromosoma. Como puede observarse, cada uno de los genes que componen el cromosoma representa el valor de uno de los parámetros de las diversas funciones que modelan la retina.

3.2. Algoritmos Genéticos

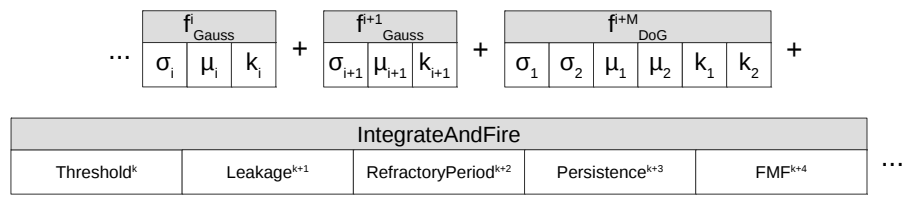


Figura 3.3: Ejemplo de codificación de un individuo.

Capítulo 4

Herramientas y métodos

En este capítulo se presenta el diseño del software que se ha desarrollado, tanto para el modelado de retinas como para la simulación del proceso evolutivo descrito en el capítulo 3. Además, también se detallan los distintos recursos tecnológicos –lenguajes de programación, librerías, hardware– que han sido empleados en el desarrollo del proyecto.

4.1. Diseño del software de modelado y simulación

En el proceso de diseño se tomó la decisión de crear un modelo de diseño evolutivo que se adaptara a cualquier modelo de retina publicado. La idea era crear un motor de simulación genético común que pudiera ajustar paramétricamente cualquier tipo de retina que quisiera modelarse. En el presente trabajo, únicamente se ha modelado un modelo de retina, pero como se puede observar en la figura 4.2, se han añadido como ejemplo (en color gris) otros posibles modelos de retina publicados, los cuales podrían ser añadidos al simulador y dar pie a nuevos casos de estudio.

Con el fin de crear un simulador escalable que pudiera integrar varios tipos de modelos de retina, se ha utilizado el patrón de diseño GoF *Factory Method*. El propósito del patrón *Factory Method* es definir una interfaz para la creación de un objeto, permitiendo a las subclasses decidir qué clase instanciar. En la figura 4.1 se presenta la estructura –es decir, el diagrama de clases UML– del patrón de diseño bajo estudio. Como puede observarse, si se compara con la figura 4.2 se observa que se ha utilizado el patrón *Factory Method* para abstraer la creación y uso de los modelos de retina concretos a la clase que se encarga de implementar la parte evolutiva del proyecto, evitando así la dependencia y por tanto, consiguiendo una alta cohesión y un bajo acoplamiento.

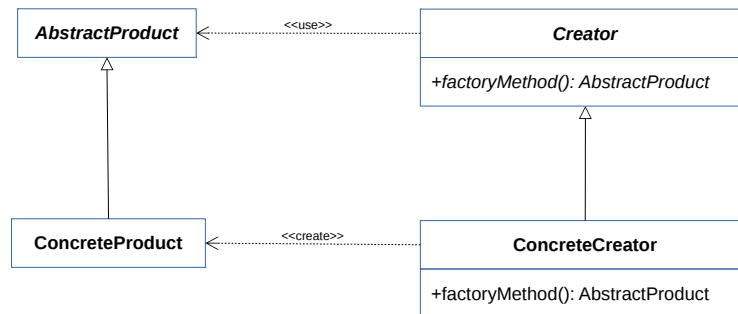


Figura 4.1: Estructura del patrón de diseño *Factory Method*.

El único requisito para cada modelo de retina que quiera ser implementado, es la definición de las funciones genéticas necesarias para que el motor de simulación genético pueda operar. Éstas son las funciones necesarias: evaluación, mutación y cruce. A continuación, en el listado 4.1 se presenta la función en la que se registran dichos operadores genéticos.

```

def __create_genetic_operators(self):
    """ Registering the operators and their default arguments
        in the toolbox is done as follow.
    """

    self.toolbox.register("evaluate",
                          self.retina_model_handler.evaluate)
    self.toolbox.register("mate",
                          self.retina_model_handler.crossover)
    self.toolbox.register("mutate",
                          self.retina_model_handler.mutate)
    self.toolbox.register("select",
                          tools.selNSGA2)
    
```

Listado 4.1: Función de registro de operadores genéticos.

En el anexo 7.1 se adjunta el código fuente de la clase *MoogaRetina.py*, que es la que implementa el motor evolutivo del proyecto.

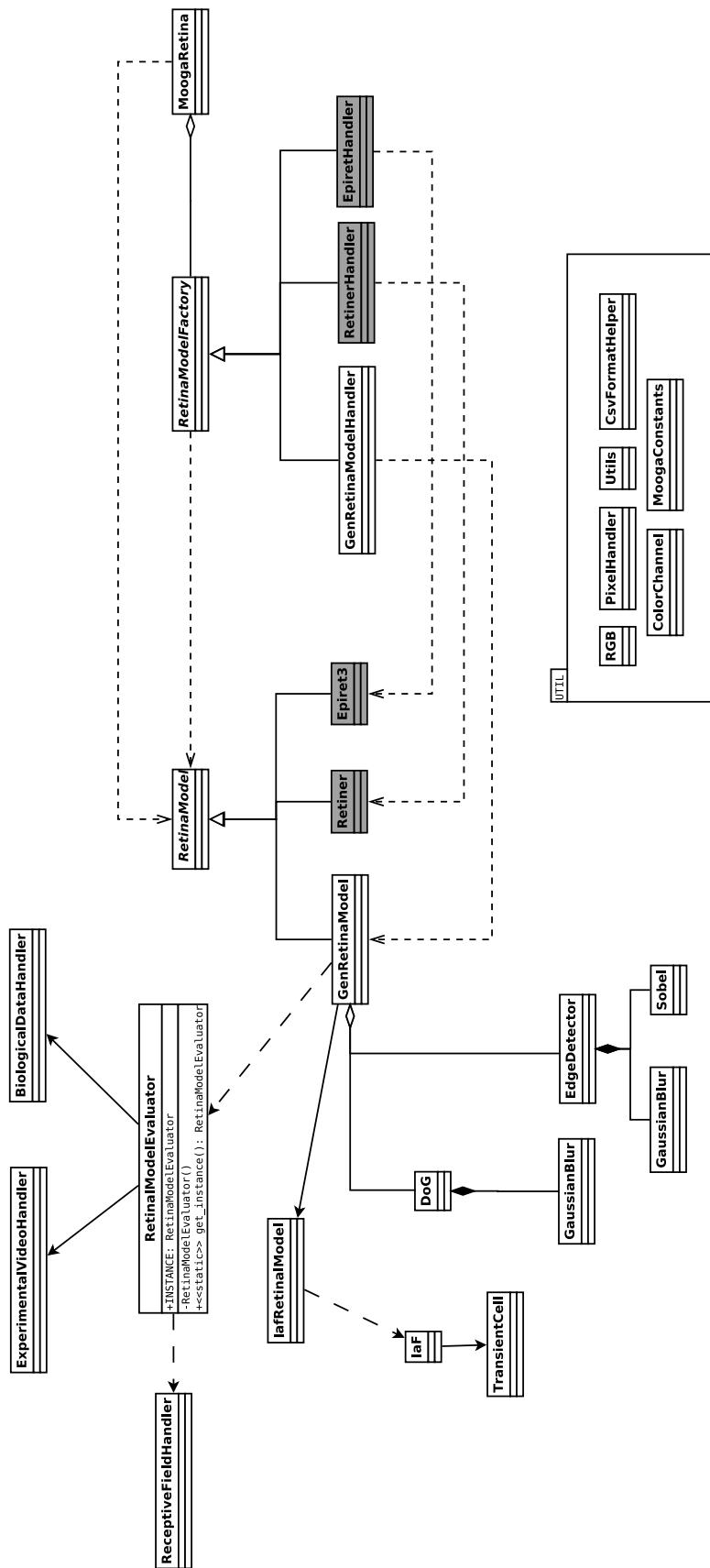


Figura 4.2: Diagrama de clases UML.

4.2. Tecnologías

4.2.1. Python

Python es un lenguaje de programación interpretado, que fue desarrollado a finales de los 1980s por Guido van Rossum. Este lenguaje de programación se deriva en gran parte del ABC, un lenguaje desarrollado en los 1980s para la enseñanza. Python presenta una amplia variedad de características de lenguajes de programación como C++, Java, Modula-3 y Scheme. Por esta razón, Python es tan atractivo para programadores profesionales, científicos, investigadores y educadores.

La elección de Python como lenguaje de programación para la realización del proyecto se ha basado en varios argumentos. El primero de ellos es que es un lenguaje perfecto para un equipo multidisciplinar, como es el nuestro, ya que la curva de aprendizaje es mucho menor que la de otros lenguajes como C o Java. Además, los estímulos visuales diseñados y desarrollados por el grupo de investigación de la UMH [33] están implementados también en Python, y es importantísimo aprovechar el KNOW-HOW. La flexibilidad, simplicidad, rapidez y el ecosistema de librerías también han sido argumentos de peso para su elección. Además, ofrece la posibilidad de exponer partes del código que hayan sido desarrolladas en otros lenguajes, como pueden ser C o C++, lo que nos daría viabilidad para desarrollar funciones a bajo nivel (como filtros, procesamiento de datos, etc) y que éstas se integraran sin ningún problema en el proyecto.

4.2.2. SciPy y NumPy

SciPy es un ecosistema open-source basado en Python para matemáticas, ciencia e ingeniería. Dentro del ecosistema de librerías que lo conforman, se han utilizado en el presente proyecto dos de ellas: *SciPy library* y *Numpy*.

SciPy library es uno de los paquetes que conforman el núcleo de *SciPy*. Provee varias rutinas eficientes e intuitivas para cálculo numérico, integración y optimización.

Numpy es el paquete fundamental para trabajar con Python en el área de computación científica, básicamente gracias al manejo y tratamiento de arrays N-dimensionales, álgebra lineal, números aleatorios y transformadas de Fourier.

4.2.3. Biblioteca de primitivas de procesamiento de imagen

OpenCV (Open Source Computer Vision Library) es una librería open-source bajo licencia BSD que incluye cientos de algoritmos de visión artificial y procesamiento de imagen. Se utiliza en una infinidad de campos, entre ellos la investigación. Debido al gran número de filtros que implementa, se ha optado por utilizarla para modelar las operaciones de los dos primeros bloques de procesamiento de la figura 2.6, como son los filtros Gaussianos, Gabor, Sobel, etc.

4.2.4. Machine Learning in Python: scikit-learn

Para el cálculo del área de los campos receptivos de las células ganglionares, explicado en la sección 5.3, se ha utilizado la librería de código abierto *scikit-learn* (licencia BSD) para el lenguaje de programación Python [34]. Cuenta con algoritmos de clasificación, regresión y agrupamiento, así como soporte para máquinas vectoriales, extracción de características y minería de datos. Se ha

diseñado para interoperar con las librerías científicas y numéricas *Numpy* y *SciPy*.

4.2.5. Distributed Evolutionary Algorithms in Python

DEAP (Distributed Evolutionary Algorithms in Python) es un framework de computación evolutiva para prototipado rápido y experimentación de ideas nuevas [35]. En dicho ecosistema, se ofrecen de forma explícita y transparente los algoritmos y las estructuras de datos, para dotar a los programadores, investigadores y científicos toda la potencia de la herramienta. Además, puede integrarse con mecanismos de paralelización tales como multiprocesamiento y SCOOP [36].

Para el modelado de la optimización multiobjetivo y el algoritmo genético se ha escogido esta librería, principalmente por su potencialidad y versatilidad.

4.2.6. Vision Egg

Vision Egg es una librería de programación, que utiliza tarjetas de vídeo estándar, que permite producir estímulos visuales para experimentos en el campo de la investigación [37]. Es una interfaz de alto nivel entre Python y OpenGL. Además de métodos para la generación automática de estímulos visuales, tiene una serie de funciones para el manejo de datos numéricos, imágenes, vídeos, texto y objetos 3D.

Esta librería se ha utilizado para generar los estímulos visuales que son los input de entrada, tanto para las retinas biológicas de ratón, como para las retinas sintéticas (en forma de vídeo).

4.2.7. Clúster de computación

Todos los experimentos han sido realizados en el clúster de computación del *Instituto Universitario de Investigación Informática* de la *Universidad de Alicante* [38], cofinanciado por el proyecto 'Construcción y optimización automáticas de bibliotecas paralelas de computación científica (TIN2008-06570-C04-04)', del Ministerio de Ciencia e Innovación de España y el Vicerrectorado de Investigación, Desarrollo e Innovación de la *Universidad de Alicante*.

El clúster HPC está formado por 26 nodos de cálculo HP Proliant SL 390 G7 (1x2U, 24x1U) con las siguientes prestaciones:

- 2 x CPU Intel Xeon X5660
 - 6 cores de 2,80 Ghz, Max Turbo 3,2 Ghz, 12 MB cache Level 2, 64 bit, 32nm.
- Front Bus 800-1333 MHz
- Memoria 48 GB DDR3-1333
- 1 disco SFF SATA 7,2 k, 500GB
- Dual port 1GbE NIC
- 10GbE SFP+ (via ConnectX2)

Sistema Operativo

El sistema operativo utilizado para llevar a cabo los experimentos es Debian 8.0 para la arquitectura x86 de 64 bits.

Capítulo 5

Casos de estudio

Para evaluar la viabilidad y calidad de la estrategia MOOGA propuesta en el ajuste paramétrico del modelo de retina, se han diseñado dos extensos casos de estudio. En el primer caso de estudio se evalúa la convergencia del modelo de retina a través de las cuatro métricas propuestas, mientras que en el segundo se presenta el ajuste del modelo de retina, haciendo uso de dichas métricas, en un problema multiobjetivo. En ambos experimentos, el objetivo ha sido demostrar la relevancia de un ajuste paramétrico óptimo para un modelo de retina dado, con el fin de conseguir la mejor correlación con los registros electrofisiológicos.

5.1. Setup experimental

Se han elegido varios estímulos visuales que nutren a los modelos de retina sintetizados de forma automática, generando registros de salida que se aproximan a los registros biológicos en virtud de una serie de métricas. Para la clasificación de las células ganglionares de la retina como tipo ON, OFF, ON/OFF [39] se han utilizado repeticiones de $700ms$ de un flash blanco ($196.25 \text{ cd}/m^2$) seguido por un periodo de oscuridad de $2300ms$. En la figura 5.1 se muestra el gráfico de dicho estímulo, en el que los flashes de color blanco se definen por el valor 255 en la escala de grises, y el periodo de oscuridad por el valor 0 en la escala de grises. Tras esto, las retinas aisladas han sido estimuladas con barras blancas con un ancho de $250\mu\text{m}$ cruzando una imagen en negro, con una frecuencia de 0.5 & 1Hz, con la finalidad de localizar los campos receptivos según el procedimiento descrito en el artículo [40]. Con los flashes se clasifican las respuestas de las células ganglionares en ON, OFF y ON/OFF, y con las barras se calculan los campos receptivos. Se han utilizado cuatro pares (ocho barras en movimiento) como estímulos: 0° , 45° , 90° , 135° , 180° , 225° , 270° y 315° . En la figura 5.2 se muestra el estímulo utilizado para la clasificación de las células ganglionares y en la figura 5.3 se muestran varias capturas de los estímulos de barras utilizados en los experimentos.

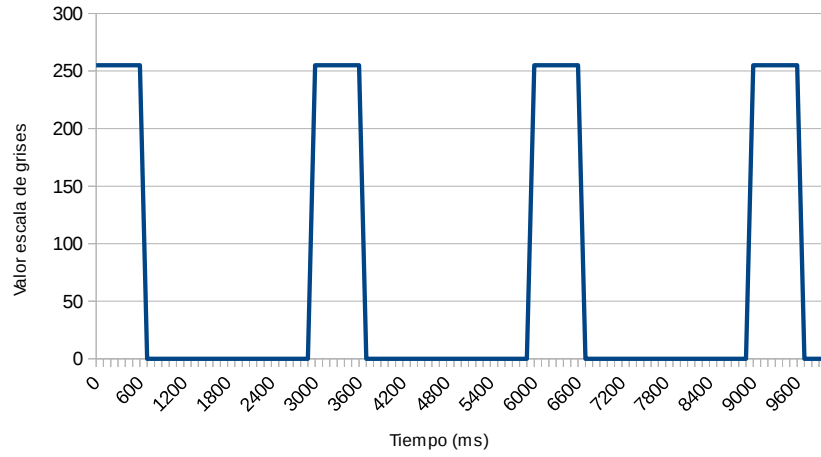


Figura 5.1: Estímulo utilizado para la clasificación de las células ganglionares.

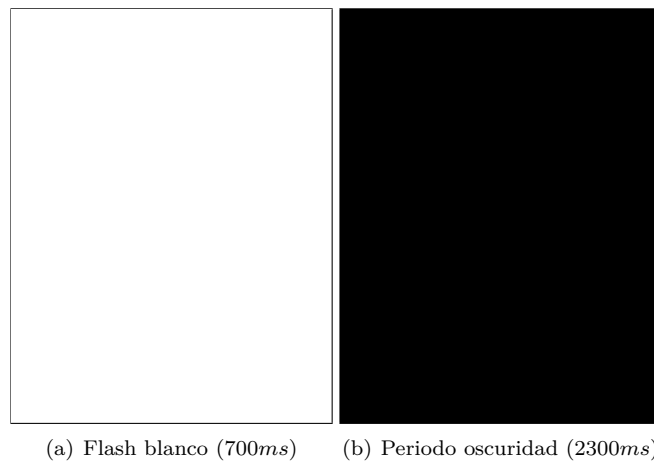


Figura 5.2: Representación del estímulo utilizado para la clasificación de las células ganglionares.

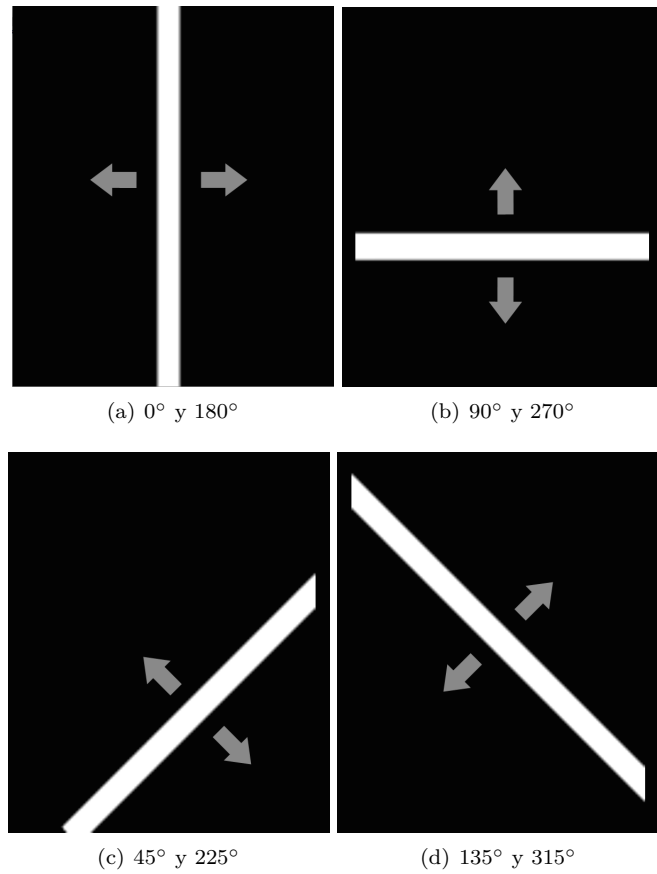


Figura 5.3: Representación de los estímulos con barras blancas utilizados en los experimentos. Las flechas de color gris indican la dirección del movimiento de las barras, desplazándose en ambos sentidos.

Los mismos experimentos han sido llevados a cabo con las retinas sintéticas utilizando estímulos idénticos, creados con *Vision Egg*. A continuación se ha desarrollado una caracterización detallada de las propiedades fisiológicas de los registros de las neuronas en ambos experimentos (registros electrofisiológicos reales y simulados). Con el fin de investigar los patrones de disparo de cada una de las neuronas se ha utilizado el *Peri-Stimulus Time Histogram* (PSTH), ya que permite medir la intensidad media de los disparos neuronales a través de varias etapas –bins o intervalos– (ver subsección 5.5.2).

En segundo lugar, el modelo de retina seleccionado para ser afinado paramétricamente, el cual se basa en la expresión general ya presentada en la sección 3.1, se define en las ecuaciones (5.1) y (5.2).

$$\begin{aligned}
 Stage_1 &= 0,5 \cdot f_{DoG}^1(\sigma_1^1; \sigma_2^1; \mu_1^1; \mu_2^1; \kappa_1^1; \kappa_2^1, R + B, 0,2 \cdot G) \\
 &+ 0,3 \cdot f_{DoG}^2(\sigma_1^2; \sigma_2^2; \mu_1^2; \mu_2^2; \kappa_1^2; \kappa_2^2, R + G, B) \\
 &+ 0,2 \cdot f_{LoG}^3(\sigma_1^3; \sigma_2^3; \mu_1^3; \mu_2^3; \kappa_1^3; \kappa_2^3, I)
 \end{aligned} \tag{5.1}$$

$$\begin{aligned}
 Stage_2 &= IaF(Stage_1; Threshold, Leakage, \\
 &RefractoryPeriod, Persistence, \\
 &FrequencyModulationFactor);
 \end{aligned} \tag{5.2}$$

Considerando que los estímulos seleccionados no tienen color (una barra de color blanco en movimiento sobre un fondo negro), se debe tener en cuenta que aquellos parámetros de la retina relacionados con el procesamiento del color tendrán un escaso o ningún impacto en los registros sintéticos. Consecuentemente, la combinación de colores de entrada, utilizando los canales de color R , G y B , y cada parámetro σ y μ de la contribución de la *Diferencia de Gaussianas*, han sido fijados de acuerdo con otros estudios [27, 41]. Por tanto, en este caso los valores han sido fijados de la siguiente forma: $\sigma_1^i = 0,9$, $\sigma_2^i = 1,2$, y $\mu_j^i = 0$ (filtros gaussianos centrados en cada célula ganglionar de la retina). El bloque $Stage_2$ del modelo de procesamiento de la retina (véase la figura 2.6) ha sido definido por el modelo de disparo neuronal *Integrate and Fire* (IaF), el cual se basa en el modelo *Leaky-Integrate&Fire* (*Leaky-IaF*) [42], con la adición única de un nuevo parámetro, *Persistence time*, el cual modela cuántas veces debe alimentar al modelo de retina cada fotograma de vídeo. Debido a que el número de *frames* del vídeo del estímulo visual es un parámetro discreto, con la finalidad de poder aproximar la naturaleza continua de la estimulación de la retina biológica, se ha decidido que este parámetro –*Persistence time*– modele el número de veces que es procesado cada *frame* del estímulo visual. Por último, la respuesta impulsiva de disparo se modela mediante el parámetro *FrequencyModulationFactor* (FMF). Este parámetro tiene el efecto de variar el patrón de disparo de una célula ganglionar de tipo *transient* alterando su tasa de disparo y el tiempo hasta llegar a una respuesta basal. En la sección 5.4 se describe cómo ha sido modelado el comportamiento de las células ganglionares *transient*.

Cada parámetro del bloque $Stage_1$ (véase la figura 2.6), y el tamaño del kernel de cada filtro del bloque $Stage_2$ han sido seleccionados para ser automáticamente modificados, lo que permite la afinación general del modelo. En cuanto al parámetro del tamaño del kernel, cada uno ha sido fijado con el mismo valor dentro de un mismo individuo. Es decir, cada κ_j^i de la ecuación (5.1) permanece constante dado un modelo de retina. La tabla 5.1 resume los parámetros que se han seleccionado para que cambien de forma automática, junto con su rango de variación. En el caso del tamaño del kernel K , sólo se utilizarán números impares dentro del intervalo de 3 a 13. Hay que destacar que, con la excepción de los parámetros *Persistence time* y K , el resto de parámetros se

mueven en un rango continuo, y por tanto, se presenta un espacio de búsqueda infinito. Los seis parámetros seleccionados han sido codificados de acuerdo al esquema de cromosoma presentado en la subsección 3.2.3.

Parámetro	Valor Mín.	Valor Máx.	Tipo de dato
$K(\text{odd number})$	3	13	int
Threshold	225.0	275.0	float
Leakage	10.0	15.0	float
Refractory period (ms)	1.0	10.0	float
Persistence time	3	7	int
FMF	0.25	0.40	float

Tabla 5.1: Parámetros del Cromosoma

Los parámetros del algoritmo genético han sido seleccionados como sigue: tamaño de la población – 60, probabilidad de mutación – 0.05, probabilidad de recombinación – 0.3 y 2500 iteraciones (o generaciones) para el caso de estudio presentado en la subsección 5.6, y 2500 iteraciones para el caso de estudio presentado en la subsección 5.7. Por tanto, una cantidad total de $60 \cdot 2500 = 150000$ modelos de retina han sido procesados en los casos de estudio 1 y 2. Cada uno de los experimentos ha necesitado un total de 917 horas para completarse (39 días de cálculo). La simulación y el cálculo de cada generación ha necesitado una media de 22 minutos, habiendo utilizado como sistema de procesamiento un Intel Xeon X5660 con 48GiB de RAM, detallado en la subsección 4.2.7.

Finalmente, para comparar los registros electrofisiológicos sintéticos y reales, y también para probar el comportamiento de la estrategia MOOGA propuesta, cuatro métricas de calidad, también llamadas funciones de fitness en el campo de los algoritmos genéticos, han sido utilizadas. En primer lugar, con el fin de comparar los datos de el PSTH, se han utilizado tanto la *Kullback-Leibler Divergence* (PSTH-KLD) como la *Firing Rate Absolute Difference* (FRAD) entre los dos vectores de datos PSTH. En segundo lugar, y en relación con los datos del ráster (la respuesta temporal de los disparos para cada célula ganglionar), también se ha utilizado la *Kullback-Leibler Divergence* sobre el *Inter-Spike Interval* (ISI-KLD) en los vectores de datos. Finalmente, la diferencia absoluta de las áreas de los campos receptivos (RFAD) ha sido calculada siguiendo el método propuesto en el estudio [40]. Las cuatro métricas comparativas se describen en la sección 5.5.

5.2. Registros electrofisiológicos

Para evaluar la estrategia MOOGA y obtener un modelo de retina ajustado, se han utilizado registros electrofisiológicos de células ganglionares de la retina, procedentes de poblaciones de ratones. Todos los procedimientos experimentales han sido realizados conforme al ARVO y a las Directivas del Consejo de las Comunidades Europeas (86/609/ECC) para el uso de investigación con animales.

Los ratones adultos de tipo salvaje (C57BL/6J) fueron criados dentro de una colonia local establecida a partir de parejas reproductivas compradas (Jackson Laboratories, Bar Harbor, ME). Los animales fueron adaptados a la oscuridad durante una hora, anestesiados con el 4% de isoflurano (IsoFlo[®], Esteve Veterinaria) y sacrificados mediante dislocación cervical. Después de la enucleación del ojo, el globo ocular se hemiseccionó con una cuchilla y la córnea y las lentes

fueron separadas desde la mitad posterior. Las retinas fueron entonces cuidadosamente retiradas de la copa ocular restante, montadas en una placa de agar para células ganglionares, cubierta con un filtro de corte Millipore. Tras este proceso, dicho montaje experimental fue colocado en una cámara de registro y perfundido con una solución Ringer a temperatura fisiológica. Todo el procedimiento fue realizado bajo una tenue iluminación de color rojo. Los registros extracelulares de las células ganglionares fueron obtenidos mediante el uso de una matriz de 100 micro-electrodos de $1,5mm$ de longitud (Utah Electrode Array).

Las respuestas simultáneas de las múltiples unidades fueron grabadas mediante un sistema de adquisición de datos (Bionic Technologies Inc) y almacenadas en un ordenador Pentium, para su análisis posterior. Los impulsos neuronales fueron detectados una vez fueron sobrepasados los umbrales establecidos en cada electrodo, utilizados en procedimientos estándares descritos en otros artículos [43, 44].

La clasificación y categorización de las diferentes unidades fueron realizadas mediante un software gratuito de código libre, basado en el método de análisis de componentes principales (PCA) junto con distintos algoritmos de agrupación [45]. Las marcas de tiempo, para cada uno de los potenciales de acción de cada unidad individual, fueron utilizadas para generar *peristimulus time histograms* (PSTH) y *peristimulus spike rasters*, utilizando la herramienta NeuroExplorer Versión 4 (Nex Technologies), así como software realizado expresamente para esta tarea [46].

El estímulo visual fue programado en Python utilizando una librería de código libre (*Vision Egg*) para la generación de estímulos visuales en tiempo real [37] y reproducido en un monitor ACER TFT de 16 bits a 60 Hz. Diferentes patrones de estímulos lumínicos fueron dibujados en este área, para ser proyectados a través de un divisor de haces, y enfocados en un área de la capa fotorreceptora de $4 \times 4mm$ con la ayuda de lentes ópticas. El proceso de creación y utilización de los estímulos visuales diseñados se detalla en la figura 5.4, donde se aprecia que el mismo estímulo creado es el que se utiliza tanto para alimentar las células biológicas, mediante su reproducción en el monitor TFT, como para alimentar las retinas sintéticas, utilizando un archivo de vídeo que el modelo de retina posteriormente procesa. Dicho archivo de vídeo es procesado por cada modelo de retina distinto que integra la población de individuos, y de donde se obtiene un conjunto de registros sintéticos para cada uno de los modelos de retina procesados por generación. Tras este proceso, se comparan los registros biológicos y sintéticos y de acuerdo al valor de las métricas obtenido, se establece un ordenamiento entre los modelos de retina, desechando aquellos modelos que peor se han adaptado, siguiendo el proceso evolutivo descrito en el capítulo 3.

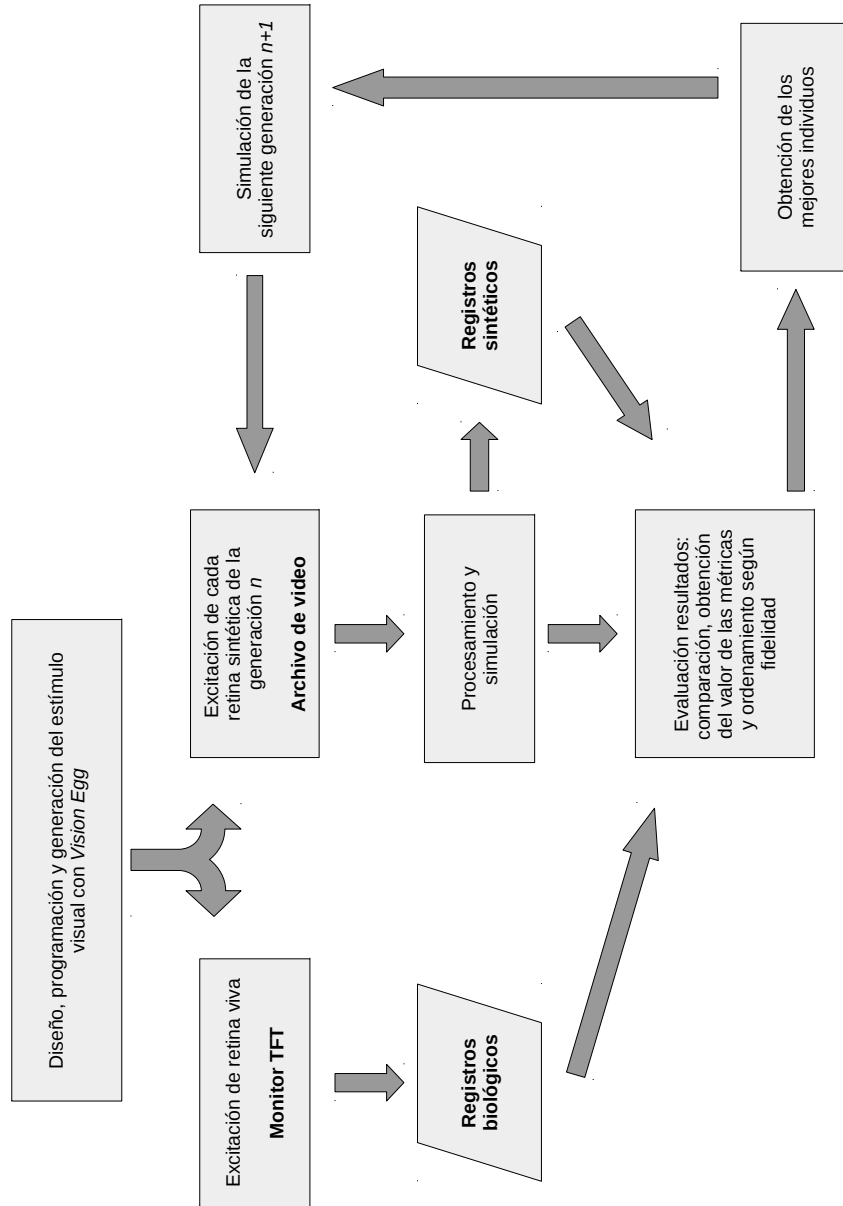


Figura 5.4: Descripción de la generación y utilización de los estímulos visuales creados con *Vision Egg* así como el diagrama de flujo del procesamiento de los registros biológicos y sintéticos.

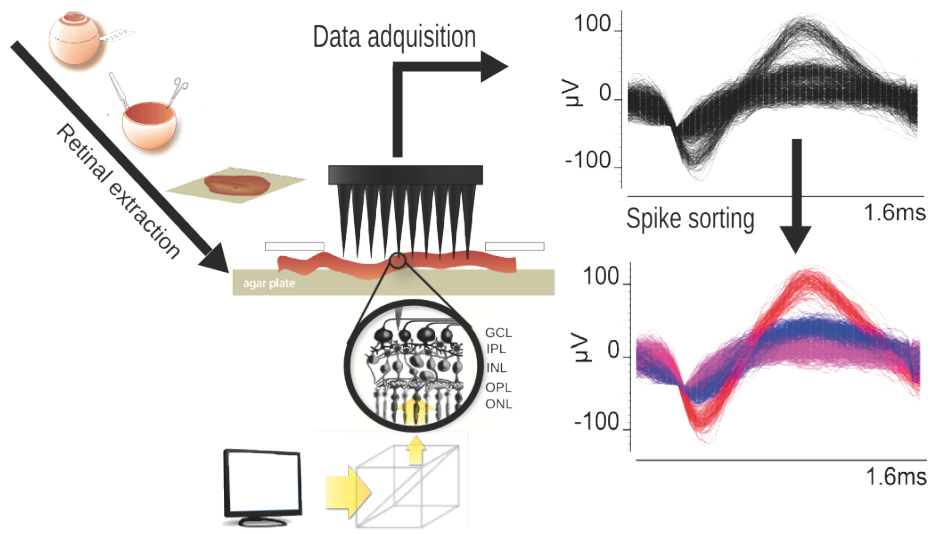


Figura 5.5: Procedimiento de adquisición de datos.

Finalmente, la figura 5.5 resume el procedimiento de adquisición de datos presentado.

Todo el trabajo de laboratorio ha sido realizado por el Grupo de Investigación en Neuroingeniería Biomédica de la UMH [33].

5.3. Campos receptivos: delimitación, localización y cálculo del área

El campo receptivo es aquella región de la retina que al incidir sobre ella un estímulo luminoso produce una respuesta en una determinada fibra de las vías neurales visuales. Por tanto, lo constituyen todas aquellas células de la retina que, al ser estimuladas, influyen en la actividad de dicha célula generando una tasa de disparo bioeléctrico. Es por ello que la retina puede considerarse como un mosaico de campos receptivos superpuestos [47].

5.3.1. Delimitación y localización

Para delimitar y localizar los campos receptivos de las células ganglionares se ha utilizado la técnica descrita en la publicación [48]. Una de las métricas que determinan cuán bueno es un modelo de retina es la correspondencia entre el tamaño y forma de los campos receptivos. Se ha desarrollado un algoritmo en el cual las respuestas a cada par derecha-izquierda e izquierda-derecha del estímulo presentado en el capítulo 5 (figura 5.3) son procesadas de forma separada para calcular sus centroides. Ante posibles señales de ruido, se ha añadido un peso al umbral de disparo con el fin de filtrar posibles respuestas erróneas. Tras esto, las señales se ponen en fase para cancelar los efectos de la latencia y así poder localizar los campos receptivos en su dirección correcta. En la figura 5.6 (publicada en [48]) se muestra cómo distintos estímulos temporales afectan a la frecuencia y distribución de los disparos. Cada uno de los 4 pares forman las matrices M_{0-180} , M_{90-270} , M_{45-225} y $M_{135-315}$. Por último, el campo receptivo se calcula mediante la multiplicación de las matrices, realizando una media ponderada de los resultados.

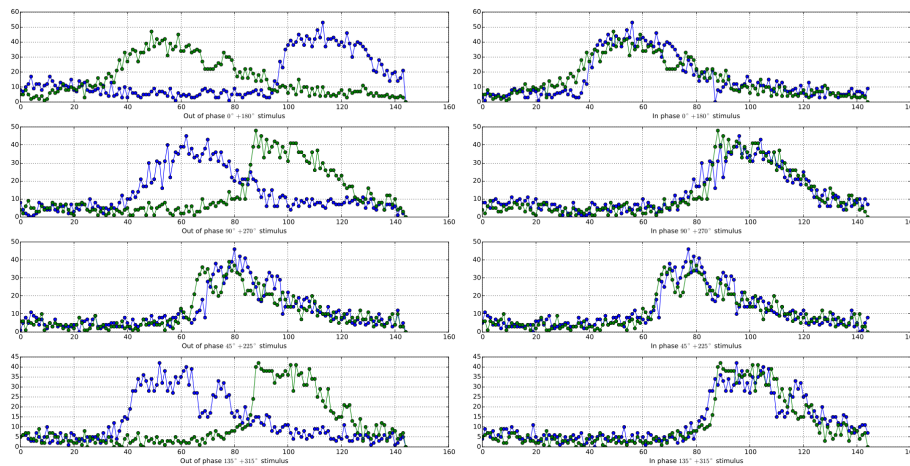


Figura 5.6: Puesta en fase de las respuestas $0^\circ - 180^\circ$, $45^\circ - 225^\circ$, $90^\circ - 270^\circ$ y $135^\circ - 315^\circ$.

La dificultad del proceso de cálculo del área de los campos receptivos de las células ganglionares puede apreciarse en la figura 5.7 (tomada del trabajo [40]). La parte derecha de la figura muestra 11 células ganglionares tipo *ON* y *transient*. La parte izquierda de la figura muestra la puesta en fase y ponderación para la obtención del centroide, a fin de poder situar dicha célula ganglionar y también saber qué forma y qué área tiene.

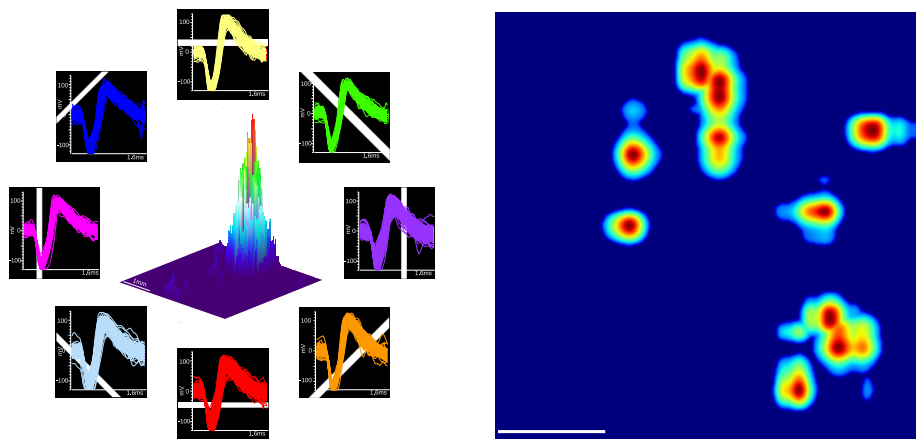


Figura 5.7: Muestra del cálculo del centroide de una célula ganglionar y conjunto de campos receptivos de distintas células ganglionares.

5.3.2. Cálculo del área del campo receptivo

Una vez localizados y delimitados los campos receptivos, hay que calcular su área para establecer una métrica comparativa entre los datos biológicos y los datos sintéticos, mediante el cálculo de la diferencia absoluta del área del campo receptivo. Para ello, se han utilizado técnicas y algoritmos pertenecientes a la rama de conocimiento del aprendizaje automático.

El *aprendizaje automático* es un campo de estudio dentro de las ciencias de la computación que evolucionó del estudio del reconocimiento de patrones y de la teoría de aprendizaje computacional de la inteligencia artificial. El aprendizaje automático abarca el estudio y construcción de algoritmos que sean capaces de aprender y hacer predicciones a partir de un conjunto de datos. En general, un problema de aprendizaje considera un conjunto de n muestras de datos y entonces intenta predecir propiedades de datos desconocidos.

Los problemas de aprendizaje se clasifican en varias categorías:

- **Aprendizaje supervisado.** Proceso de aprendizaje automático de inferencia de una función a partir de datos de entrenamiento etiquetados. El conjunto de muestras provienen con atributos adicionales que se quieren predecir. Por tanto, se desea obtener una función capaz de predecir el valor/categoría correspondiente de una muestra de entrada a partir de los datos de entrenamiento. Este problema puede ser de dos tipos:
 - **Clasificación estadística.** Las muestras pertenecen a dos o más clases y se quiere aprender a partir de los datos ya etiquetados (datos de entrenamiento) cómo predecir la clase de los datos no etiquetados. Un ejemplo de problema de clasificación puede ser el reconocimiento de la escritura manual, en el que el objetivo es asignar cada vector de entrada a una de las categorías finitas existentes. Otra forma de pensar en la clasificación como un problema discreto (en lugar de continuo), donde existen un número de categorías limitadas, es el intento de clasificación de cada una de las n muestras proporcionadas en la categoría correcta.
 - **Análisis de la regresión.** Se trata de un proceso estadístico de estimación de las relaciones entre un conjunto de variables. Si la

salida deseada consiste en una o más variables continuas, entonces la tarea es llamada regresión. Un ejemplo de un problema de regresión puede ser la predicción de la longitud de un pez como función de su edad y peso.

- **Aprendizaje no supervisado.** Los datos de entrenamiento consisten en un conjunto de vectores de entrada x sin ningún tipo de valores objetivo correspondientes. El objetivo de este tipo de problemas puede ser el descubrimiento de grupos con características similares, al cual se denomina *clustering* o *agrupamiento*, o la determinación de la distribución de los datos dentro del espacio de entrada, conocido como *estimación de la densidad*.

Para poder calcular el área de los distintos campos receptivos, es necesario primero poder clasificar cada uno de los campos receptivos, mediante técnicas de *clustering*. El modelo de clasificación más estrechamente relacionado con la estadística se basa en modelos de distribución. Los *clusters* pueden ser definidos como objetos que pertenecen de forma muy probable a la misma distribución. Uno de los métodos más destacados es conocido como *Gaussian mixture models* (GMM). En este método, el conjunto de datos es modelado con un número fijo de distribuciones Gaussianas (para evitar el sobreajuste), que son inicializadas de forma aleatoria, y cuyos parámetros son optimizados de forma iterativa para que se ajusten mejor al conjunto de datos. Este proceso convergerá hacia un óptimo local, por lo que múltiples ejecuciones pueden producir diferentes resultados. En la figura 5.8 se muestra un ejemplo de un *Gaussian mixture model* de dos componentes.

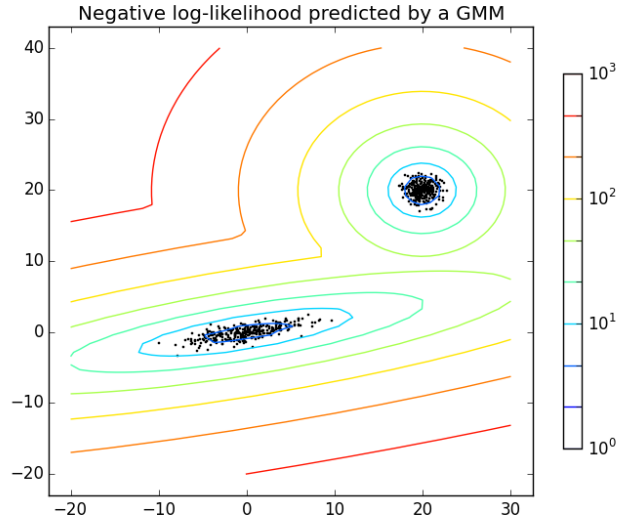


Figura 5.8: Modelo GMM de dos componentes.

Mediante esta técnica se obtiene, dado un conjunto c de campos receptivos, su correspondiente área, a través de la obtención del tamaño de cada uno de los *clusters*. Todo este procedimiento de cálculo se ha realizado mediante la librería *scikit-learn* presentada en la subsección 4.2.4. Esta librería implementa diferentes clases para estimar modelos GMM, correspondientes a diferentes estrategias de estimación.

5.4. Modelado de las células ganglionares *transient*

Como se ha descrito en la sección 2.1, ante un contraste estacionario las células ganglionares tipo *transient* presentan una respuesta impulsiva decreciente, mostrando un estado de saturación. Además, en los primeros ciclos del proceso de integración, presenta una respuesta creciente. Por tanto, puede aproximarse su comportamiento mediante la función Gaussiana, definida por la ecuación 5.3.

$$f(x) = a \cdot e^{-\frac{(x-b)^2}{2 \cdot c^2}} \quad (5.3)$$

El valor de actividad acumulado en cada iteración del modelo neuronal de integración y disparo (descrito en la subsección 2.3.1) es ponderado mediante el cálculo del valor de la función Gaussiana, aproximando de esta forma el comportamiento *transient* de las células ganglionares. En la figura 5.9 se muestra un ejemplo de la respuesta impulsiva que ofrecería una célula ganglionar *transient*, modelada mediante nuestra herramienta, ante un estímulo visual. Se observa que la célula se va saturando y pasado un tiempo deja de emitir impulsos. Cuando deja de ser estimulada, pasa por un proceso de reinicio y cuando vuelve a ser estimulada nuevamente, responde de manera continua al estímulo hasta que vuelve a saturarse una vez más.

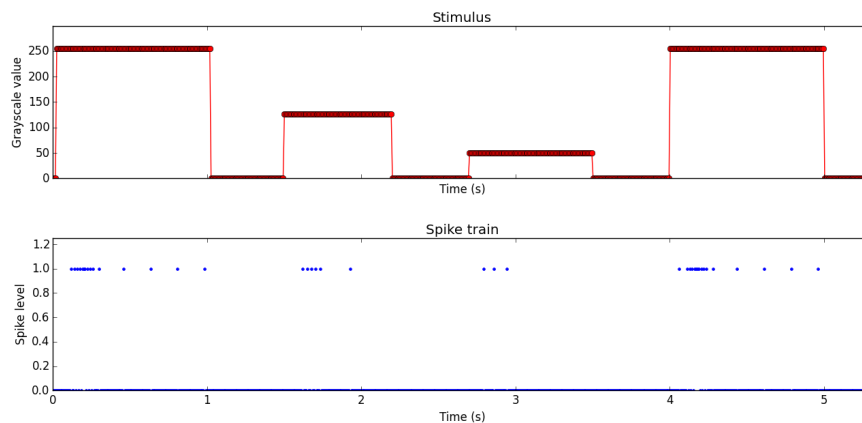


Figura 5.9: Modelado de la respuesta impulsiva de una célula ganglionar *transient*.

El parámetro del cromosoma que modela el comportamiento de las células *transiente*, *FrequencyModulationFactor* (FMF) es precisamente la variable c de la ecuación 5.3, la cual modela el ancho de la función Gaussiana.

5.5. Métricas

A continuación se describen de forma detallada las cuatro métricas que se han seleccionado para la realización de los dos casos de estudio, con la finalidad de poder comparar los registros sintéticos de retina con los registros biológicos obtenidos en el laboratorio. Todas las métricas comparativas han sido validadas en distintos estudios como aptas para la comparación de registros de retina.

5.5.1. Firing rate absolute difference (FRAD)

Dentro del ámbito de la neurociencia, la codificación neuronal es el campo de estudio que trata la representación de la información sensorial de las respuestas neuronales y su relación con la actividad eléctrica. El problema de la codificación neuronal es una de las cuestiones fundamentales en neurociencia. Tradicionalmente se ha tenido la creencia de que mucha de la información relevante está contenida en la tasa de disparo de la neurona. La tasa de disparo o *firing rate* se define generalmente por un promedio temporal. Se establece una ventana temporal T y se cuenta el número de disparos $n_{sp}(T)$ que se producen dentro de dicha ventana temporal. En la figura 5.10 se puede ver de forma gráfica el tren de impulsos y la ventana temporal seleccionada. La división entre la longitud de la ventana temporal proporciona la media de la tasa de disparos, cuyas unidades son s^{-1} o Hz . La ecuación 5.4 presenta la definición formal del cálculo de la tasa de disparo.

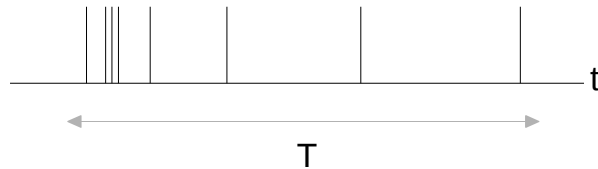


Figura 5.10: Definición de la tasa de disparo media dada una ventana temporal.

$$FR = \frac{n_{sp}(T)}{T} \quad (5.4)$$

Definida esta métrica, para poder comparar la tasa de disparos entre los registros biológicos y sintéticos, se utiliza la diferencia en valor absoluto de ambas tasas de disparo (FRAD), siendo el objetivo la minimización de dicha diferencia: cuanto más próxima a cero sea, mayor precisión habrá entre ambos modelos de retina.

5.5.2. Peri-Stimulus Time Histogram (PSTH) - Kullback-Leibler divergence (KLD)

Cuando se estimula una neurona mediante algún tipo de secuencia de entrada, se almacenan los datos –impulsos eléctricos– para poder ser estudiados posteriormente. Esta secuencia de estimulación se repite varias veces, siguiendo el mismo patrón, y las respuestas neuronales se siguen almacenando para poder construir posteriormente un Peri-Stimulus-Time Histogram (PSTH) con un ancho de *bin* Δt . El tiempo t se mide con respecto al inicio de cada secuencia de estimulación (iteración) y el intervalo Δt define la duración del *bin* para la construcción del histograma, típicamente en unidades del orden de milisegundos. En la figura 5.11 puede verse de forma gráfica la construcción de el PSTH tras tres ejecuciones –o iteraciones– del mismo experimento [49].

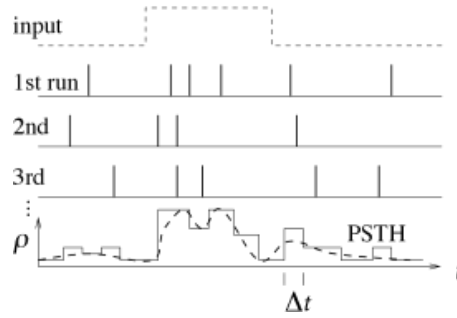


Figura 5.11: Peri-Stimulus-Time Histogram (PSTH) tras varias ejecuciones del experimento.

El número de ocurrencias de disparos $n_K(t; t + \Delta t)$ sumado sobre todas las repeticiones del experimento, dividido entre el número de repeticiones K es la medida de la actividad neuronal típica entre el tiempo t y $t + \Delta t$. Si se divide entre el tamaño del intervalo Δt se obtiene la densidad de la tasa de disparos.

$$\rho(t) = \frac{1}{\Delta t} \cdot \frac{n_K(t; t + \Delta t)}{K} \quad (5.5)$$

Para poder comparar las PSTH obtenidas tras los experimentos biológicos y sintéticos, se ha optado por hacer uso de la Kullback-Leibler divergence como método comparativo entre ambos histogramas. A continuación se detalla este procedimiento de medida de disparidad.

Kullback-Leibler divergence (KLD)

La divergencia de Kullback-Leibler (KLD) mide la distancia entre dos distribuciones de densidad. Esta divergencia es también conocida como información de divergencia y entropía relativa. Para distribuciones de probabilidad discretas P y Q , la divergencia de Kullback-Leibler se define mediante la ecuación 5.6.

$$D_{KL}(P \parallel Q) = \sum_i P(i) \cdot \log \frac{P_i}{Q_i} \quad (5.6)$$

D_{KL} es mayor que cero, no simétrica, cero si las distribuciones coinciden exactamente y puede ser potencialmente igual a infinito. La divergencia de Kullback-Leibler es central en la teoría de la información y en estadística, así como en neurociencia, jugando un papel importante en el campo de estudio del aprendizaje automático. Por estas razones se ha seleccionado la divergencia de KL como el método de comparación entre las distintas métricas que modelan un histograma.

5.5.3. Interspike Interval Histogram (ISI) - Kullback-Leibler divergence (KLD)

La estimación del Interspike Interval Histogram (ISI) de un conjunto de datos experimentales, es un método común de estudio de la variabilidad neuronal ante una cierta entrada estacionaria. En un experimento típico, el tren de impulsos de una neurona se almacena mientras es estimulada por un estímulo constante, y posteriormente se analiza mediante el análisis de la distribución de los intervalos s_k entre dos impulsos consecutivos a través de la creación del histograma. Si el

tren de impulsos es suficientemente largo, el histograma proporciona una buena estimación de la distribución ISI, que se denota como $P_0(s)$ [49]. En la figura 5.12 se muestra una distribución de los intervalos estacionaria.

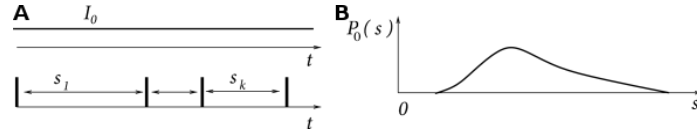


Figura 5.12: **A.** Tren de impulsos con intervalos variables. **B.** Distribución del ISI.

El método de comparación entre el ISI biológico y sintético es el mismo que el que se ha utilizado para comparar el PSTH, la Kullback-Leibler divergence 5.5.2, ya que ambas métricas son histogramas, y por tanto pueden ser analizadas de la misma forma.

5.5.4. Receptive field absolute difference (RFAD)

La última métrica seleccionada es el área del campo receptivo. Se ha descrito en la sección 5.3 tanto el concepto de campo receptivo como el proceso de obtención y cálculo de su área. Para poder comparar el área del campo receptivo entre los datos proporcionados por la retina biológica y la retina sintética, se ha optado por utilizar la diferencia en valor absoluto del área del campo receptivo como medio de comparación entre ambos conjuntos de datos. Al igual que ocurre con la métrica FRAD (subsección 5.5.1), cuanto menor sea el valor de la diferencia en valor absoluto, más correlación habrá entre los campos receptivos.

5.6. Caso de estudio 1: Estudio de la convergencia del método

Antes de proceder con el experimento multiobjetivo, se tiene que comprobar y asegurar la convergencia del método para cada métrica particular. Con este fin, cuatro experimentos monobjetivo, que se corresponden con las cuatro métricas presentadas, han sido diseñados para aproximar los registros electrofisiológicos sintéticos y reales. La tabla 5.2 muestra la correspondencia entre las métricas seleccionadas y los experimentos realizados.

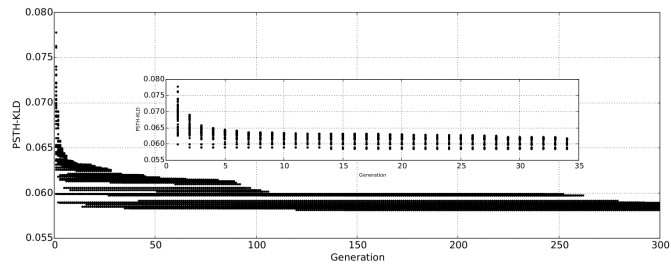
Métrica de Calidad	Experimento
<i>PSTH – KLD</i>	Experimento 1
<i>FRAD</i>	Experimento 2
<i>ISI – KLD</i>	Experimento 3
<i>RFAD</i>	Experimento 4

Tabla 5.2: Etiquetado de las pruebas de convergencia

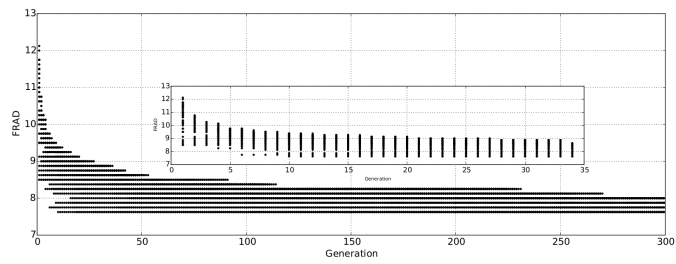
En el proceso de experimentación, antes de concluirse la simulación de las 2500 generaciones de individuos, se extrajeron los datos experimentales cuando el experimento iba por la generación 300. Por ello, se muestran dos figuras en la que la primera de ellas, la figura 5.13, refleja los datos para 300 generaciones, y

la figura 5.14, refleja los datos finales del experimento, tras la simulación de 2500 generaciones. Se observa en las gráficas de la figura 5.13 que todas las métricas de calidad convergen en aproximadamente el mismo número de iteraciones. Se puede observar también que cuando el número de iteraciones supera la cifra de 150, el espacio de búsqueda inicial se reduce hasta llegar a un 20% en cada experimento, lo cual representa una buena tendencia de convergencia dentro de las iteraciones iniciales. Para la métrica RFAD, el mejor resultado se consigue aproximadamente a partir de la iteración 2000, resultado que no podría haberse obtenido si se hubiera detenido la simulación en la iteración 300. Para la métrica FRAD, el mejor resultado se consigue aproximadamente a partir de la iteración 600. Para el resto de métricas, PSTH-KLD e ISI-KLD, no hacen falta más de 300 iteraciones para conseguir el mejor resultado. En todas las gráficas se observa una manifiesta convergencia para cada métrica y por tanto, se pueden dar por validados los procedimientos realizados para la experimentación. En aras de observar con una mayor claridad el proceso de convergencia, se ha adjuntado a cada figura una ventana ampliada utilizando un reducido margen de iteraciones.

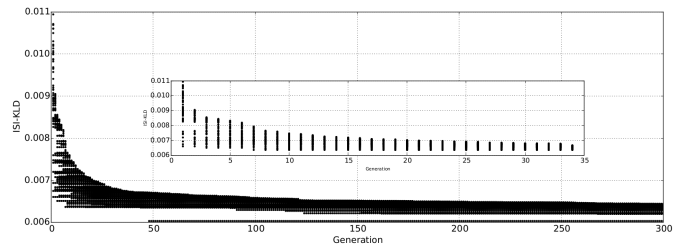
5.6. Caso de estudio 1: Estudio de la convergencia del método



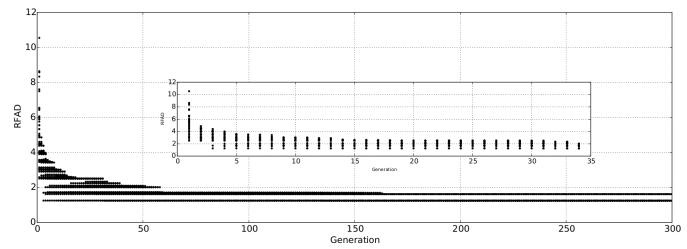
(a) Test de convergencia PSTH-KLD para 300 generaciones (experimento 1).



(b) Test de convergencia FRAD para 300 generaciones (experimento 2).

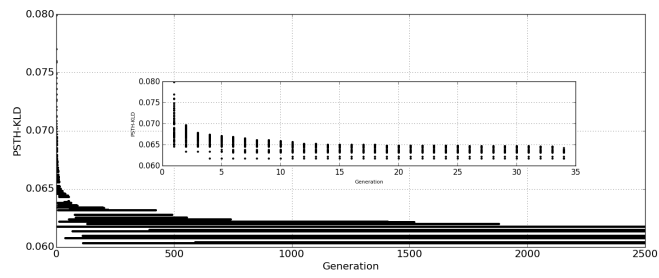


(c) Test de convergencia ISI-KLD para 300 generaciones (experimento 3).

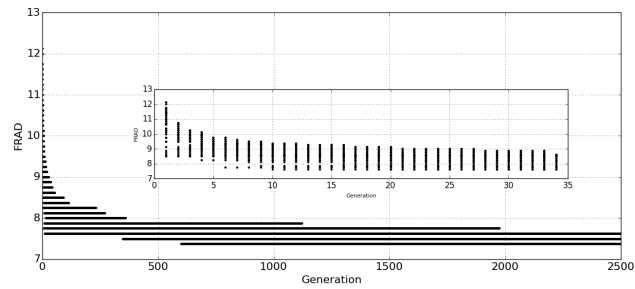


(d) Test de convergencia RFAD para 300 generaciones (experimento 4).

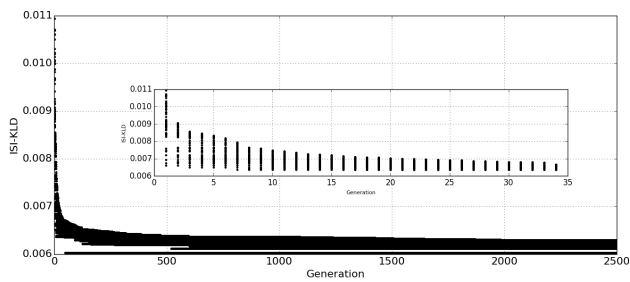
Figura 5.13: Test de convergencia para 300 generaciones de las métricas PSTH-KLD (a), FRAD (b), ISI-KLD (c) y RFAD (d) en el proceso de comparación de los registros sintéticos y biológicos.



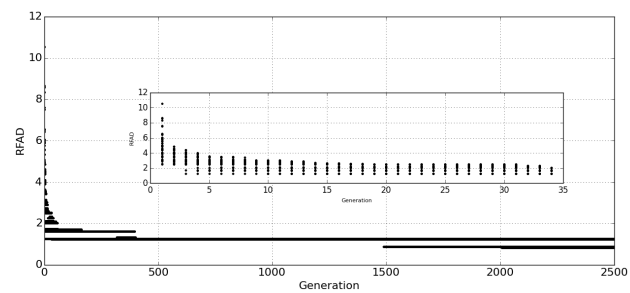
(a) Test de convergencia PSTH-KLD para 2500 generaciones (experimento 1).



(b) Test de convergencia FRAD para 2500 generaciones (experimento 2).



(c) Test de convergencia ISI-KLD para 2500 generaciones (experimento 3).



(d) Test de convergencia RFAD para 2500 generaciones (experimento 4).

Figura 5.14: Test de convergencia para 2500 generaciones de las métricas PSTH-KLD (a), FRAD (b), ISI-KLD (c) y RFAD (d) en el proceso de comparación de los registros sintéticos y biológicos.

5.7. Caso de estudio 2: Ajuste paramétrico multiobjetivo de un modelo de retina

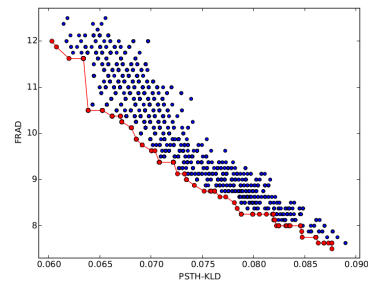
Para evaluar la efectividad de la propuesta MOOGA, se ha diseñado un experimento multiobjetivo cuya finalidad es encontrar un conjunto de modelos de retina que optimicen, al mismo tiempo, una serie de criterios de interés que han sido previamente definidos en este capítulo. Para tal fin, las cuatro métricas estudiadas, y resumidas en la tabla 5.2 de la subsección 5.6, han sido seleccionadas como métricas de calidad en un esfuerzo para ajustar y mejorar el modelo de retina propuesto en esta sección. En este experimento, se ha adoptado como criterio la minimización de la diferencia absoluta entre los valores sintéticos y biológicos de cada una de las métricas, con el fin de aproximar los registros electrofisiológicos. De igual forma que en el caso de estudio presentado en la sección 5.6, se muestran las gráficas del experimento tanto para 300 generaciones (figura 5.15) como para 2500 generaciones (figura 5.16). Como resultado, el conjunto de figuras 5.15 y 5.16 muestran los distintos resultados de la minimización de los valores, al mismo tiempo, para todas las métricas. Los datos se muestran en gráficos mediante el enfrentamiento por pares de cada una de las métricas. Aquellos individuos que pertenecen al frente de Pareto, y consecuentemente representan una solución no dominada válida para el problema, están identificados por medio de puntos de color rojo.

Si se comparan los resultados para 300 y 2500 generaciones, se observa que las nubes de puntos presentan la misma forma, por tanto, a partir de la generación 300 se sigue la misma tendencia. Esto quiere decir que es un buen número de generaciones para poder predecir el aspecto que tendrá el resultado final. Por otro lado, aunque presenten la misma forma las nubes de puntos, tanto para 300 generaciones como para 2500 generaciones, el número de individuos que forman el frente de Pareto es distinto. Por tanto, no pueden extraerse conclusiones hasta la finalización del experimento, ya que muy posiblemente un individuo que esté presente en el frente de Pareto en la generación 300, no lo esté en la generación 2500.

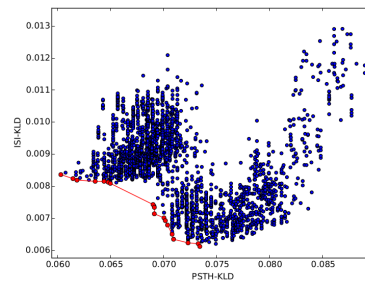
En las figuras 5.16(c), 5.16(d), 5.16(e) y 5.16(f), el frente de Pareto muestra una forma convexa. Por otro lado, la figura 5.16(b) presenta un frente de Pareto con una forma no convexa. En los casos en los que el frente de Pareto presenta una forma convexa, es más fácil encontrar un punto de equilibrio entre todas las posibles soluciones del problema, es decir, el mejor modelo de retina. Es en estos casos donde los expertos en neurología, atendiendo a las necesidades de cada individuo, podrían llegar a un acuerdo para la elección del mejor modelo, a pesar del carácter multiobjetivo del problema.

La figura 5.16(a) muestra una nube de puntos formada por un conjunto de soluciones obtenidas al comparar las métricas *PSTH-KLD* y *FRAD*, las cuales tienen una forma pseudo-lineal. En la figura 5.16(b) se muestra la comparación entre las métricas *PSTH-KLD* and *ISI-KLD*. Existen dos grupos (dos *clusters*) claramente diferenciados. Esta situación es de interés ya que muestra que existe algún gen que, dependiendo de su valor, clasifica a los individuos de la población en dos *clusters* diferenciados. El frente de Pareto, en este caso, está compuesto por aquellos individuos que forman los dos *clusters*, lo que significa que ambas nubes de puntos exhiben parte de los mejores modelos de retina y por tanto, no es posible descartar ninguno de ellos. En la figura 5.16(d) se comparan las métricas *FRAD* y *ISI-KLD*. De forma similar a la figura anterior (figura 5.16(b)), en ella existen dos grupos diferenciados, siendo en este caso mucho más compactos.

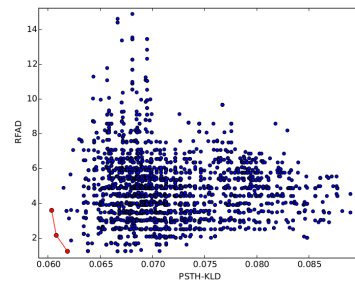
Las figuras 5.16(c), 5.16(e) y 5.16(f) muestran similares patrones y por ello, una misma tendencia, con pequeños *clusters* diferenciados y frentes de Pareto.



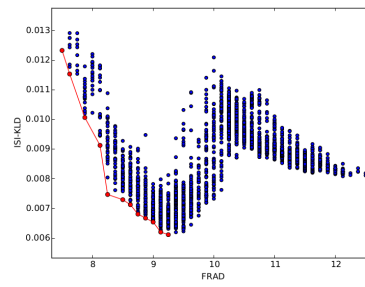
(a) PSTH-KLD contra FRAD.



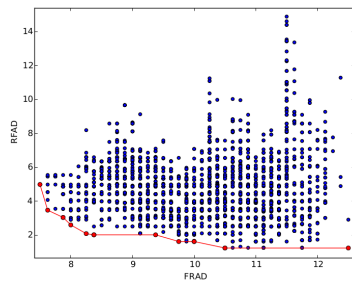
(b) PSTH-KLD contra ISI-KLD.



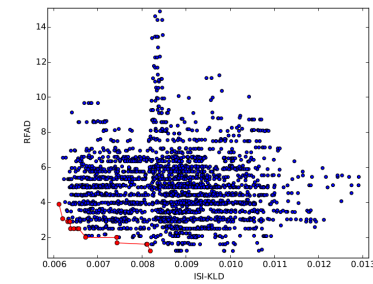
(c) PSTH-KLD contra RFAD.



(d) FRAD contra ISI-KLD.

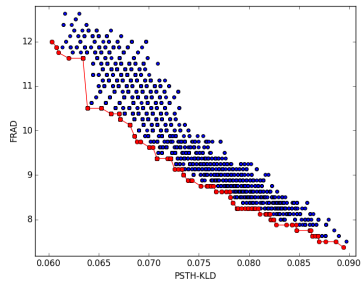


(e) FRAD contra RFAD.

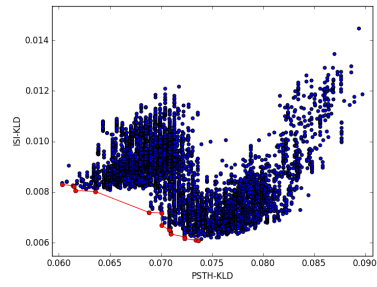


(f) ISI-KLD contra RFAD.

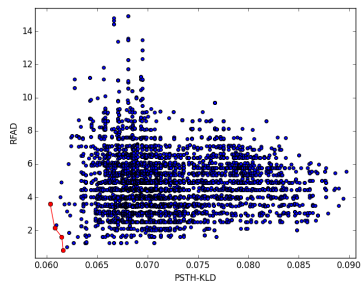
Figura 5.15: Gráficos multiobjetivo que enfrentan a cada par de métricas (300 generaciones).



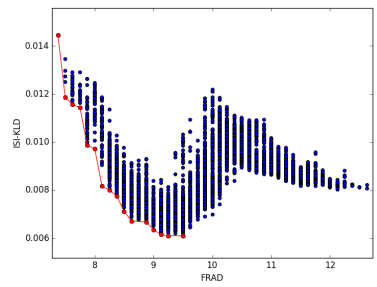
(a) PSTH-KLD contra FRAD.



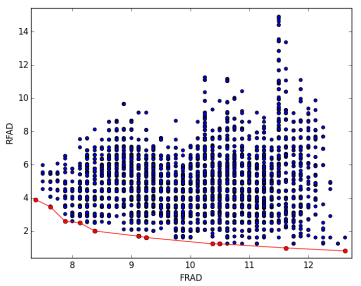
(b) PSTH-KLD contra ISI-KLD.



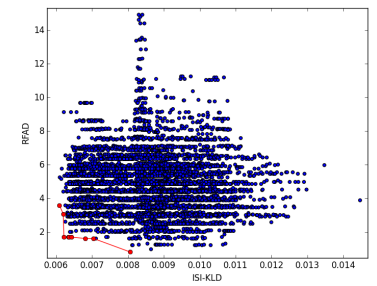
(c) PSTH-KLD contra RFAD.



(d) FRAD contra ISI-KLD.



(e) FRAD contra RFAD.



(f) ISI-KLD contra RFAD.

Figura 5.16: Gráficos multiobjetivo que enfrentan a cada par de métricas (2500 generaciones).

La forma de los *clusters* no está tan definida y pronunciada como ocurre en las figuras 5.16(b) y 5.16(d).

En relación a trabajos futuros, en las figuras 5.16(b) y 5.16(d), se observan varias agregaciones o acumulaciones de puntos que corresponden posiblemente a una misma mutación en algún gen. La comprensión pormenorizada de cuáles son los genes que más afectan a la variabilidad de los resultados en las gráficas de Pareto constituye una tarea de investigación que nos ayudará a acotar el tiempo de convergencia de nuestro algoritmo. La forma lineal de la nube de puntos representada en las gráficas de PSTH-KLD contra FRAD (figuras 5.15(a) y 5.16(a)), sugiere que existe una leve dependencia lineal entre estos dos objetivos. Como trabajo futuro se propone ampliar los rangos de los parámetros del modelo y repetir este experimento para desmentir o no esta suposición. Por otra parte, las gráficas 5.16(c), 5.16(e) y 5.16(f) manifiestan una total independencia de los objetivos, llegando a ser contrapuestos, si nos fijamos en la forma indefinida de las respectivas nubes de puntos.

Estos resultados muestran que la estrategia MOOGA reduce el espacio de búsqueda y ofrece un conjunto de modelos de retina óptimos, siguiendo una amplia serie de objetivos al mismo tiempo, y por tanto, el proceso de computación puede ser completado en una razonable cantidad de tiempo. Vistos los tiempos de cálculo, el análisis por fuerza bruta es prácticamente imposible de llevar a cabo, y más si se pretende mejorar el modelo en un futuro con la codificación de más parámetros genéticos.

Capítulo 6

Conclusión

Tras haber dado una perspectiva del funcionamiento y estructura de la retina humana, se ha procedido a realizar un estado del arte de los distintos modelos de retina que han sido publicados por la comunidad científica. Se ha realizado una división entre modelos de bajo y alto nivel, atendiendo a la naturaleza de su funcionamiento. Se ha propuesto en base a los textos científicos publicados, un modelo de retina de alto nivel, ya que el objetivo final del trabajo científico es llegar a producir un modelo de retina que sea fácilmente optimizable, que pueda correr en tiempo real –produciendo la salida del sistema en intervalos de tiempo biocompatibles– y que permita una implementación portable.

También se han propuesto las métricas que permiten evaluar y comparar el modelo de retina sintético con el biológico, así como la forma en la que evaluar dichas métricas y poder argumentar si un modelo simulado de retina es más fidedigno o menos, dependiendo de cuánto se aproximen sus resultados –registros sintéticos– a los obtenidos en el laboratorio de las retinas biológicas –registros biológicos–.

Debido a la naturaleza del problema, se ha presentado una estrategia automática multiobjetivo para el modelado de retinas sintéticas. El ajuste se realiza mediante la selección de aquellos parámetros que mejor aproximan, de forma automática, un modelo de retina sintético, con los registros electrofisiológicos obtenidos en el laboratorio. Se han descrito los genes que componen cada cromosoma, así como el rango de valores entre los que se mueven. De la misma manera, se ha diseñado un simulador para modelos de retina escalable, con bajo acoplamiento y alta cohesión, capaz de trabajar con los distintos modelos de retina publicados en revistas científicas. Gracias a esto, la herramienta creada puede concluirse que es muy potente, ya que bastaría con realizar la codificación de cualquier modelo de alto nivel, e implementar las interfaces necesarias, para que el motor de simulación pudiera evaluar la calidad del modelo a través de sus métricas. Igualmente, la herramienta podría comparar la calidad de distintos modelos de retina, siempre que su codificación se integrara con el motor de simulación.

Los resultados muestran que esta estrategia de búsqueda es fructífera cuando trata de resolver problemas de optimización con múltiples criterios de satisfacción. Asimismo, las métricas de calidad convergen en una cantidad de tiempo razonable, lo que sugiere que esta estrategia, basada en el algoritmo de optimización multiobjetivo NSGA-II, es útil para el ajuste paramétrico de modelos neuronales. Además, la estrategia multiobjetivo, revela información valiosa cuando los criterios propuestos se enfrentan entre sí. En los gráficos presentados en el primer caso de estudio (sección 5.6), se observa que únicamente con

300 generaciones no es posible obtener los mejores resultados para las métricas propuestas, y por tanto, hay que esperar a la conclusión del experimento para obtener los resultados óptimos. En el segundo caso de estudio (sección 5.7), se presentan las distintas nubes de puntos, con las soluciones obtenidas, comparando cada par de métricas, obteniéndose 6 gráficos distintos. En varios de ellos se observa que el frente de Pareto presenta una forma convexa, situación que es más favorable para poder encontrar un punto de equilibrio entre todas las posibles soluciones del problema, es decir, el mejor modelo de retina. De igual forma que para el primer caso de estudio, se presentan los gráficos para una simulación de 300 generaciones y 2500 generaciones, donde se observa que las nubes de puntos prácticamente tienen la misma forma (con mayor densidad de puntos para 2500 generaciones), pero el frente de Pareto es distinto. Esto quiere decir que, las mejores soluciones al problema no son las mismas para el resultado de simulación obtenido tras 300 generaciones que para 2500 generaciones.

Finalmente, puede concluirse que los experimentos demuestran que la propuesta de estrategia presentada en el trabajo, para el modelado de retinas, funciona correctamente, ya que ofrece resultados con un nivel de optimización, en un tiempo de cálculo y simulación aceptable.

Bibliografía

- [1] “International work-conference on the interplay between natural and artificial computation.” <http://www.iwinac.uned.es/>.
- [2] R. Crespo-Cano, A. Martínez-Álvarez, A. Díaz-Tahoces, S. Cuenca-Asensi, J. Ferrández, and E. Fernández, “On the automatic tuning of a retina model by using a multi-objective optimization genetic algorithm,” in *Artificial Computation in Biology and Medicine* (J. M. Ferrández Vicente, J. R. Álvarez Sánchez, F. de la Paz López, F. J. Toledo-Moreo, and H. Adeli, eds.), vol. 9107 of *Lecture Notes in Computer Science*, pp. 108–118, Springer International Publishing, 2015.
- [3] S. Ghosh-Dastidar and H. Adeli, “Spiking neural networks,” *International Journal of Neural Systems*, vol. 19, no. 4, pp. 295–308, 2009. cited By 0.
- [4] E. Fernandez, F. Pelayo, S. Romero, M. Bongard, C. Marin, A. Alfaro, and L. Merabet, “Development of a Cortical Visual Neuroprosthesis for the Blind: the Relevance of Neuroplasticity,” *Journal of Neural Engineering*, vol. 2, no. 4, p. R1, 2005.
- [5] R. A. Normann, B. A. Greger, P. House, S. F. Romero, F. Pelayo, and E. Fernandez, “Toward the Development of a Cortically Based Visual Neuroprosthesis,” *Journal of Neural Engineering*, vol. 6, no. 3, p. 035001, 2009.
- [6] H. Ikeda and M. J. Wright, “Receptive field organization of ‘sustained’ and ‘transient’ retinal ganglion cells which subserve different functional roles,” *The Journal of Physiology*, vol. 227, no. 3, pp. 769–800, 1972.
- [7] N. R. Kolb H, Fernandez E, *Webvision: The Organization of the Retina and Visual System*. University of Utah Health Sciences Center, 1995.
- [8] H. Plesser, M. Diesmann, M.-O. Gewaltig, and A. Morrison, “NEST Initiative.” <http://www.nest-initiative.org>, 2012.
- [9] H. Plesser, M. Diesmann, M.-O. Gewaltig, and A. Morrison, “Nest: the neural simulation tool,” in *Encyclopedia of Computational Neuroscience* (D. Jaeger and R. Jung, eds.), pp. 1849–1852, Springer New York, 2015.
- [10] M. L. Hines and N. T. Carnevale, “The NEURON simulation environment.,” *Neural Comput*, vol. 9, pp. 1179–209, Aug 1997.
- [11] N. Carnevale and S. Rosenthal, “Kinetics of diffusion in a spherical cell. i. no solute buffering,” *Journal of Neuroscience Methods*, vol. 41, no. 3, pp. 205 – 216, 1992.
- [12] D. Goodman and R. Brette, “Brian spiking neural network simulator,” in *Encyclopedia of Computational Neuroscience* (D. Jaeger and R. Jung, eds.), pp. 452–455, Springer New York, 2015.

-
- [13] J. Bower, H. Cornelis, and D. Beeman, “Genesis, the general neural simulation system,” in *Encyclopedia of Computational Neuroscience* (D. Jaeger and R. Jung, eds.), pp. 1287–1293, Springer New York, 2015.
- [14] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision*. Kluwer Academic Publishers, 1994.
- [15] B. I. Jordan T. Berry M., *Anticipation of moving stimuli by the retina*. Nature (Lond), 398:334338, 1999.
- [16] M. J. Schnitzer and M. Meister., *Multineuronal firing patterns in the signal from eye to brain*. Neuron, 37:499–511, 2003.
- [17] S. A. B. y Markus Meister Bence P. Olveczky., *Segregation of object and background motion in the retina*. Nature, 423:401–408, 2003.
- [18] M. y M. J. Berry Meister, *The neural code of the retina*. Neuron, 22:435–450, 1999.
- [19] L. I. y Christof Koch, *Computational modelling of visual attention*. Nature Reviews Neuroscience, 2:194–203, 2001.
- [20] “Gipsa-lab is a CNRS research unit joint with Grenoble-INP (Grenoble Institute of Technology), Université Joseph Fourier and Université Stendhal.” <http://www.gipsa-lab.fr>.
- [21] A. Benoit, A. Caplier, B. Durette, and J. Herault, “Using human visual system modeling for bio-inspired low level image processing,” *Comput. Vis. Image Underst.*, vol. 114, pp. 758–773, July 2010.
- [22] A. Wohrer and P. Kornprobst, “Virtual retina: A biological retina model and simulator, with contrast gain control,” *Journal of Computational Neuroscience*, vol. 26, no. 2, pp. 219–249, 2009.
- [23] A. Wohrer, “The vertebrate retina: a functional review,” Research Report 6532, INRIA, 2008.
- [24] A. Ortiz-Rosario and H. Adeli, “Brain-computer interface technologies: From signal to action,” *Reviews in the Neurosciences*, vol. 24, no. 5, pp. 537–552, 2013. cited By 0.
- [25] C. Morillas, S. Romero, A. Martínez, F. Pelayo, L. Reyneri, M. Bongard, and E. Fernández, “A neuroengineering suite of computational tools for visual prostheses,” *Neurocomputing*, vol. 70, no. 16–18, pp. 2817 – 2827, 2007.
- [26] C. Morillas, S. Romero, A. Martinez, F. Pelayo, and E. Fernández, “A computational tool to test neuromorphic encoding schemes for visual neuroprostheses,” in *Computational Intelligence and Bioinspired Systems* (J. Cabestany, A. Prieto, and F. Sandoval, eds.), vol. 3512 of *Lecture Notes in Computer Science*, pp. 510–517, Springer Berlin Heidelberg, 2005.
- [27] A. Martínez-Álvarez, A. Olmedo-Payá, S. Cuenca-Asensi, J. M. Ferrández, and E. Fernández, “Retinastudio: A bioinspired framework to encode visual information,” *Neurocomputing*, vol. 114, no. 0, pp. 45 – 53, 2013.
- [28] N. Siddique and H. Adeli, *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*. John Wiley and Sons, 2013. cited By 25.

- [29] A. Burkitt, “A review of the integrate-and-fire neuron model: I. homogeneous synaptic input,” *Biological Cybernetics*, vol. 95, no. 1, pp. 1–19, 2006.
- [30] A. Martínez-Álvarez, S. Cuenca-Asensi, A. Ortiz, J. Calvo-Zaragoza, and L. A. V. Tejuelo, “Tuning compilations by multi-objective optimization: Application to apache web server,” *Applied Soft Computing*, vol. 29, no. 0, pp. 461 – 470, 2015.
- [31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, apr 2002.
- [32] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization,” in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems* (K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, eds.), (Athens, Greece), pp. 95–100, International Center for Numerical Methods in Engineering, 2001.
- [33] “Grupo de investigación en Neuroingeniería Biomédica de la UMH.” <http://nbio-blog.zeus-desarrollo.umh.es/>.
- [34] “Machine Learning in Python: scikit-learn.” [http://http://scikit-learn.org/stable/index.html/](http://scikit-learn.org/stable/index.html/).
- [35] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, “Deap: Evolutionary algorithms made easy,” *J. Mach. Learn. Res.*, vol. 13, pp. 2171–2175, July 2012.
- [36] Y. Hold-Geoffroy, O. Gagnon, and M. Parizeau, “Once you scoop, no need to fork,” in *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, p. 60, ACM, 2014.
- [37] A. D. Straw, “Vision egg: an open-source library for realtime visual stimulus generation,” *Frontiers in Neuroinformatics*, vol. 2, no. 4, 2008.
- [38] “Cluster de computación del instituto universitario de ingeniería informática.” <http://web.ua.es/cluster-iuii/>.
- [39] M. Van Wyk, H. Wässle, and W. R. Taylor, “Receptive field properties of on- and off-ganglion cells in the mouse retina,” *Visual Neuroscience*, vol. 26, pp. 297–308, 5 2009.
- [40] A. Díaz-Tahoces, A. Martínez-Álvarez, A. García-Moll, L. Humphreys, J. Bolea, and E. Fernández, “Towards the Reconstruction of Moving Images by Populations of Retinal Ganglion Cells,” in *Artificial Computation in Biology and Medicine* (J. M. Ferrández Vicente, J. R. Álvarez Sánchez, F. de la Paz López, F. J. Toledo-Moreo, and H. Adeli, eds.), vol. 9107 of *Lecture Notes in Computer Science*, pp. 220–227, Springer International Publishing, 2015.
- [41] F. J. Pelayo, S. Romero, C. A. Morillas, A. Martínez, E. Ros, and E. Fernández, “Translating image sequences into spike patterns for cortical neurostimulation,” *Neurocomputing*, vol. 58-60, no. 0, pp. 885–892, 2004.
- [42] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

-
- [43] E. Fernández, J.-M. Ferrández, J. Ammermüller, and R. A. Normann, “Population coding in spike trains of simultaneously recorded retinal ganglion cells1,” *Brain Research*, vol. 887, no. 1, pp. 222 – 229, 2000.
- [44] R. A. Normann, D. J. Warren, J. Ammermuller, E. Fernandez, and S. Guillory, “High-resolution spatio-temporal mapping of visual pathways using multi-electrode arrays,” *Vision Research*, vol. 41, no. 10–11, pp. 1261 – 1275, 2001.
- [45] M. Bongard, D. Micol, and E. Fernández, “Nev2lkit: A new open source tool for handling neuronal event files from multi-electrode recordings,” *International Journal of Neural Systems*, vol. 24, no. 04, p. 1450009, 2014. PMID: 24694167.
- [46] G. J. Ortega, M. Bongard, E. Louis, and E. Fernández, “Conditioned spikes: a simple and fast method to represent rates and temporal patterns in multielectrode recordings,” *Journal of Neuroscience Methods*, vol. 133, no. 1–2, pp. 135 – 141, 2004.
- [47] “Psicología de la percepción visual: Campos receptivos de células ganglionares y del N.G.L.” http://www.ub.edu/pa1/node/campo_receptivo/.
- [48] A. Díaz-Tahoces, A. Martínez-Álvarez, A. García-Moll, L. Humphreys, J. Bolea, and E. Fernández, “Towards the reconstruction of moving images by populations of retinal ganglion cells,” in *Artificial Computation in Biology and Medicine*, vol. 9107 of *Lecture Notes in Computer Science*, pp. 220–227, Springer International Publishing, 2015.
- [49] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

Capítulo 7

Anexos

7.1. Anexo 1: Clase MoogaRetina.py

```
# -*- coding: utf-8 -*-
"""
@author: rcrespocano
"""
import random
import numpy
import os

from MoogaConstants import MoogaConstants
from Utils import Utils
from deap import base, creator, tools

class MoogaRetina():
    def __init__(self, retina_model_handler, weights=(-1.0,)):
        self.retina_model_handler = retina_model_handler
        self.mooga_constants = MoogaConstants()
        self.__create_types(weights)
        self.__create_toolbox()
        self.__create_genetic_operators()
        self.log_path = self.mooga_constants.get_iaf_log_directory_path()

    def __create_types(self, weights):
        """ Type Creation: First step with DEAP is to create the
        required
        """
        types.
        """
        creator.create("FitnessMin", base.Fitness, weights=weights)
        creator.create("Individual", self.retina_model_handler.get_retina_model_base_class(), fitness=creator.FitnessMin)

    def __create_toolbox(self):
        """ The toolbox is intended to store functions with their
        arguments
        """
        under standard aliases for uses in algorithms.
        """
        self.toolbox = base.Toolbox()
        self.toolbox.register("retina_model", self.retina_model_handler.create_retina_model)
        self.toolbox.register("individual", tools.initRepeat, creator.Individual, self.toolbox.retina_model, n=1)
        self.toolbox.register("population", tools.initRepeat, list, self.toolbox.individual)
```

```

def __create_genetic_operators(self):
    """ Registering the operators and their default arguments
    in the toolbox is done as follow.
    """
    self.toolbox.register("evaluate", self.retina_model_handler
        .evaluate)
    self.toolbox.register("mate", self.retina_model_handler.
        crossover)
    self.toolbox.register("mutate", self.retina_model_handler.
        mutate)
    self.toolbox.register("select", tools.selNSGA2)

def __create_stats(self):
    stats = tools.Statistics(lambda ind: ind.fitness.values)
    stats.register("avg", numpy.mean, axis=0)
    stats.register("std", numpy.std, axis=0)
    stats.register("min", numpy.min, axis=0)
    stats.register("max", numpy.max, axis=0)
    return stats

def __create_log_book(self):
    logbook = tools.Logbook()
    logbook.header = "gen", "evals", "std", "min", "avg", "max"
    return logbook

def start_evolutionary_strategy(self, verbose=None):
    """ Evolving the Population. generating a population, and
    giving it to the algorithm for it to evolve a solution.
    """
    NGEN = self.mooga_constants.get_number_of_generations()
    MU = self.mooga_constants.get_total_population()
    CXPB = self.mooga_constants.get_crossing_probability()
    MUTPB = self.mooga_constants.get_mutation_probability()

    # Create population
    pop = self.toolbox.population(n=MU)

    # Hall of fame
    hall_of_fame = tools.HallOfFame(1)

    # Create stats and logbook
    stats = self.__create_stats()
    logbook = self.__create_log_book()

    # Evaluate the individuals with an invalid fitness
    invalid_ind = [ind for ind in pop if not ind.fitness.valid]
    fitnesses = self.toolbox.map(self.toolbox.evaluate,
        invalid_ind)
    for ind, fit in zip(invalid_ind, fitnesses):
        ind.fitness.values = fit

    # Start hall of fame
    hall_of_fame.update(pop)

    # This is just to assign the crowding distance to the
    individuals
    # no actual selection is done
    pop = self.toolbox.select(pop, len(pop))

    record = stats.compile(pop)

```

```
logbook.record(gen=0, evals=len(invalid_ind), **record)
if verbose:
    print(logbook.stream)
    self.__save_generational_data_to_file(logbook, pop,
hall_of_fame)

# Begin the generational process
for gen in range(1, NGEN):
    # Vary the population
    offspring = tools.selTournamentDCD(pop, len(pop))
    offspring = [self.toolbox.clone(ind) for ind in
offspring]

    for ind1, ind2 in zip(offspring[::2], offspring[1::2]):
        if random.random() < CXPB:
            self.toolbox.mate(ind1, ind2)
            del ind1.fitness.values
            del ind2.fitness.values

    for mutant in offspring:
        if random.random() < MUTPB:
            self.toolbox.mutate(mutant)
            del mutant.fitness.values

    # Evaluate the individuals with an invalid fitness
    invalid_ind = [ind for ind in offspring if not ind.
fitness.valid]
    fitnesses = self.toolbox.map(self.toolbox.evaluate,
invalid_ind)
    for ind, fit in zip(invalid_ind, fitnesses):
        ind.fitness.values = fit

    # Update the hall of fame with the generated individuals
    hall_of_fame.update(offspring)

    # Select the next generation population
    pop = self.toolbox.select(pop + offspring, MU)

    # Append the current generation statistics to the logbook
    record = stats.compile(pop)
    logbook.record(gen=gen, evals=len(invalid_ind), **
record)
    if verbose:
        print(logbook.stream)
        self.__save_generational_data_to_file(logbook, pop,
hall_of_fame)

    return pop, logbook, hall_of_fame

def __save_generational_data_to_file(self, logbook, pop,
hall_of_fame):
    _current_generation = (logbook.select("gen")[-1])
    log_file = self.log_path + "gen_" + str(_current_generation
) + ".log"

    _evals = (logbook.select("evals")[-1])
    _std = (logbook.select("std")[-1])
    _min = (logbook.select("min")[-1])
    _avg = (logbook.select("avg")[-1])
    _max = (logbook.select("max")[-1])

# Fitnesses of all individuals of the population
fitnesses = ""
for i in range(len(pop)):
```



```
        fitnesses += str(pop[i].fitness.values) + ";"

# Best individual
best_individual = "BEST [" + str(hall_of_fame[0]) + "]"

if not os.path.exists(self.log_path):
    os.makedirs(self.log_path)

with open(log_file, "w") as text_file:
    text_file.write(log_data + "\n")
    text_file.write(fitnesses + "\n")
    text_file.write(best_individual + "\n")
```