

**ANÁLISIS NO INVASIVO DE FLUJOS EN
REACTORES QUÍMICOS INDUSTRIALES
MEDIANTE APRENDIZAJE PROFUNDO**

TESIS

QUE PARA OBTENER EL GRADO
ACADÉMICO DE

**MAESTRO EN CIENCIA Y TECNOLOGÍA EN
LA ESPECIALIDAD DE
MECATRÓNICA**

PRESENTA

ING. DANIEL ANTONIO GUTIÉRREZ JIMÉNEZ

DIRECTOR DE TESIS

DR. SEBASTIÁN SALAZAR COLORES

CO-DIRECTOR

DR. FERNANDO FELIPE RIVERA ITURBE

LEÓN, GUANAJUATO, NOVIEMBRE, 2024.



ANÁLISIS NO INVASIVO DE FLUJOS EN REACTORES QUÍMICOS INDUSTRIALES MEDIANTE APRENDIZAJE PROFUNDO

Agradecimientos

Al finalizar esta tesis, quiero tomar un momento para reconocer y agradecer a quienes me han acompañado y apoyado en este camino.

En primer lugar, agradezco profundamente a mi asesor, el Dr. Sebastián Salazar Colores y al Dr. Fernando Felipe Rivera Iturbe, por su paciencia, guía y apoyo incondicional durante todo el proceso. Sus valiosas observaciones, consejos y motivación constante fueron esenciales para la realización de esta tesis.

Agradezco al Centro de Investigaciones en Óptica (CIO), a mis profesores, colegas, directivos y al personal administrativo, por su conocimiento, disposición y apoyo, así como el acceso a sus instalaciones, contribuyeron de manera significativa a mi formación académica y al desarrollo de este proyecto. También extendo mi gratitud al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) y al Posgrado Interinstitucional en Ciencia y Tecnología (PICYT) por el financiamiento y los recursos brindados, que fueron fundamentales para la realización de esta tesis y para alcanzar mis objetivos académicos.

Quiero dar un especial agradecimiento a mi pareja, Erika Jannin Córdova Ramírez, quien desde el inicio de esta idea me alentó y creyó en mí. Siempre estuvo a mi lado en los buenos momentos, pero sobre todo en los malos, brindándome su apoyo incondicional,

palabras de aliento y amor. Su paciencia, comprensión y fortaleza fueron un pilar fundamental para que pudiera superar los desafíos que surgieron a lo largo de este camino. Este logro no habría sido posible sin su compañía y confianza constante, te amo.

A mi familia, gracias por su apoyo incondicional, incluso en la distancia y a pesar de los desafíos y carencias. Aunque no siempre estén físicamente a mi lado, sé que siempre me desean lo mejor. Desde pequeño, me alentaron y me enseñaron que el trabajo constante es el camino al éxito y que, a pesar de las dificultades, siempre hay que encontrar motivos para sonreír, recordándome que no hay mal, que por bien no venga. Su ejemplo y enseñanzas han sido fundamentales para llegar hasta aquí.

Finalmente, quiero compartir una frase que ha guiado mi camino: "Puedes sobre mí dar opinión sesgada o criticar mi oficio que no es porvenir; que alimento la hoguera de la imaginación. Hasta que el cuerpo aguante, seguiré viviendo tal como soy."

Documentos de conferencias

- A novel method based on Gunnar Farneback Method, Mathematical Morphology, and Artificial Vision for Flow Analysis in Electrochemical Reactors. **Daniel A. Gutiérrez-Jiménez**, Sebastian Colores, Fernando Rivera, José López. *WITCOM 2023 (PUBLISHED)* https://doi.org/10.1007/978-3-031-45316-8_11.
- Preliminary Results of the Creation of an Optical Flow Database in Electrochemical Reactors: A Tool for the Development and Evaluation of Deep Learning Algorithms **Daniel A. Gutiérrez-Jiménez**, Sebastián Salazar-Colores, Fernando F. Rivera, José Trinidad López-Maldonado. *ICAppIEC 2024 (PUBLISHED)* <https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/4211>

Colaboración de documentos de conferencias

- Application and Comparison of State-of-the-Art Models in Semantic Segmentation for Laparoscopic Images ICAppIEC 2023 Diana J. Torres-López, **Daniel A. Gutiérrez-Jiménez**, Sebastián Salazar-Colores (ACCEPTED).

- Prueba de concepto de la aplicación de redes generativas adversarias cíclicas (CycleGANs) en el aumento de base de datos de imágenes laparoscópicas Diana Jazmín Torres-López, Sebastián Salazar-Colores, **Daniel A. Gutiérrez-Jiménez**, Alfonso Ramírez-Pedraza, J. Manuel Morales-Babines, Alejandra Aguilera-Barrientos, *Tomo VI: Ingeniería (PUBLICADO)*, ISBN 978-607-8821-48-8, Temacilli Editorial. https://cio.mx/archivos/avances_cient%C3%ADficos_tecnologicos_24/tomo_6.pdf
- Generación de bases de datos para entrenamiento de redes neuronales: comparativa y evaluación de métodos para la generación de imágenes con lluvia J. Manuel Morales-Babines, Sebastián Salazar-Colores, **Daniel A. Gutiérrez-Jiménez**, Diana Jazmín Torres-López, Alejandra Aguilera-Barrientos, *Tomo VI: Ingeniería (PUBLICADO)*, ISBN 978-607-8821-48-8, Temacilli Editorial. https://cio.mx/archivos/avances_cient%C3%ADficos_tecnologicos_24/tomo_6.pdf

El análisis no invasivo de flujos en reactores químicos industriales es esencial para optimizar el rendimiento y la eficiencia de estos sistemas en aplicaciones industriales. Esta tesis presenta Quetzal-Flow, un modelo de aprendizaje profundo diseñado específicamente para la estimación de flujo óptico en reactores químicos, que busca superar las limitaciones de las técnicas tradicionales de medición de flujo, como la velocimetría láser de efecto Doppler y la velocimetría de imágenes de partículas. Dichos métodos, aunque efectivos, implican altos costos y complejidades de infraestructura, limitando su aplicabilidad.

Quetzal-Flow implementa una arquitectura bifurcada con extracción jerárquica de características y módulos de refinamiento adaptativo, permitiendo una estimación de flujo óptico precisa y no invasiva en diversos escenarios de reactor. Evaluado en conjuntos de datos como FlyingChairs, FlyingThings3D, Sintel y KITTI2015, el modelo alcanza resultados de precisión competitivos, logrando un Error de Punto Final (EPE) de 0.64 en el subconjunto limpio de Sintel y 0.96 en el subconjunto final, superando a otros modelos avanzados como FlowNet, PWC-Net y RAFT.

Además, las visualizaciones generadas por Quetzal-Flow muestran su capacidad para capturar patrones complejos de movimiento y oclusiones, lo cual es crucial para aplicaciones en reactores químicos industriales. Este método ofrecerá una alternativa escalable

y más económica para el análisis hidrodinámico en reactores, facilitando un monitoreo más preciso del comportamiento del flujo sin la necesidad de equipos costosos. A pesar de sus logros, existen áreas de mejora en términos de eficiencia computacional y adaptabilidad a escenarios industriales reales, aspectos que se destacan como oportunidades para investigaciones futuras.

Índice general

Agradecimientos	III
Publicaciones	v
Resumen	vii
1. Introducción	1
1.1. Antecedentes	1
1.2. Definición del problema	4
1.3. Justificación	5
1.4. Objetivos	5
1.5. Hipótesis	6
2. Marco teórico	7
2.1. Reactores de Placas Paralelas en Procesos Electroquímicos	7

2.1.1.	Dinámica de Fluidos Computacional	9
2.1.2.	Velocimetría de imágenes de partículas (PIV)	9
2.1.3.	Flujo Óptico	10
2.1.4.	Matiz-Saturación-Valor y su Aplicación en el Análisis del Flujo Óptico	11
2.1.5.	Visión Artificial Basada en Flujo Óptico	12
2.1.6.	Algoritmo de Flujo Óptico de Gunnar Farneback	13
2.1.7.	Aprendizaje Automático	15
2.1.8.	Redes Neuronales Artificiales	16
2.1.9.	Aprendizaje Profundo	17
2.1.10.	Redes Neuronales Convolucionales	17
2.1.11.	Métricas para Evaluar el Flujo Óptico en Aprendizaje Profundo . . .	18
2.2.	Bases de Datos para la Caracterización del Flujo Óptico	20
2.2.1.	KITTI Paquete de Evaluación para Visión	21
2.2.2.	MPI-Sintel	23
2.2.3.	Flying Chairs	27
2.2.4.	FlyingChairsOcc	29
2.2.5.	ChairsSDHom	31
2.2.6.	HD1K	33
2.2.7.	Conjunto de Datos PIV para el Entrenamiento de Redes Neuronales	36
2.3.	Cálculo de Flujo Óptico Mediante Aprendizaje Profundo	38
2.3.1.	Transformaciones Recurrentes de Campos de Todos los Pares (RAFT)	38
2.3.2.	RAFT-PIV: Aprendizaje Profundo Recurrente de Flujo Óptico para Imágenes de Velocimetría de Partículas	40
2.3.3.	FlowNet	42

2.3.4. FlowNet 2.0	45
2.3.5. LiteFlowNet	46
2.3.6. PWC-Net	48
2.3.7. Refinamiento residual iterativo (IRR)	50
2.3.8. MaskFlowNet	52
2.3.9. Agregación Global de Movimiento (GMA)	53
3. Metodología	56
3.1. Desarrollo del proyecto	56
3.2. Hardware y Software	58
3.3. Desarrollo del método clásico para la estimación de flujo óptico	58
3.3.1. Implementación del Método de Gunnar Farneback	59
3.4. Generación de base de datos sintética	63
3.5. Prueba de modelos con bases de datos del estado del arte	68
3.5.1. Configuraciones de Entrenamiento	69
3.6. Resultados de las Redes de Flujo Óptico	70
3.6.1. FlowNet	70
3.6.2. FlowNet2	71
3.6.3. PWC-Net	73
3.6.4. LiteFlowNet	74
3.6.5. LiteFlowNet2	75
3.6.6. IRR	77
3.6.7. MaskFlowNet	78
3.6.8. RAFT	79

3.6.9. GMA	80
3.7. Prueba de modelo con base de datos propia	81
3.7.1. Preprocesamiento de Datos	81
3.7.2. Entrenamiento y Validación del Modelo	82
3.8. Desarrollo de un modelo basado en aprendizaje profundo	82
3.8.1. Selección de la Arquitectura de la Red Neuronal	83
3.8.2. Diseño de la red Quetzal-Flow	83
3.8.3. Bloque de Similitudes Visuales	84
3.8.4. Bloque de Actualización Iterativo	84
3.8.5. DenseNet (D)	85
3.8.6. Salida de la Red	86
3.8.7. Innovaciones de Quetzal-Flow	86
3.8.8. Preparación de Datos y Preprocesamiento	87
3.8.9. Implementación del Modelo	88
3.8.10. Entrenamiento del Modelo	89
3.8.11. Evaluación del Método Basado en Aprendizaje Profundo	90
4. Resultados	91
4.1. Volumen de Datos Generado	91
4.1.1. Resultados del Entrenamiento de la Red con Base de Datos Particles	92
4.2. Evaluación de los Resultados del Método Clásico	93
4.3. Resultados Generales de modelos con bases de datos del estado del arte	94
4.3.1. Rendimiento en Datasets Sintéticos	95
4.3.2. Rendimiento en Sintel (training)	95

4.3.3. Rendimiento en KITTI2012 y KITTI2015	95
4.3.4. Impacto del Fine-Tuning	96
4.3.5. Eficiencia Computacional	96
4.4. Resultados del Modelo de Aprendizaje Profundo Quetzal-Flow	96
4.4.1. Resultados del Entrenamiento	97
4.5. Evaluación Visual del Flujo Óptico Estimado	98
4.5.1. Resultados Visuales	99
4.6. Comparación de Quetzal-Flow y Gunnar Farneback	100
4.6.1. Comparación en $Y = 15$ mm	100
4.6.2. Comparación en $Y = 65$ mm	103
4.6.3. Comparación en $Y = 117$ mm	105
4.6.4. Análisis General	107
5. Conclusión y trabajo a futuro	110
Bibliografía	112

Índice de cuadros

3.1. Resultados de FlowNet: Métricas obtenidas en diversos conjuntos de datos. Las métricas incluyen el EPE, evaluado en los conjuntos FlyingChairs, Sintel (entrenamiento limpio y final) y KITTI2012. Los valores más bajos indican un mejor desempeño del modelo.	71
3.2. Resultados de FlowNet2: Métricas obtenidas en diversos conjuntos de datos. Las métricas incluyen EPE, FI-all para KITTI2015, y ChairsSDHom. Los valores más bajos indican un mejor desempeño del modelo.	72
3.3. Resultados de PWC-Net: Métricas obtenidas en diversos conjuntos de datos.	73
3.4. Resultados de LiteFlowNet: Métricas obtenidas en diversos conjuntos de datos.	75
3.5. Resultados de LiteFlowNet2: Métricas obtenidas en diversos conjuntos de datos.	76
3.6. Resultados de IRR-PWC: Métricas obtenidas en diversos conjuntos de datos.	77
3.7. Resultados de MaskFlowNet: Métricas obtenidas en diversos conjuntos de datos.	78

3.8. Resultados de RAFT: Métricas obtenidas en diversos conjuntos de datos.	79
3.9. Resultados de GMA: Métricas obtenidas en diversos conjuntos de datos.	80
4.1. Error cuadrático medio (MSE) en configuraciones de canal vacío, canalizado y bifurcado en diferentes posiciones.	107
4.2. Similitud de coseno en configuraciones de canal vacío, canalizado y bifurcado en diferentes posiciones.	108
4.3. Distancia de correlación para diferentes configuraciones de flujo y posiciones.	109

Índice de figuras

2.1. Configuraciones del reactor utilizadas en el proceso de experimentación: a) Canalizado, b) Bifurcado, c) Canal vacío. (Autoría propia).	9
2.2. Rueda de colores representando el modelo HSV (Matiz, Saturación, Valor), donde los grados indican las tonalidades de color en función de su ángulo en el círculo cromático (Autoría propia).	10
2.3. Análisis del flujo óptico en un sistema de partículas: (a) Flujo inicial en el tiempo t ; (b) desplazamiento de partículas en el tiempo t' , mostrando direcciones de flujo con flechas; (c) combinación de flujos ($t + t'$) resaltando el movimiento global y direcciones principales; (d) codificación de dirección de movimiento utilizando matices de color (HUE), donde los colores indican el ángulo del movimiento (Autoría propia).	12
2.4. Imagen original con el flujo óptico superpuesto del conjunto de datos KITTI Optical Flow . La imagen muestra una representación combinada de la escena original y el flujo óptico, lo cual es fundamental para la evaluación de algoritmos de flujo óptico en aplicaciones como la conducción autónoma y la visión por computadora. Generada a partir de datos de [1].	21

<p>2.5. Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos KITTI. El <i>Fotograma 1</i> muestra una escena urbana con varios objetos en movimiento, y el <i>Fotograma 2</i> captura la misma escena un instante después. El flujo óptico, en la parte inferior, representa el desplazamiento aparente de los objetos entre los dos fotogramas, lo cual es utilizado como referencia para la evaluación de algoritmos de flujo óptico. KITTI ofrece datos de alta precisión capturados en escenarios reales, lo que permite un análisis exhaustivo de algoritmos en condiciones urbanas desafiantes. Generada a partir de datos de [2].</p>	23
<p>2.6. Visualización de dos fotogramas consecutivos y el flujo óptico del conjunto de datos MPI-Sintel. El flujo óptico, representado en la imagen (c), muestra el desplazamiento de píxeles entre el <i>Fotograma 1</i> (a) y el <i>Fotograma 2</i> (b). Esta información es clave para analizar el movimiento aparente de objetos en la escena, proporcionando un <i>ground truth</i> preciso para la evaluación de algoritmos en tareas de visión por computadora. Generada a partir de datos de [3,4].</p>	26
<p>2.7. Visualización de dos <i>fotogramas</i> consecutivos y su correspondiente flujo óptico del conjunto de datos Flying Chairs. (a) <i>Fotograma 1</i> muestra una escena sintética con sillas 3D sobre un fondo de paisaje, en (b) <i>Fotograma 2</i> representa el mismo escenario tras una transformación afín aleatoria. En (c), el flujo óptico ilustra el desplazamiento aparente de los objetos entre los dos fotogramas, utilizado como referencia para la evaluación y entrenamiento de algoritmos de flujo óptico. Generada a partir de datos de [5]. . .</p>	28

<p>2.8. Visualización de dos <i>fotogramas</i> consecutivos y sus mapas de oclusión del conjunto de datos FlyingChairsOcc. En (a) y (b), los <i>fotogramas</i> 1 y 2 con sus respectivos mapas de oclusión destacan las áreas ocultas por el movimiento de los objetos. En (c, d) se observa el flujo óptico, y en (e) se muestra el desplazamiento aparente entre ambos <i>fotograma</i>. Este conjunto de datos es clave para mejorar la estimación del flujo óptico en escenarios con oclusiones, facilitando un análisis más preciso en entornos complejos. Generada a partir de datos de [6].</p>	30
<p>2.9. Fotogramas consecutivos y flujo óptico del conjunto de datos ChairsSDHom. Los <i>fotogramas</i> 1 y 2 presentan movimientos mínimos, diseñados para entrenar algoritmos en la detección de desplazamientos subpíxel. En (c), el flujo óptico captura estos movimientos sutiles, proporcionando un <i>ground truth</i> clave para evaluar modelos en escenarios con cambios delicados. ChairsSDHom mejora la capacidad de generalización de los modelos en condiciones reales. Generada a partir de datos de [7].</p>	32
<p>2.10. Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos hd1k. El <i>Fotograma 1</i> y el <i>Fotograma 2</i> muestran escenas urbanas capturadas en condiciones dinámicas y desafiantes para la estimación de flujo óptico, tales como variaciones en la iluminación y objetos en movimiento. En inciso (c), el flujo óptico representa el desplazamiento aparente entre los dos <i>fotogramas</i>, proporcionando una <i>ground truth</i> precisa para la evaluación de algoritmos en condiciones del mundo real. hd1k se utiliza para probar la robustez de los modelos en escenarios urbanos complejos, caracterizados por condiciones adversas como lluvia y baja iluminación. Generada a partir de datos de [8].</p>	35

2.11. Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos PIV. el inciso (a), el <i>Fotograma 1</i> muestra la distribución de partículas en un flujo isotrópico simulado mediante dinámica de fluidos computacional (CFD), y en (b), el <i>Fotograma 2</i> refleja los desplazamientos de las partículas un instante después. En (c) se aprecia, el flujo óptico proporciona el desplazamiento detallado de las partículas entre ambos fotogramas, actuando como un <i>ground truth</i> para entrenar y evaluar modelos de estimación de flujo óptico en imágenes de partículas. Este conjunto de datos es clave para mejorar la generalización de los modelos en experimentos reales de PIV. Generada a partir de datos de [9].	37
2.12. Componentes principales de RAFT: (1) Un codificador de características que extrae detalles píxel por píxel de ambas imágenes de entrada. (2) Una capa de correlación que crea un volumen de correlación 4D al comparar todas las combinaciones posibles de características. Las últimas dos dimensiones del volumen 4D se agrupan en varias escalas para obtener volúmenes de diferentes tamaños. (3) Un operador de actualización que mejora repetidamente el flujo óptico utilizando la estimación actual para buscar información en el conjunto de volúmenes de correlación. Adaptado de [10].	39
2.13. Esquema de RAFT-PIV adaptado a la estimación de flujo en imágenes de PIV. La arquitectura conserva el codificador de características y el volumen de correlación del RAFT original, con mejoras en la robustez frente a ruido y distribuciones irregulares de partículas. Adaptado de [11].	42
2.14. Dos arquitecturas de FlowNet: (a) FlowNetSimple y (b) FlowNetCorr. El cono verde representa la fase de refinamiento expansiva (c), como se ilustra en la figura. Estas redes, que incluyen la fase de refinamiento, se entrenan completamente de extremo a extremo. Adaptado de [12].	44
2.15. Esquema de la arquitectura de FlowNet 2.0 mostrando la estructura apilada y los refinamientos iterativos. Adaptado de [13].	46

2.16. Esquema de la arquitectura de LiteFlowNet destacando su diseño optimizado y la estructura en cascada para la inferencia de flujo. Adaptado de [14].	48
2.17. Esquema de la arquitectura de PWC-Net, destacando el uso de pirámides de características y la construcción del volumen de coste para la estimación de flujo óptico. Adaptado de [15].	49
2.18. Diagrama esquemático de IRR: (1) Refinamiento residual iterativo que reutiliza el mismo bloque de red en múltiples iteraciones para mejorar las estimaciones de flujo. (2) Estimación de oclusiones integrada para mejorar la precisión general. (3) Capas de refinamiento bilateral y reescalado que mejoran la definición en los bordes de movimiento y oclusiones. Adaptado de [16].	51
2.19. Esquema de la arquitectura de MaskFlowNet, destacando el módulo AsymOFMM y la integración de pirámides de características conscientes de oclusiones para la estimación de flujo óptico. Adaptado de [17].	53
2.20. Diagrama de la arquitectura GMA. La red toma dos imágenes de referencia y coincidencia, pasa ambas por una red de características que genera un volumen de correlación 4D. Posteriormente, un codificador de características de movimiento extrae características 2D del movimiento, las cuales son refinadas mediante el módulo GMA, que agrega información de movimiento global. Finalmente, las características concatenadas se introducen en una unidad GRU para producir un flujo residual refinado. Adaptado de [18].	55
3.1. Diagrama de flujo de la metodología para el proyecto. El proceso incluye el desarrollo de un método clásico, la generación de una base de datos sintética, y pruebas de modelos utilizando bases de datos del estado del arte y una base de datos propia. A continuación, se desarrolla un modelo basado en aprendizaje profundo, seguido de una comparación de desempeño entre modelos con y sin aprendizaje profundo. Finalmente, se elabora el reporte de resultados obtenidos.	57

3.2. Diagrama de flujo del proceso de estimación y análisis del flujo óptico utilizando un modelo neuronal. El proceso abarca desde la detección de la región de interés (ROI) y la estimación de la distribución del flujo, hasta el diseño, entrenamiento y evaluación del modelo neuronal. Las flechas indican el flujo de las etapas, con íconos que destacan puntos de evaluación: rojo para corrección y verde para éxito.	59
3.3. Flujo óptico estimado utilizando el método de Gunnar Farneback. En los incisos (a) y (b) se muestran los fotogramas de la base sintética creada, utilizados para calcular el flujo óptico; en (c) se presenta el flujo óptico calculado. Los colores representan la dirección y magnitud del flujo en cada píxel (Autoría propia).	62
3.4. Diagrama de flujo del proceso de estimación y análisis del flujo óptico utilizando un modelo neuronal. El proceso abarca desde la detección de la región de interés (ROI) y la estimación de la distribución del flujo, hasta el diseño, entrenamiento y evaluación del modelo neuronal. Las flechas indican el flujo de las etapas, con íconos que destacan puntos de evaluación: rojo para corrección y verde para éxito.	83
3.5. Diagrama de la arquitectura Quetzal-Flow, mostrando el bloque de similitudes visuales, el bloque de actualización iterativo y la red DenseNet para refinamiento del flujo óptico (Autoría propia).	87
4.1. Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos particles . A la izquierda, el <i>Fotograma 1</i> muestra la distribución de partículas simulando la caída de un fluido, y a la derecha, el <i>Fotograma 2</i> refleja los desplazamientos de las partículas un instante después. En la parte inferior, el flujo óptico proporciona el desplazamiento detallado de las partículas entre ambos fotogramas, actuando como un <i>ground truth</i> para entrenar y evaluar modelos de estimación de flujo óptico en imágenes de partículas (Autoría propia).	92

4.2. Evaluación visual de los resultados obtenidos con Quetzal-Flow en tres escenarios: canal vacío (primera fila), canal canalizado (segunda fila) y estructura bifurcada (tercera fila).	99
4.3. Comparación del perfil de velocidad en el eje Y = 15 mm para la configuración de canal vacío entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	101
4.4. Comparación del perfil de velocidad en el eje Y = 15 mm para la configuración canalizada entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	101
4.5. Comparación del perfil de velocidad en el eje Y = 15 mm para la configuración bifurcada entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	102
4.6. Comparación del perfil de velocidad en el eje Y = 65 mm para la configuración de canal vacío entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	103
4.7. Comparación del perfil de velocidad en el eje Y = 65 mm para la configuración canalizada entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	104
4.8. Comparación del perfil de velocidad en el eje Y = 65 mm para la configuración bifurcada entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	104
4.9. Comparación del perfil de velocidad en el eje Y = 117 mm para la configuración de canal vacío entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	105
4.10. Comparación del perfil de velocidad en el eje Y = 117 mm para la configuración canalizada entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	106

4.11. Comparación del perfil de velocidad en el eje $Y = 117$ mm para la configuración bifurcada entre los datos originales de CFD, Gunnar Farnebäck (GF), y Quetzal-Flow (QFlow).	106
--	-----

CAPÍTULO 1

Introducción

En este capítulo introductorio se aborda la importancia del flujo óptico en diversas aplicaciones, destacando los desafíos actuales y las oportunidades de mejora en el área. Se presentan los objetivos de la investigación, así como una revisión del estado del arte en las técnicas de estimación de flujo óptico. Además, se discuten las principales contribuciones del trabajo y se ofrece una guía sobre la estructura de los capítulos que conforman este documento.

1.1. Antecedentes

Los electrodos de placa plana se utilizan ampliamente en procesos electroquímicos debido a su versatilidad, bajo coste de mantenimiento, distribución uniforme del potencial y de la corriente, facilidad de construcción de diferentes arquitecturas de electrodos, capacidad de incorporar espaciadores en forma de red o mezcladores estáticos estructurados para mejorar el transporte de masa a la superficie del electrodo, y facilidad de escalado y control [19]. Entre sus aplicaciones están:

- Producción de cloro e hidróxido de sodio: Se utilizan electrodos de placa plana en el proceso de cloro-álcali, que es un método para producir cloro e hidróxido de sodio a partir de agua salada [19].
- Fabricación farmacéutica: Se utilizan electrodos de placa plana en la producción de diversos productos farmacéuticos, incluidos antibióticos, hormonas y vitaminas [20].
- Reacciones orgánicas e inorgánicas: Se utilizan electrodos de placa plana en la síntesis de diversos compuestos orgánicos e inorgánicos, incluyendo polímeros, tintes y catalizadores [21].
- Tratamiento de agua: Se utilizan electrodos de placa plana en el tratamiento de aguas residuales, incluida la eliminación de metales pesados, contaminantes orgánicos y agentes patógenos [22].
- Baterías: Se utilizan electrodos de placa plana en diversos tipos de baterías, incluidas las baterías de plomo-ácido, níquel-cadmio y Ion-litio [23].
- Pilas de combustible: Se utilizan electrodos de placa plana en pilas de combustible, que son dispositivos que convierten la energía química en energía eléctrica [24].

Sin embargo, los avances recientes en procesos electroquímicos requieren la exploración de diferentes diseños y configuraciones de reactores adaptados a procesos electroquímicos específicos. En el caso de reactores electroquímicos que utilizan electrodos de placa plana, la implementación de configuraciones de distribución de flujo se vuelve esencial para optimizar los patrones de fluidos y lograr un rendimiento competitivo. Estas configuraciones tienen como objetivo mejorar la distribución de transporte de masa sobre la superficie del electrodo, al tiempo que se minimizan los costos y la caída de presión. Por ejemplo, Escudero. [25] diseñó distribuidores de flujo para un reactor electroquímico de placa plana con un área proyectada de electrodo y membrana de 925 cm². Estudios numéricos y experimentales demostraron la uniformidad del campo de flujo a través de la zona de reacción, aunque a costa de una alta caída de presión, validando las simulaciones numéricas mediante visualización directa, pruebas de distribución de tiempo de residencia (RTD) y mediciones globales de caída de presión.

Marquez. [26] presentaron un estudio que involucra simulaciones numéricas y validación experimental para evaluar la hidrodinámica de un reactor electroquímico de flujo de placas paralelas hecho en casa. Se evaluaron varias configuraciones de distribuidores de entrada-salida, incluidos canales vacíos, colectores de flujo en serpentina, colectores de flujo rectos y separadores con forma de red en forma de diamante. Los estudios de simulación permitieron la evaluación de la distribución hidrodinámica en el entorno de reacción con diferentes configuraciones, que posteriormente se validaron mediante mediciones globales de RTD y visualizaciones directas.

Aunque las pruebas RTD y globales son técnicas poderosas para elucidar el comportamiento hidrodinámico y validar indirectamente los resultados de simulaciones computacionales, también se han explorado técnicas que implican mediciones locales de velocidad. Por ejemplo, velocimetría láser de efecto Doppler (LVD) y velocimetría de imágenes de partículas, especialmente en el dominio de mezcladores rotativos y reactores electroquímicos con generación de gas [27, 28]. Estas técnicas se basan en análisis ópticos para calcular la velocidad vectorial, utilizando partículas trazadoras o midiendo pérdidas de intensidad láser. Si bien son adecuadas para medir la velocidad local en recipientes transparentes de reactores de flujo, estos métodos a menudo requieren infraestructura costosa, como cámaras de alta velocidad, fuentes láser de alta potencia y software especializado, lo que puede limitar su resolución y aplicabilidad en reactores a escala de laboratorio.

Para superar estas limitaciones, es necesario explorar técnicas ópticas alternativas, especialmente para medir la velocidad en reactores electroquímicos a escala de laboratorio, donde las técnicas mencionadas anteriormente pueden ser susceptibles a diversas interferencias de luz. El uso de algoritmos de visión por computadora emerge como una opción prometedora para analizar mediciones de velocidad, aprovechando pruebas de visualización bien establecidas. Los algoritmos de visión por computadora, como el flujo óptico, se han utilizado ampliamente en mediciones a gran escala, incluyendo trayectorias de vehículos aéreos urbanos [29], control de robots voladores [30], flujos de ríos [31], patrones de velocidad en flujos pulsátiles y constantes en angiografía [32], dinámicas espacio-temporales de la actividad cerebral [33] y velocidad de crecimiento de cristales [34].

Después de realizar una búsqueda exhaustiva en las bases de datos más importantes, se observó que, en el campo de los reactores electroquímicos, los métodos de procesamiento de imágenes utilizados han sido generalmente básicos. Un ejemplo de ello es el enfoque presentado en [35], donde se utilizó un método simple para calcular indirectamente las respuestas experimentales de RTD mediante pruebas de visualización dentro del recipiente del reactor. En contraste, el método desarrollado en este trabajo introduce técnicas avanzadas de procesamiento de imágenes que ofrecen mayor precisión y robustez. Estas técnicas incluyen la apertura morfológica, el algoritmo de Ramer-Douglas-Peucker, el método de Otsu y un innovador filtro de desviación estándar para el preprocesamiento de datos. Tras este preprocesamiento, se aplica el método de flujo óptico de Gunnar Farneback [36] para estimar las mediciones de velocidad, mejorando significativamente la capacidad de análisis y cuantificación de las dinámicas del reactor.

1.2. Definición del problema

El problema que se aborda en este trabajo es la necesidad de implementar en reactores electroquímicos a escala de laboratorio métodos ópticos o computacionales alternativos para visualizar y analizar la velocidad local y global. Actualmente, se utilizan técnicas como la velocimetría láser de efecto Doppler y la velocimetría de imágenes de partículas para realizar estas mediciones. Sin embargo, estas técnicas presentan limitaciones en términos de costos de infraestructura y su aplicabilidad en reactores a escala de laboratorio.

Aunque existen propuestas computacionales muy recientes para el uso de métodos tradicionales de flujo óptico, como el de Gunnar Farneback, en el análisis de reactores electroquímicos, esta investigación propone el desarrollo de un algoritmo de flujo óptico basado en aprendizaje profundo como una alternativa más precisa y robusta para la estimación de velocidades en estos sistemas. Con este enfoque, se busca superar las limitaciones de los métodos tradicionales, facilitando un análisis efectivo de perfiles de velocidad en reactores electroquímicos. La meta es diseñar una solución fácil de implementar, que permita visualizar y analizar de manera detallada la velocidad del fluido, aspecto crucial para optimizar el rendimiento y la eficiencia de los procesos electroquímicos.

1.3. Justificación

En los reactores químicos es importante comprender cómo afecta las velocidades al proceso. Hoy en día no se cuenta en la literatura con métodos ampliamente aceptados para validar perfiles de velocidad. Por lo tanto, en un trabajo previo se propuso trabajar con flujo óptico, utilizando el algoritmo de Gunnar Farnebäck, con buenos resultados [36]. Este enfoque abre la puerta a trabajar con aprendizaje profundo para mejorar dicho proceso, haciéndolo más eficiente y permitiendo comparar los datos obtenidos de las simulaciones con los experimentales, validando así la eficacia del método propuesto.

La mejora de los procesos de distribución de flujo y eficiencia energética en los reactores electroquímicos ha ganado mucha relevancia. Este tipo de unidades es muy útil en el campo de la electrosíntesis de productos químicos especializados, incluidos los utilizados como fuentes de energía, como los electrolizadores para la producción de hidrógeno, las pilas de combustible, el tratamiento de residuos industriales y, en particular, en procesos innovadores como el diseño de células electrolíticas microbianas para la producción de biocombustibles y células de flujo redox para el almacenamiento de energía [35].

Partiendo de estos puntos, surge la necesidad de desarrollar nuevas formas de predecir el comportamiento dentro de los reactores químicos. Una opción prometedora es el uso de algoritmos de aprendizaje profundo con el objetivo de lograr rendimientos competitivos y más eficientes.

1.4. Objetivos

Objetivo general: Proponer y desarrollar un sistema de aprendizaje profundo, orientado a analizar el flujo en reactores químicos, que permitirá aproximaciones más precisas que otros métodos que no están basados en inteligencia artificial (tradicionales).

Objetivo específicos:

- Obtener secuencias de imágenes a partir de un montaje desarrollado para determinar los patrones de flujo.
- Generar una base de datos con la información generada a partir de un montaje desarrollado para determinar los patrones de flujo.
- Diseñar una red neuronal que permita estimar los vectores de movimiento del montaje, para determinar los patrones de flujo en el reactor.
- Entrenar un sistema de redes neuronales capaz de predecir el flujo en el reactor químico.
- Probar, evaluar y comparar el algoritmo de aprendizaje profundo de flujo óptico en distintas configuraciones de reactores electroquímicos de placas paralelas, contrastando los métodos de referencia en el estado del arte.

1.5. Hipótesis

Estimar vectores de movimiento del flujo en un reactor químico, mediante un método propuesto de inteligencia artificial (aprendizaje profundo), basado en técnicas novedosas del área, permitirá aproximaciones más precisas que otros métodos que no están basados en inteligencia artificial, para resolver problemas semejantes.

En este capítulo se presentan los fundamentos teóricos y técnicos del flujo óptico, describiendo los principios básicos, las metodologías clásicas y las arquitecturas modernas basadas en aprendizaje profundo. Además, se detallan los conjuntos de datos más relevantes utilizados para entrenar y evaluar las redes neuronales, así como las métricas estándar que permiten comparar su desempeño. Este marco teórico proporciona una base sólida para comprender las innovaciones desarrolladas en este trabajo.

2.1. Reactores de Placas Paralelas en Procesos Electroquímicos

Los reactores electroquímicos de placas paralelas son ampliamente utilizados en una variedad de procesos electroquímicos, destacándose por su versatilidad y las múltiples ventajas que ofrecen. Su diseño sencillo no solo facilita un mantenimiento económico, sino que también asegura una distribución uniforme del potencial y de la corriente, permitiendo una fácil adaptación a diversas configuraciones de electrodos. Además, estos reactores pueden incorporar separadores en forma de red o mezcladores estáticos estructurados,

lo que optimiza el transporte de masa hacia la superficie del electrodo.

Estos electrodos son fundamentales en la producción de una amplia gama de productos químicos y tienen un rol crucial en la industria farmacéutica, donde se utilizan en la síntesis de medicamentos. En el tratamiento de aguas, los reactores de placas paralelas son herramientas esenciales para la eliminación de metales pesados, contaminantes orgánicos y patógenos. Asimismo, tienen aplicaciones significativas en la industria de baterías, siendo componentes clave en tecnologías como las baterías de plomo-ácido, níquel-cadmio y ion-litio, así como en celdas de combustible [37].

Con los avances tecnológicos y las demandas crecientes de la industria, ha surgido la necesidad de explorar y adaptar diseños específicos de reactores para distintos procesos electroquímicos. Particularmente en aquellos reactores que emplean electrodos de placas planas, la implementación de distribuidores de flujo se ha vuelto esencial. En la Figura 2.1, se presentan tres configuraciones distintas de distribuidores de flujo, cada una nombrada según su diseño característico:

- **Canalizado:** Esta configuración utiliza canales lineales que guían el flujo de manera controlada a lo largo del reactor, optimizando la distribución uniforme del reactivo sobre la superficie del electrodo. Este diseño minimiza las zonas de estancamiento y las desviaciones de flujo, lo que resulta en una mayor eficiencia en el transporte de masa y en la distribución de corriente.
- **Bifurcado:** En esta disposición, el flujo se divide en varias ramas o bifurcaciones, lo que permite maximizar la cobertura en áreas más amplias del electrodo. Aunque puede generar mayores caídas de presión debido a la complejidad geométrica, mejora el transporte de masa al reducir las zonas de baja velocidad y facilitar la mezcla uniforme dentro del reactor.
- **Canal vacío:** Aquí, el flujo no sigue trayectorias específicas, ya que no se incluyen elementos de guiado interno, permitiendo un comportamiento más libre del fluido. Este diseño se utiliza frecuentemente como referencia para evaluar el efecto de configuraciones más complejas, ya que proporciona un patrón de flujo simplificado

y ayuda a identificar zonas críticas en condiciones hidrodinámicas generales.

Es en este contexto donde herramientas como la Dinámica de Fluidos Computacional (CFD), la Velocimetría de Imagen de Partículas (PIV), y más recientemente los algoritmos basados en flujo óptico, desempeñan un papel crucial para optimizar y evaluar el rendimiento de estos reactores.

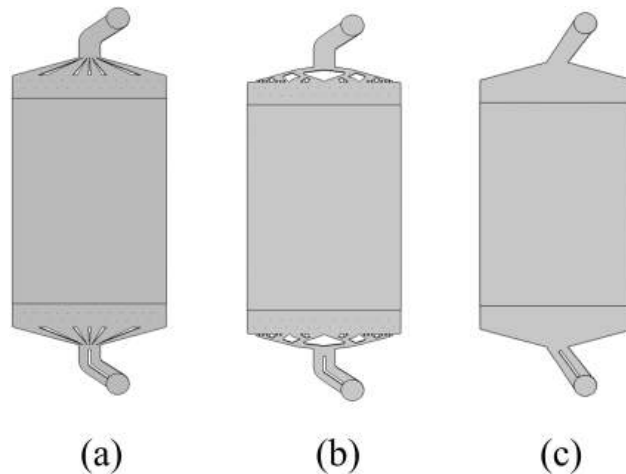


Figura 2.1: Configuraciones del reactor utilizadas en el proceso de experimentación: a) Canalizado, b) Bifurcado, c) Canal vacío. (Autoría propia).

2.1.1. Dinámica de Fluidos Computacional

La Dinámica de Fluidos Computacional (CFD) es una rama de la mecánica de fluidos que emplea algoritmos numéricos y análisis matemático para resolver y analizar problemas relacionados con el flujo de fluidos. Mediante la aplicación de CFD, es posible predecir el comportamiento del flujo, la transferencia de calor, y otros fenómenos asociados en un sistema o proceso determinado [38].

2.1.2. Velocimetría de imágenes de partículas (PIV)

PIV es una técnica experimental en la mecánica de fluidos que utiliza partículas trazadoras, iluminadas por láser, para medir las componentes de velocidad en un campo de flujo.

Es una técnica óptica no intrusiva que proporciona información detallada sobre los patrones de velocidad, siendo utilizada en diversas aplicaciones como el análisis de motores de combustión y estudios de flujo sanguíneo [11].

2.1.3. Flujo Óptico

El flujo óptico se define como el cambio aparente en la posición de los objetos o partículas entre dos imágenes consecutivas, derivado del análisis de sus intensidades lumínicas. Este desplazamiento se calcula entre un fotograma inicial (t) y uno posterior (t'), permitiendo la identificación de trayectorias, velocidades y direcciones relativas [39]. Esta técnica tiene aplicaciones en diversos campos, como el análisis de tráfico, la robótica y, más recientemente, la caracterización de procesos físico-químicos.

En la Figura 2.3, se ilustra cómo se pueden visualizar los desplazamientos de partículas en un flujo utilizando codificación por colores para representar la dirección y magnitud de los movimientos. Este enfoque facilita la interpretación de patrones complejos en sistemas dinámicos y es clave para mejorar la comprensión de procesos como la mezcla de fluidos o la distribución de reactivos en reactores electroquímicos.

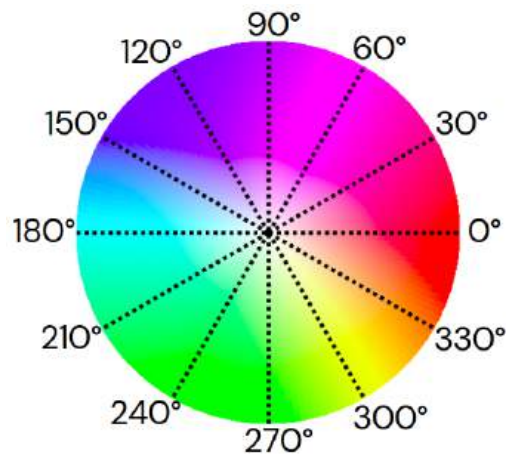


Figura 2.2: Rueda de colores representando el modelo HSV (Matiz, Saturación, Valor), donde los grados indican las tonalidades de color en función de su ángulo en el círculo cromático (Autoría propia).

2.1.4. Matiz-Saturación-Valor y su Aplicación en el Análisis del Flujo Óptico

El flujo óptico se representa comúnmente utilizando el modelo de color Matiz-Saturación-Valor (HSV, por sus siglas en inglés), el cual describe los colores en términos de tres componentes principales [40]:

- **Matiz (Hue):** Se refiere al tipo de color puro percibido, como rojo, azul o verde. Es esencial para distinguir un color de otro y sus valores oscilan entre 0° y 360° .
- **Saturación (Saturation):** Indica la intensidad o pureza del color, es decir, el grado de vibrancia o saturación. Un color con alta saturación es más intenso y vibrante. Los valores de saturación se expresan en porcentaje, variando entre el 0 % (color completamente desaturado o gris) y el 100 % (color completamente saturado o puro).
- **Valor (Value):** Representa la percepción de la cantidad de luz que emite o refleja un color, determinando su claridad u oscuridad. Los valores se miden en un rango del 0 % al 100 %, donde 0 % corresponde a negro y 100 % a blanco, con valores intermedios que representan diferentes niveles de luminosidad.

Estas características permiten representar los 360° de movimiento y su magnitud asociada en el análisis de flujo óptico. En la práctica, para calcular el flujo óptico, se emplean diversos métodos, como el método de Gunnar Farneback, que utiliza algoritmos computacionales para analizar los cambios en la intensidad lumínica entre imágenes sucesivas y estimar el movimiento de los objetos.

La aplicación del flujo óptico tiene importantes implicaciones en diversas industrias. Por ejemplo, se puede utilizar para detectar y corregir desviaciones en la distribución del flujo, identificar zonas muertas donde la mezcla es insuficiente o monitorear la formación de gradientes de concentración. Estas aplicaciones son fundamentales para optimizar el rendimiento y la productividad de los reactores de placas paralelas en una amplia variedad de contextos industriales.

2.1.5. Visión Artificial Basada en Flujo Óptico

En los últimos años, el uso de técnicas de visión artificial ha ganado relevancia en la caracterización de procesos electroquímicos complejos. Un ejemplo destacado es el desarrollo de sistemas basados en flujo óptico para analizar dinámicas de partículas y evaluar patrones de comportamiento en estos procesos [41]. Estas herramientas han demostrado ser de gran utilidad en la industria, proporcionando información detallada sobre flujos y movimientos internos, esenciales para el diseño y optimización de reactores electroquímicos.

En la Figura 2.3, se observa cómo el flujo óptico puede analizar el desplazamiento de partículas en movimiento, mostrando información clave sobre la dirección y velocidad de cada partícula a lo largo del tiempo.

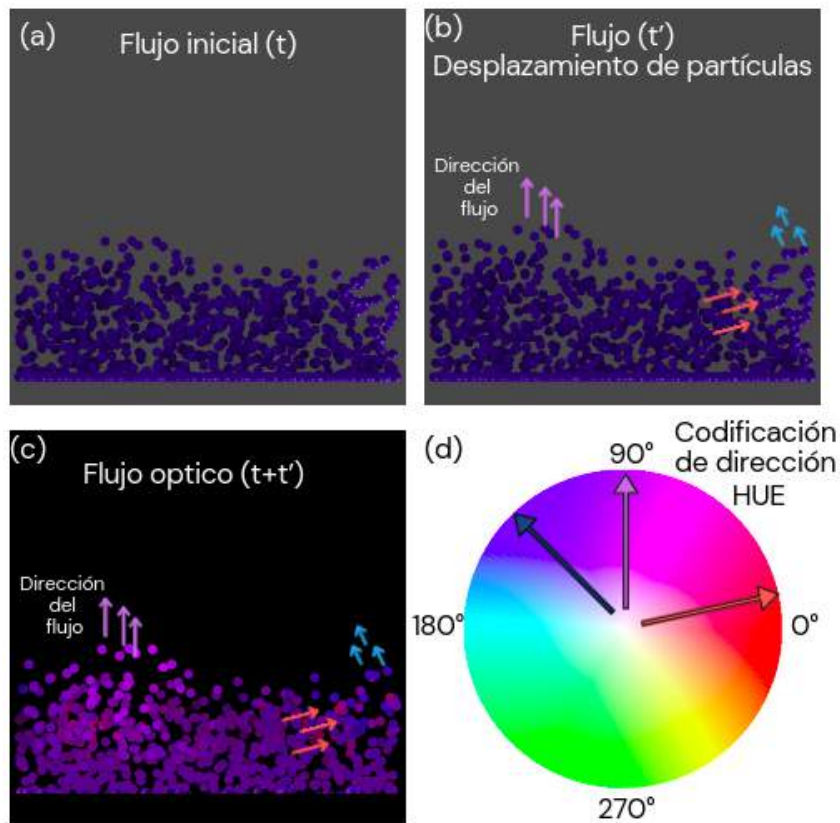


Figura 2.3: Análisis del flujo óptico en un sistema de partículas: (a) Flujo inicial en el tiempo t ; (b) desplazamiento de partículas en el tiempo t' , mostrando direcciones de flujo con flechas; (c) combinación de flujos ($t + t'$) resaltando el movimiento global y direcciones principales; (d) codificación de dirección de movimiento utilizando matices de color (HUE), donde los colores indican el ángulo del movimiento (Autoría propia).

La integración de la visión artificial basada en flujo óptico introduce una metodología innovadora en el análisis de patrones de flujo. Combina técnicas avanzadas de procesamiento de imágenes con enfoques tradicionales como CFD (Dinámica de Fluidos Computacional) y PIV (Velocimetría por Imágenes de Partículas), permitiendo una caracterización más precisa y eficiente de los sistemas electroquímicos. Esto abre nuevas posibilidades para optimizar los procesos industriales y acelerar la transición hacia tecnologías más sostenibles.

2.1.6. Algoritmo de Flujo Óptico de Gunnar Farnebäck

El campo de algoritmos para el cálculo del flujo óptico es amplio, destacándose la propuesta de Gunnar Farnebäck, basada en aproximaciones polinómicas de la vecindad de los píxeles y utilizando series de Taylor. Este enfoque ofrece una estimación densa del movimiento para cada píxel en la imagen [42].

El algoritmo de Farnebäck utiliza un método denominado expansión polinómica cuadrática 2.1, para estimar las variaciones en la intensidad de los píxeles dentro de una ventana alrededor de cada uno. El método predice cómo cambiará el brillo en los píxeles vecinos utilizando una fórmula matemática que involucra una matriz simétrica (A), un vector (b) y un valor constante (c). En esta fórmula, T representa la transposición de un vector o matriz, necesaria para realizar correctamente las operaciones matemáticas en el cálculo. Esto permite al algoritmo comprender los cambios de brillo en diferentes direcciones alrededor de cada píxel.

$$f(x) = x^T Ax + b^T x + c \quad (2.1)$$

El tamaño de la ventana en la expansión polinómica determina la escala de las características que se capturarán. Esto es crucial, ya que la región en la que se realiza la expansión influye en sí, se capturan detalles finos o más amplios. Si la ventana es muy pequeña, solo se capturan detalles locales, mientras que una ventana demasiado grande puede pasar

por alto información relevante. Por ejemplo, en imágenes con una resolución espacial de 1 metro, donde las características de interés tienen dimensiones menores a 10 metros, se elegiría un tamaño de ventana de aproximadamente 5 metros para asegurar una captura adecuada de las características deseadas.

El algoritmo de Farnebäck parte del supuesto de que hay pequeños desplazamientos entre las imágenes. Esta estimación se realiza mediante la ecuación de consistencia de brillo 2.2. Los términos de primer y segundo orden en esta ecuación se utilizan para calcular el movimiento d entre las imágenes.

$$\begin{aligned}
 f_1(x - d) &= f_2(x) \\
 &= x^T A_1 x + (b_1 - 2A_1 d)^T x + d^T A_1 d - b_1^T d + c_1 \\
 &= x^T A_2 x + b_2^T x + c_2,
 \end{aligned} \tag{2.2}$$

donde $A_2 = A_1$ y $b_2 = b_1 - 2A_1 d$.

En la práctica, pueden surgir discrepancias entre los valores de gradiente de las imágenes, lo que lleva a discrepancias entre A_1 y A_2 . Estos desajustes se corrigen minimizando los errores aplicando la ecuación 2.3. Debido a los diferentes movimientos locales entre las imágenes, la estimación del movimiento se realiza dentro de ventanas locales, como se describe en la ecuación 2.4 [43].

$$A = \frac{1}{2}(A_1 + A_2), \quad \Delta b = -\frac{1}{2}(b_2 - b_1), \quad A(x) \cdot d(x) = \Delta b \tag{2.3}$$

$$\sum_{w(\Delta x)} w(\Delta x) \|A(x + \Delta x) \cdot d(x) - \Delta b(x + \Delta x)\|^2 \tag{2.4}$$

Si bien este algoritmo ha sido ampliamente adoptado, presenta limitaciones en escenarios

donde se requiere una alta precisión, como en reactores electroquímicos. Para superar estas limitaciones, los algoritmos basados en técnicas de Aprendizaje Automático, particularmente aprendizaje profundo, han ganado relevancia. Estos enfoques ofrecen ventajas significativas, como la capacidad de aprender representaciones jerárquicas. A diferencia de las técnicas convencionales, los métodos basados en aprendizaje profundo identifican automáticamente patrones complejos en los datos, mejorando la precisión y adaptabilidad de los algoritmos.

Además, los algoritmos de aprendizaje profundo tienen una mayor capacidad de generalización, al entrenarse en una amplia gama de conjuntos de datos, lo que les permite adaptarse a escenarios variados. Finalmente, estos algoritmos se benefician del aprendizaje continuo, actualizándose con nuevos datos, lo que incrementa su precisión y robustez con el tiempo.

2.1.7. Aprendizaje Automático

El Aprendizaje Automático es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar automáticamente a partir de los datos y la experiencia previa. En lugar de programar explícitamente cada paso de un proceso, el aprendizaje automático proporciona datos a un algoritmo, permitiendo que la máquina descubra patrones y relaciones de manera autónoma.

El aprendizaje automático se puede clasificar en tres categorías principales:

1. **Aprendizaje Supervisado:** En este enfoque, los modelos se entrenan utilizando un conjunto de datos etiquetados, donde se conoce la salida deseada para cada ejemplo. El modelo aprende a mapear las entradas a las salidas a través de estos ejemplos, ajustando sus parámetros para minimizar el error de predicción.
2. **Aprendizaje No Supervisado:** En este tipo de aprendizaje, los modelos se entrenan con datos no etiquetados, donde el objetivo es descubrir patrones o estructuras sub-

yacentes en los datos. Es útil para tareas como la segmentación, el agrupamiento y la detección de anomalías.

3. **Aprendizaje por Refuerzo:** En este caso, los modelos interactúan con un entorno y aprenden mediante la retroalimentación en forma de recompensas o penalizaciones. El objetivo es que el modelo aprenda a tomar decisiones óptimas que maximicen la recompensa acumulada a lo largo del tiempo.

Aplicaciones del Aprendizaje Automático

El impacto del aprendizaje automático se extiende a una amplia variedad de campos, incluyendo:

- Reconocimiento de voz y procesamiento del lenguaje natural.
- Visión por computadora y reconocimiento de imágenes.
- Análisis predictivo en finanzas y mercadotecnia.
- Optimización de procesos industriales.
- Diagnóstico médico y pronóstico de enfermedades.
- Conducción autónoma y robótica.

2.1.8. Redes Neuronales Artificiales

En el ámbito del aprendizaje automático, las redes neuronales artificiales (ANN, por sus siglas en inglés) son sistemas matemáticos que emulan el funcionamiento del cerebro humano, permitiendo a las máquinas aprender a partir de datos. Estas redes están formadas por un conjunto de nodos interconectados, denominados neuronas artificiales, que se organizan en capas. Cada neurona artificial recibe entradas, las procesa mediante una función de activación y luego transmite la salida a otras neuronas en las capas posteriores.

Una red neuronal profunda (DNN, por sus siglas en inglés) es, esencialmente, un modelo que aproxima funciones no lineales a través de una extensa colección de unidades simples. Estas unidades están interconectadas mediante pesos variables, y su funcionalidad surge de la interacción colectiva entre las unidades, la conectividad definida por la arquitectura de la red, los pesos que enlazan las neuronas y las funciones de activación no lineales utilizadas. Esta estructura permite a las redes neuronales realizar tareas complejas de reconocimiento de patrones de manera eficiente, aprovechando su capacidad para modelar relaciones no lineales en los datos [44].

2.1.9. Aprendizaje Profundo

El aprendizaje profundo (o *deep learning*) es una subcategoría del aprendizaje automático que utiliza redes neuronales con tres o más capas. Estas redes intentan simular el comportamiento del cerebro humano, al menos de forma rudimentaria, para aprender a partir de grandes cantidades de datos. Mientras que una red neuronal con una sola capa puede aproximar únicamente relaciones lineales, las redes neuronales profundas tienen la capacidad de aproximar relaciones no lineales mediante el uso de múltiples capas que procesan datos de forma jerárquica. Esto las hace especialmente poderosas para tareas como la clasificación y la predicción en conjuntos de datos complejos y de gran volumen, como imágenes, texto o datos temporales [44].

2.1.10. Redes Neuronales Convolucionales

Una de las arquitecturas más utilizadas en el ámbito del aprendizaje profundo son las redes neuronales convolucionales (CNN, por sus siglas en inglés). Las CNN están diseñadas específicamente para el procesamiento de datos estructurados en forma de cuadrícula, como imágenes. Estas redes están compuestas por múltiples capas, entre las que se incluyen capas de convolución, de agrupación (*pooling*) y capas completamente conectadas. Cada una de estas capas contribuye a la extracción y aprendizaje de características jerárquicas de las imágenes de manera eficiente. Las CNN han demostrado ser altamente

efectivas en una variedad de tareas de visión por computadora, como el reconocimiento de objetos, la clasificación de imágenes y la segmentación semántica [45].

2.1.11. Métricas para Evaluar el Flujo Óptico en Aprendizaje Profundo

El rendimiento de los modelos de aprendizaje profundo en la estimación de flujo óptico se mide utilizando diversas métricas que cuantifican la precisión y robustez de las predicciones. Dos de las métricas más comunes en este campo son el **error de punto final** y el **F1-all**, cada una de las cuales evalúa el rendimiento del modelo desde diferentes perspectivas.

Error de Punto Final (EPE)

El EPE es una métrica fundamental en la evaluación de algoritmos de flujo óptico. Esta métrica calcula la distancia euclidiana entre el flujo óptico estimado y el *ground truth* en cada píxel. El EPE se define como:

$$EPE = \sqrt{(u_{\text{estimado}} - u_{\text{ground truth}})^2 + (v_{\text{estimado}} - v_{\text{ground truth}})^2}$$

donde u y v representan las componentes horizontales y verticales del flujo óptico, respectivamente. Cuanto menor sea el EPE, más precisa es la predicción del modelo. Esta métrica tiene sus raíces en los primeros algoritmos de flujo óptico como el método de Lucas-Kanade [46] y fue formalizada en el conjunto de datos Middlebury Optical Flow para proporcionar una evaluación estandarizada de los algoritmos [47].

F1-all (F1-all)

El **F1-all** o **F1-all** es otra métrica clave, especialmente utilizada en aplicaciones como la conducción autónoma. Esta métrica mide el porcentaje de píxeles con un error de estimación superior a un umbral predeterminado. Es útil para identificar cuántos píxeles fallan

en la estimación del flujo, proporcionando una evaluación del rendimiento en términos de robustez frente a grandes errores. La métrica **F1-all** fue popularizada por el conjunto de datos KITTI, donde se utiliza para medir el rendimiento de los algoritmos de flujo óptico en escenas urbanas complejas [1].

La fórmula para **F1-all** se define como:

$$F1-all = \frac{\text{Número de píxeles con error superior a un umbral}}{\text{Número total de píxeles}} \times 100\%$$

Esta métrica es especialmente relevante en escenarios del mundo real donde la presencia de oclusiones, objetos dinámicos y cambios rápidos en la iluminación pueden afectar significativamente la precisión de la estimación, ya que nos ayuda a identificar cuántos píxeles no están correctamente calculados.

Importancia de las Métricas

Ambas métricas, **EPE** y **F1-all**, proporcionan una visión complementaria del rendimiento de los algoritmos de flujo óptico. Mientras que el **EPE** evalúa la precisión global del flujo óptico en todos los píxeles, el **F1-all** mide la robustez del modelo frente a grandes errores en escenarios desafiantes. Estas métricas son esenciales para evaluar la generalización y aplicabilidad de los modelos en diferentes entornos, desde simulaciones controladas hasta aplicaciones del mundo real, como la conducción autónoma.

Referencias Clave

- **EPE** tiene sus raíces en los primeros trabajos de flujo óptico, como el método de Lucas-Kanade [46], y fue formalizado como métrica en la base de datos Middlebury Optical Flow [47].
- **F1-all** fue introducido en el conjunto de datos KITTI para la evaluación de algoritmos en entornos urbanos reales, proporcionando una medición robusta de los errores en la estimación del flujo [1].

2.2. Bases de Datos para la Caracterización del Flujo Óptico

El aprendizaje profundo ha revolucionado el campo de la visión por computadora, especialmente en la estimación del flujo óptico, al proporcionar herramientas más precisas y adaptables en comparación con los métodos tradicionales [42]. Estas metodologías utilizan redes neuronales profundas que, debido a su capacidad para modelar relaciones complejas en los datos, han superado las técnicas clásicas en términos de exactitud y rendimiento. Sin embargo, el éxito de estos algoritmos depende significativamente de la calidad y la cantidad de los datos empleados durante su fase de entrenamiento. Por lo tanto, disponer de bases de datos robustas, detalladas y representativas es fundamental para el desarrollo de algoritmos de aprendizaje profundo eficaces. Dichas bases de datos no solo facilitan el entrenamiento de los modelos, sino que también son esenciales para su evaluación y validación antes de su aplicación en escenarios del mundo real.

Entre las bases de datos más reconocidas en la literatura se encuentran el **KITTI Optical Flow 2012 y 2015**, desarrolladas por el Karlsruhe Institute of Technology y el Toyota Technological Institute at Chicago (ver Figura 2.4). Estas bases de datos están orientadas principalmente a aplicaciones de conducción autónoma y análisis de escenas urbanas [1]. Asimismo, otras bases de datos clave en el campo incluyen **MPI Sintel**, **Flying Chairs**, **FlyingChairsOcc**, **ChairsSDHom**, **hd1k**, y el **PIV dataset for neural network**. Estas bases permiten entrenar modelos robustos y adaptables, capaces de generalizar su rendimiento en diversos entornos y condiciones.

La mayoría de estas bases de datos incluyen un par de fotogramas consecutivos, T_0 y T_1 , junto con archivos `.flo` que contienen mapas de desplazamiento en las direcciones u y v . Estos mapas son esenciales para el análisis detallado del flujo óptico, permitiendo a los modelos aprender las complejidades del movimiento entre imágenes secuenciales.



Figura 2.4: Imagen original con el flujo óptico superpuesto del conjunto de datos **KITTI Optical Flow**. La imagen muestra una representación combinada de la escena original y el flujo óptico, lo cual es fundamental para la evaluación de algoritmos de flujo óptico en aplicaciones como la conducción autónoma y la visión por computadora. Generada a partir de datos de [1].

2.2.1. KITTI Paquete de Evaluación para Visión

KITTI Paquete de Evaluación para Visión en inglés KITTI Vision Benchmark Suite es uno de los conjuntos de datos más reconocidos en la investigación de visión por computadora, especialmente en el contexto de la conducción autónoma. Desarrollado por el Karlsruhe Institute of Technology y el Toyota Technological Institute at Chicago, KITTI ofrece una plataforma robusta para la evaluación de algoritmos en tareas fundamentales como el flujo óptico, la odometría visual (SLAM) y la detección de objetos en 3D. En cuanto al flujo óptico, KITTI incluye 194 pares de imágenes para entrenamiento y 195 para pruebas en su versión 2012, así como 200 pares de imágenes para entrenamiento y otros 200 para pruebas en su versión 2015.

Características Principales

El conjunto de datos KITTI fue diseñado para capturar escenarios del mundo real, incluyendo entornos urbanos, rurales y autopistas. A diferencia de otros conjuntos de datos que se limitan a entornos controlados o imágenes generadas sintéticamente, KITTI se caracteriza por incluir datos obtenidos en situaciones complejas y variadas, lo que añade un nivel considerable de dificultad a las tareas de visión por computadora.

El sistema de adquisición de datos utilizado en KITTI incluye los siguientes componentes:

- **Cámaras estéreo de alta resolución:** Se emplean dos sistemas de cámaras estéreo, una en blanco y negro y otra en color, con una resolución de 1240x376 píxeles después de la rectificación.
- **Escáner láser 3D Velodyne HDL-64E:** Este escáner es capaz de generar más de un millón de puntos 3D por segundo, proporcionando una información precisa sobre la profundidad de las escenas.
- **Sistema de localización avanzada OXTS RT 3003:** Combina GPS, GLONASS, una unidad de medida inercial (IMU) y señales de corrección RTK para obtener datos de localización con una precisión de menos de 5 cm bajo condiciones de cielo despejado.

Estas características permiten que KITTI proporcione *ground truth* de alta precisión y semidensidad para las imágenes capturadas, lo cual es fundamental para una evaluación rigurosa de algoritmos en tareas como la estimación de flujo óptico.

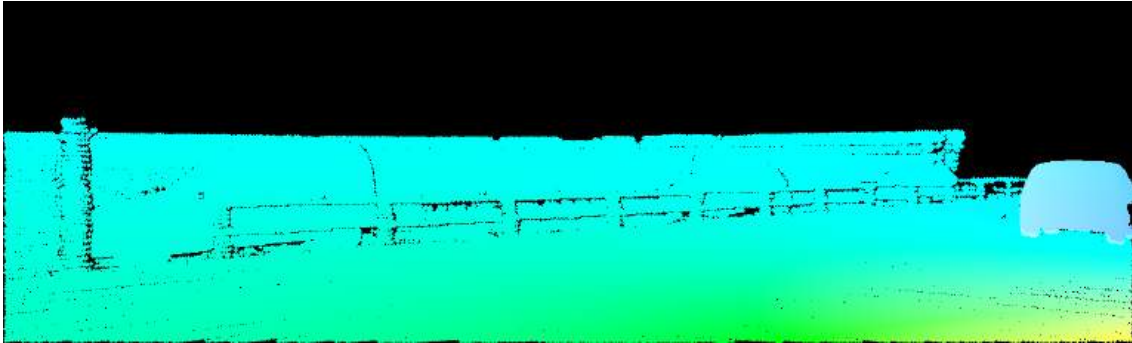
A continuación, en la Figura 2.5, se muestran dos fotogramas consecutivos de una secuencia del conjunto de datos KITTI, junto con la visualización del flujo óptico generado. El flujo óptico ilustra el movimiento aparente de los objetos entre ambos fotogramas, lo que se utiliza como referencia para la evaluación de algoritmos de visión por computadora.



(a) Fotograma 1



(b) Fotograma 2



(c) Flujo óptico

Figura 2.5: Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos **KITTI**. El *Fotograma 1* muestra una escena urbana con varios objetos en movimiento, y el *Fotograma 2* captura la misma escena un instante después. El flujo óptico, en la parte inferior, representa el desplazamiento aparente de los objetos entre los dos fotogramas, lo cual es utilizado como referencia para la evaluación de algoritmos de flujo óptico. KITTI ofrece datos de alta precisión capturados en escenarios reales, lo que permite un análisis exhaustivo de algoritmos en condiciones urbanas desafiantes. Generada a partir de datos de [2].

El conjunto de datos KITTI ha demostrado ser un recurso clave para el desarrollo de algoritmos robustos en visión por computadora. Los resultados obtenidos en este conjunto de datos revelan que muchos métodos que funcionan bien en conjuntos más simples, como Middlebury, encuentran dificultades cuando se enfrentan a las complejidades del mundo real capturadas en KITTI. Esto subraya la importancia de utilizar datos realistas para evitar el sobreajuste a entornos controlados y fomentar el desarrollo de soluciones que sean efectivas en aplicaciones prácticas, como la conducción autónoma [2].

2.2.2. MPI-Sintel

En los últimos años, los conjuntos de datos estandarizados han desempeñado un papel esencial en la visión por computadora, permitiendo la evaluación objetiva de algoritmos y

fomentando la competencia dentro de la comunidad científica. Históricamente, uno de los conjuntos de datos más utilizados para la estimación de flujo óptico fue el de Middlebury. Sin embargo, con la creciente complejidad de los algoritmos actuales, Middlebury ya no presenta un desafío suficiente.

Para cubrir esta necesidad, se desarrolló el conjunto de datos sintético **MPI-Sintel**, basado en la película de animación CGI de código abierto "Sintel". Este conjunto de datos destaca por ofrecer una amplia variedad de movimientos complejos y objetos no rígidos, lo que lo convierte en un recurso más realista y desafiante para la evaluación de algoritmos de flujo óptico. MPI-Sintel también incorpora degradaciones visuales como desenfoque por movimiento y efectos atmosféricos, características ausentes en conjuntos de datos anteriores como Middlebury. El conjunto incluye aproximadamente 1064 pares de imágenes para entrenamiento en las modalidades Clean y Final [3, 4].

MPI-Sintel fue creado mediante el uso del software de renderizado avanzado **Blender**, lo que permitió generar datos sintéticos de alta calidad con un control total sobre las condiciones de las escenas. Esto facilita la evaluación de algoritmos en situaciones variadas y complejas, ayudando a identificar limitaciones y áreas de mejora. Los experimentos han demostrado que los algoritmos que obtenían excelentes resultados en Middlebury, enfrentan dificultades significativas con los desafíos presentados por MPI-Sintel, subrayando la importancia de este conjunto de datos en la investigación continua del flujo óptico.

Características Principales

El conjunto de datos MPI-Sintel presenta varias características clave que lo posicionan como un recurso fundamental para la evaluación de algoritmos de flujo óptico:

- **CGI:** Al estar basado en escenas generadas por computadora (CGI) de la película "Sintel", el conjunto permite un control total sobre los elementos de la escena, como la iluminación, texturas y movimientos de los objetos, asegurando condiciones consistentes para la evaluación.
- **Diversidad de movimientos:** Las secuencias incluyen movimientos grandes y com-

plejos, así como objetos no rígidos, lo que presenta un desafío adicional para los algoritmos de flujo óptico que no es abordado en conjuntos de datos más sencillos, como Middlebury.

- **Tres modalidades de renderizado:** MPI-Sintel ofrece tres variantes de cada secuencia:
 - **Albedo:** Imágenes con superficies sin sombras, usadas para evaluar el rendimiento del flujo óptico en condiciones controladas.
 - **Clean:** Introduce sombras suaves y algunos efectos de luz, aumentando la complejidad visual.
 - **Final:** Contiene todos los efectos visuales, como desenfoques por movimiento, sombras complejas y efectos atmosféricos, representando el mayor desafío para los algoritmos.

- **Condiciones adversas:** Introduce degradaciones visuales, como desenfoques por movimiento, niebla y variaciones de iluminación, permitiendo evaluar el rendimiento de los algoritmos en situaciones más realistas y complicadas.

- **Compatibilidad con benchmarks previos:** A pesar de su mayor complejidad, MPI-Sintel es compatible con la metodología de evaluación utilizada en Middlebury, facilitando la comparación directa entre algoritmos evaluados en ambos conjuntos.

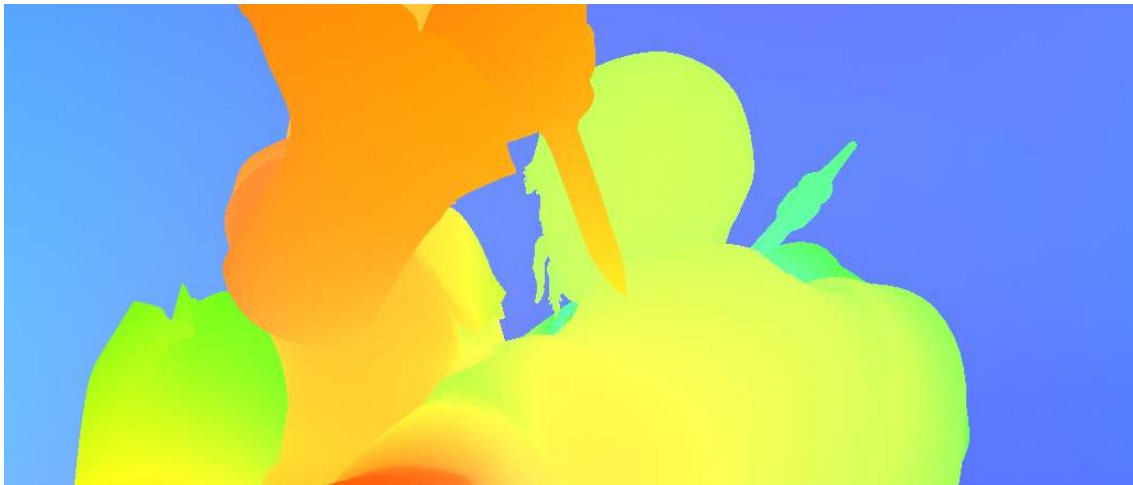
A continuación, en la Figura 2.6, se muestran dos fotogramas consecutivos de una secuencia del conjunto de datos MPI-Sintel, junto con la visualización del flujo óptico proporcionado. El flujo óptico ilustra el desplazamiento de los objetos y el entorno entre ambos fotogramas, y se utiliza como referencia para la evaluación de algoritmos de flujo óptico.



(a) Fotograma 1



(b) Fotograma 2



(c) Flujo óptico

Figura 2.6: Visualización de dos fotogramas consecutivos y el flujo óptico del conjunto de datos **MPI-Sintel**. El flujo óptico, representado en la imagen (c), muestra el desplazamiento de píxeles entre el *Fotograma 1* (a) y el *Fotograma 2* (b). Esta información es clave para analizar el movimiento aparente de objetos en la escena, proporcionando un *ground truth* preciso para la evaluación de algoritmos en tareas de visión por computadora. Generada a partir de datos de [3,4].

La variedad de complejidad visual y degradaciones atmosféricas incluidas en MPI-Sintel lo convierten en un banco de pruebas exhaustivo y robusto para la investigación en flujo óptico. Estas características permiten a los investigadores evaluar los límites de los algoritmos y desarrollar soluciones más generales y robustas para aplicaciones del mundo real [3,4].

2.2.3. Flying Chairs

El conjunto de datos **Flying Chairs** fue desarrollado para superar las limitaciones del conjunto de datos MPI-Sintel, el cual resulta insuficiente en tamaño para entrenar redes neuronales convolucionales (CNN) de gran escala. Este conjunto sintético fue diseñado específicamente para generar grandes volúmenes de datos de entrenamiento, esenciales para el aprendizaje profundo en tareas como la estimación de flujo óptico.

El proceso de creación de Flying Chairs se basa en la recopilación de 964 imágenes de Flickr, clasificadas en categorías como ciudad, paisaje y montaña. Estas imágenes se dividieron en cuadrantes y se utilizaron como fondos sobre los cuales se superpusieron imágenes de sillas 3D, obtenidas de un conjunto público de modelos. Se seleccionaron 809 tipos de sillas, cada una con 62 vistas diferentes, y se les aplicaron transformaciones afines aleatorias. Estas transformaciones, tanto en las sillas como en los fondos, generaron la segunda imagen de cada par, junto con el correspondiente campo de flujo óptico y las regiones de oclusión.

Características Principales

- **Cantidad de datos:** El conjunto contiene un total de 22,872 pares de imágenes con sus respectivos campos de flujo óptico.
- **Imágenes base:** Se utilizaron 964 imágenes de Flickr, divididas en cuadrantes, pertenecientes a categorías como ciudad, paisaje y montaña.
- **Modelos 3D de sillas:** Se emplearon 809 modelos de sillas 3D, cada uno con 62 vistas distintas, obtenidos de un conjunto público de modelos 3D.
- **Transformaciones afines aleatorias:** A cada silla se le aplicaron transformaciones afines aleatorias, que también se aplicaron al fondo para generar la segunda imagen de cada par.
- **Propósito:** Este conjunto está diseñado para el preentrenamiento de CNNs, mejorando significativamente el rendimiento en conjuntos de datos más complejos como

MPI-Sintel.

- **Generalización a datos reales:** A pesar de ser un conjunto de datos sintético, Flying Chairs ha demostrado que las CNNs entrenadas con este conjunto generalizan bien a datos reales como los de Sintel y KITTI.

A continuación, en la figura 2.7, se muestran dos fotogramas consecutivos de una secuencia del conjunto de datos Flying Chairs junto con la visualización del flujo óptico generado. El flujo óptico representa el movimiento aparente de los objetos y el entorno entre ambos fotogramas, y es utilizado como referencia para la evaluación de algoritmos.



(a) Fotograma 1



(b) Fotograma 2



(c) Flujo óptico

Figura 2.7: Visualización de dos *fotogramas* consecutivos y su correspondiente flujo óptico del conjunto de datos **Flying Chairs**. (a) *Fotograma 1* muestra una escena sintética con sillas 3D sobre un fondo de paisaje, en (b) *Fotograma 2* representa el mismo escenario tras una transformación afín aleatoria. En (c), el flujo óptico ilustra el desplazamiento aparente de los objetos entre los dos fotogramas, utilizado como referencia para la evaluación y entrenamiento de algoritmos de flujo óptico. Generada a partir de datos de [5].

El conjunto de datos Flying Chairs ha demostrado ser esencial para el preentrenamiento de CNNs, mejorando significativamente el rendimiento en la estimación de flujo óptico cuando se aplica a conjuntos de datos más complejos como MPI-Sintel [5].

2.2.4. FlyingChairsOcc

El conjunto de datos **FlyingChairsOcc** se desarrolló para abordar las limitaciones del conjunto original Flying Chairs en la estimación de flujo óptico y oclusiones. Aunque Flying Chairs proporcionó un valioso recurso para entrenar redes neuronales convolucionales (CNN) en la estimación de flujo óptico, no incluía información detallada sobre oclusiones, lo cual es crucial para mejorar la precisión de los modelos en escenarios reales.

FlyingChairsOcc sigue un protocolo similar al de Flying Chairs para la generación de datos, pero incorpora etiquetado explícito para mapas de oclusión y flujo inverso. Este conjunto de datos fue creado utilizando 964 imágenes de fondo de alta resolución, obtenidas de plataformas como Flickr y Google, con palabras clave como ciudad, calle y montaña. Se añadieron 809 modelos de sillas 3D, renderizadas desde modelos CAD, aplicándoles transformaciones afines aleatorias. Este enfoque permitió generar pares de imágenes junto con sus respectivos flujos ópticos y mapas de oclusión.

Las visualizaciones de los *fotogramas*, sus mapas de oclusión y el flujo óptico asociado se muestran en la Figura 2.8. Estos ejemplos ilustran cómo FlyingChairsOcc proporciona un análisis más detallado al incluir tanto la información de movimiento como las áreas ocluidas, mejorando la capacidad de las redes para estimar el flujo en situaciones complejas.

Características Principales

- **Etiquetas de oclusión:** FlyingChairsOcc incorpora mapas detallados de oclusiones, permitiendo entrenar modelos en escenarios donde las áreas ocluidas son comunes.
- **Etiquetas de flujo inverso:** Además del flujo óptico directo, el conjunto de datos proporciona información sobre el flujo óptico inverso, lo cual es útil para mejorar la precisión de las predicciones.

- **Pares de imágenes:** El conjunto contiene 22,232 pares de imágenes para entrenamiento y 640 para validación, con etiquetas para flujo óptico directo, inverso y mapas de oclusión.
- **Escenarios diversos:** Las imágenes de fondo incluyen entornos urbanos, naturales y montañosos, con una amplia variedad de sillas 3D que simulan condiciones realistas.
- **Transformaciones afines:** Al igual que en Flying Chairs, se aplicaron transformaciones afines aleatorias para generar movimiento entre los pares de imágenes.



(a) Fotograma 1



(b) Fotograma 2



(c) Oclusión (Fotograma 1)



(d) Oclusión (Fotograma 2)



(e) Flujo óptico

Figura 2.8: Visualización de dos *fotogramas* consecutivos y sus mapas de oclusión del conjunto de datos **FlyingChairsOcc**. En (a) y (b), los *fotogramas* 1 y 2 con sus respectivos mapas de oclusión destacan las áreas ocultas por el movimiento de los objetos. En (c, d) se observa el flujo óptico, y en (e) se muestra el desplazamiento aparente entre ambos *fotograma*. Este conjunto de datos es clave para mejorar la estimación del flujo óptico en escenarios con oclusiones, facilitando un análisis más preciso en entornos complejos. Generada a partir de datos de [6].

Este etiquetado adicional es esencial para mejorar la capacidad de las redes neuronales en la estimación conjunta de flujo y oclusiones, abordando uno de los principales desafíos en visión por computadora: la correcta identificación y manejo de áreas ocluidas [6].

2.2.5. ChairsSDHom

El conjunto de datos **ChairsSDHom** fue desarrollado para abordar una limitación específica identificada en los modelos de flujo óptico entrenados con conjuntos de datos como **Flying Chairs** y **FlyingThings3D**: la dificultad para estimar movimientos pequeños, típicos en datos del mundo real. Mientras que **Flying Chairs** es efectivo para entrenar modelos que capturan grandes desplazamientos, este enfoque no siempre se traduce bien a la estimación de movimientos sutiles, que son comunes en escenarios como los videos de acción real.

ChairsSDHom mantiene la simplicidad visual del conjunto de datos **Flying Chairs**, pero introduce una mayor precisión en la estimación de desplazamientos pequeños. Se creó un conjunto de datos con movimientos mínimos y un fondo homogéneo o con gradientes de color, diseñado para entrenar redes neuronales en la tarea de capturar estos pequeños movimientos. El conjunto fue diseñado de manera que su histograma de desplazamientos fuera más representativo de lo que se encuentra en conjuntos de datos del mundo real, como **UCF101**.

La visualización de los *fotograma* y el flujo óptico correspondiente de **ChairsSDHom** se muestra en la Figura 2.9. En los *fotogramas* consecutivos, se presentan desplazamientos sutiles, lo que permite entrenar modelos en la detección de movimientos pequeños y subpíxel.

Características Principales

- **Simplicidad Visual:** Similar al conjunto de datos **Flying Chairs**, este conjunto utiliza objetos de apariencia sencilla (principalmente sillas) en escenas artificiales para mantener un enfoque claro en los desplazamientos.

- **Enfoque en Movimientos Pequeños:** Diseñado específicamente para capturar movimientos mínimos, el conjunto de datos introduce desplazamientos sutiles que suelen pasar desapercibidos en otros conjuntos de datos.
- **Fondos Homogéneos o con Gradientes de Color:** Los fondos no complejos permiten que el enfoque del entrenamiento se mantenga en los desplazamientos del objeto, ayudando a las redes neuronales a especializarse en la detección de cambios sutiles en la escena.
- **Histograma de Desplazamientos Representativo:** El histograma de desplazamientos en **ChairsSDHom** es más equilibrado, acercándose a lo que se observa en escenarios del mundo real, como en los conjuntos de datos **UCF101**, lo que ayuda a mejorar la generalización de los modelos.
- **Mejora del Rendimiento en Desplazamientos Subpíxel:** **ChairsSDHom** está diseñado para mejorar la precisión en la estimación de desplazamientos subpíxel, crucial en aplicaciones que requieren un alto nivel de detalle.



(a) Fotograma 1



(b) Fotograma 2



(c) Flujo óptico

Figura 2.9: Fotogramas consecutivos y flujo óptico del conjunto de datos ChairsSDHom. Los *fotogramas* 1 y 2 presentan movimientos mínimos, diseñados para entrenar algoritmos en la detección de desplazamientos subpíxel. En (c), el flujo óptico captura estos movimientos sutiles, proporcionando un *ground truth* clave para evaluar modelos en escenarios con cambios delicados. ChairsSDHom mejora la capacidad de generalización de los modelos en condiciones reales. Generada a partir de datos de [7].

El conjunto de datos **ChairsSDHom** se posiciona como una herramienta esencial para mejorar la estimación de flujo óptico en situaciones que requieren precisión en movimientos pequeños. Al especializarse en desplazamientos subpíxel y ofrecer un histograma de desplazamientos más realista, este conjunto de datos ayuda a entrenar modelos que no solo sean competentes en manejar grandes desplazamientos, sino también en capturar con precisión los movimientos sutiles que son comunes en el mundo real. **ChairsSDHom** complementa efectivamente otros conjuntos de datos más complejos y equilibrados, asegurando que los modelos desarrollados sean versátiles y precisos en una amplia gama de escenarios de flujo óptico [7].

2.2.6. HD1K

El conjunto de datos **hd1k**, parte de la suite de evaluación **HCI Benchmark**, fue diseñado específicamente para abordar los desafíos que presenta la conducción autónoma en entornos urbanos. Este conjunto de datos proporciona una referencia altamente realista y desafiante para la estimación de flujo óptico y estéreo, con un enfoque particular en situaciones complejas, tales como condiciones de baja luminosidad y mal tiempo [8].

hd1k se compone de secuencias de imágenes capturadas con un sistema de cámaras estéreo de alta resolución y alto rango dinámico, montado en un vehículo. A diferencia de otros conjuntos de datos, **hd1k** incluye variaciones radiométricas y geométricas controladas de forma sistemática, lo que permite evaluar el rendimiento de los algoritmos en condiciones que emulan las dificultades reales de los escenarios urbanos.

Una de las innovaciones más destacadas de **hd1k** es su enfoque en la precisión de la *ground truth* mediante el uso de un escáner LIDAR de alta precisión. Este sistema captura mapas de profundidad y flujo óptico con un nivel de detalle sin precedentes, incluyendo distribuciones de error por píxel derivadas de simulaciones de Monte Carlo. Además, se proporcionan máscaras binarias que indican la incertidumbre en cada píxel, lo cual es crucial para evaluar y mejorar la robustez de los algoritmos bajo condiciones del mundo real.

El conjunto de datos incluye 55 secuencias seleccionadas manualmente, con entre 19 y 100 fotogramas cada una, abarcando una variedad de situaciones urbanas complejas, como cambios bruscos de iluminación, lluvia intensa y presencia de objetos dinámicos. Este nivel de detalle y precisión convierte a **hd1k** en una herramienta esencial para el desarrollo y validación de algoritmos de visión por computadora, especialmente en el contexto de la conducción autónoma [8].

A continuación, en la figura 2.10, se muestra una visualización de dos *fotogramas* consecutivos de una secuencia del conjunto de datos **hd1k** junto con el flujo óptico correspondiente. La visualización destaca cómo el flujo óptico captura el desplazamiento aparente de objetos en una escena urbana compleja, lo que es fundamental para evaluar la robustez de los algoritmos en condiciones del mundo real.

Características Principales

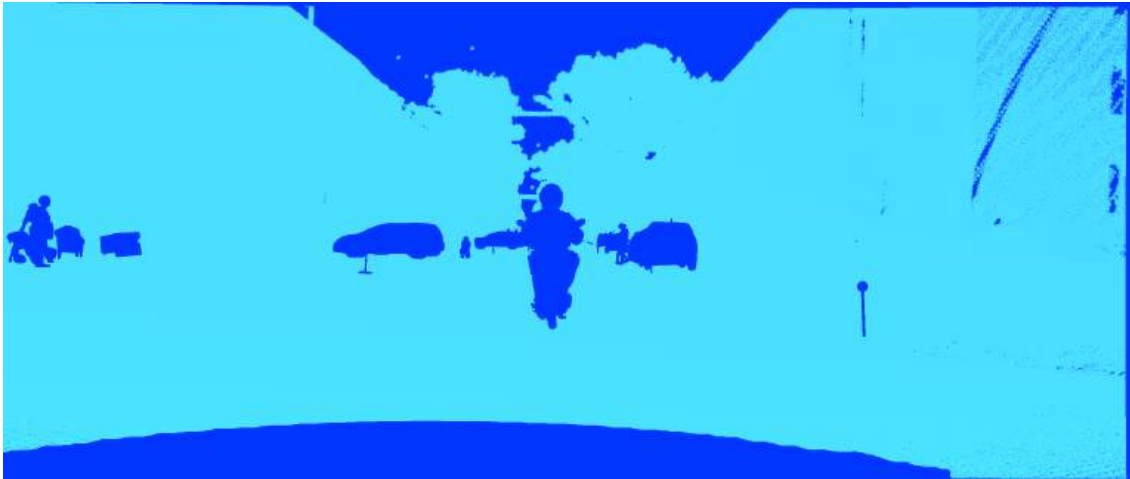
- **Sistema de cámaras estéreo:** Captura imágenes de alta resolución y alto rango dinámico, simulando condiciones reales de conducción urbana.
- **Escáner LIDAR de alta precisión:** Utilizado para generar mapas detallados de profundidad y flujo óptico, proporcionando información precisa por cada píxel.
- **Máscaras de incertidumbre:** Proporciona máscaras binarias que indican la incertidumbre en cada píxel, evaluando así la robustez de los algoritmos.
- **Variaciones radiométricas y geométricas:** Incluye escenas con variaciones controladas para simular diferentes condiciones de iluminación y geometría.
- **55 secuencias:** Consta de 55 secuencias urbanas seleccionadas manualmente, con entre 19 y 100 fotogramas cada una, presentando desafíos como cambios de iluminación, lluvia y objetos en movimiento.



(a) Fotograma 1



(b) Fotograma 2



(c) Flujo óptico

Figura 2.10: Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos **hd1k**. El *Fotograma 1* y el *Fotograma 2* muestran escenas urbanas capturadas en condiciones dinámicas y desafiantes para la estimación de flujo óptico, tales como variaciones en la iluminación y objetos en movimiento. En inciso (c), el flujo óptico representa el desplazamiento aparente entre los dos *fotogramas*, proporcionando una *ground truth* precisa para la evaluación de algoritmos en condiciones del mundo real. **hd1k** se utiliza para probar la robustez de los modelos en escenarios urbanos complejos, caracterizados por condiciones adversas como lluvia y baja iluminación. Generada a partir de datos de [8].

El conjunto de datos **hd1k** es fundamental para el desarrollo de algoritmos robustos en la conducción autónoma, abordando las dificultades que presentan los entornos urbanos complejos [8].

2.2.7. Conjunto de Datos PIV para el Entrenamiento de Redes Neuronales

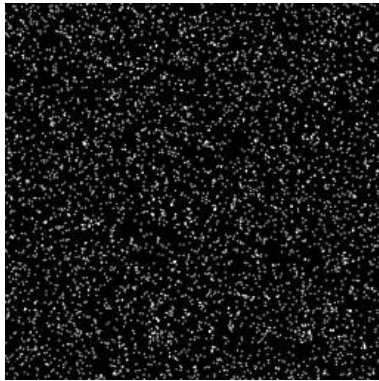
Este trabajo presenta una red neuronal convolucional (CNN) diseñada para la estimación densa de movimiento en imágenes de partículas, utilizando un conjunto de datos sintético creado específicamente para esta tarea. A diferencia de los conjuntos de datos tradicionales empleados en velocimetría por imágenes de partículas (PIV), este conjunto de datos sintético permite generar una cantidad considerablemente mayor de datos con un control riguroso sobre las condiciones de simulación.

Características Principales

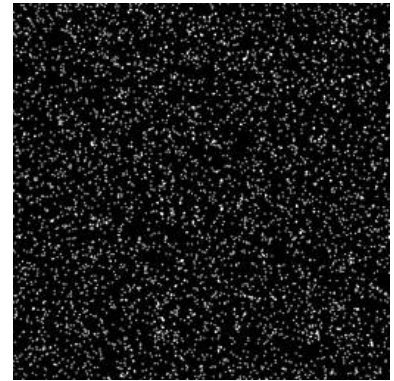
- **Generación mediante CFD:** La base de datos se creó utilizando simulaciones de dinámica de fluidos computacional (CFD), lo que permite capturar el movimiento detallado de las partículas distribuidas en el fluido. Estas simulaciones proporcionan campos de velocidad precisos para cada imagen generada.
- **Campos de velocidad como *ground truth*:** Los campos de velocidad generados por CFD actúan como la verdad de suelo (*ground truth*) para entrenar el modelo CNN, proporcionando vectores de desplazamiento exactos para cada partícula en la imagen.
- **Diversidad de condiciones:** El conjunto de datos incluye una variedad amplia de configuraciones de flujo, desde flujos laminares hasta turbulentos, lo que ayuda al modelo a adaptarse a diferentes escenarios de movimiento en fluidos.
- **Simulación de ruido:** Para mejorar la robustez del modelo, se introduce ruido en las imágenes, simulando las imperfecciones presentes en experimentos reales de PIV. Esto asegura que el modelo funcione de manera efectiva en entornos experimentales reales.
- **Entrenamiento y validación:** La base de datos permitió entrenar el modelo CNN bajo un esquema supervisado, donde las imágenes sintéticas de partículas sirven

como entrada y los campos de velocidad simulados son las etiquetas de entrenamiento.

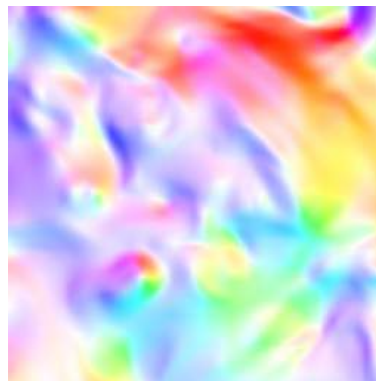
En la figura 2.11 se presenta un ejemplo visual de dos frames consecutivos y el flujo óptico generado, proporcionando un contexto claro para la utilidad del conjunto de datos.



(a) Fotograma 1



(b) Fotograma 2



(c) Flujo óptico

Figura 2.11: Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos PIV. el inciso (a), el *Fotograma 1* muestra la distribución de partículas en un flujo isotrópico simulado mediante dinámica de fluidos computacional (CFD), y en (b), el *Fotograma 2* refleja los desplazamientos de las partículas un instante después. En (c) se aprecia, el flujo óptico proporciona el desplazamiento detallado de las partículas entre ambos fotogramas, actuando como un *ground truth* para entrenar y evaluar modelos de estimación de flujo óptico en imágenes de partículas. Este conjunto de datos es clave para mejorar la generalización de los modelos en experimentos reales de PIV. Generada a partir de datos de [9].

Esta base de datos sintética fue fundamental para el entrenamiento de la red FlowNetS, adaptada específicamente para la estimación de movimiento en imágenes de partículas. El uso de datos generados artificialmente permitió entrenar el modelo con una gran cantidad de ejemplos, mejorando su capacidad para generalizar a imágenes experimentales del mundo real [9].

2.3. Cálculo de Flujo Óptico Mediante Aprendizaje Profundo

La aproximación convencional al cálculo del flujo óptico ha consistido en resolver un problema de optimización sobre el espacio de los campos de desplazamiento densos entre dos imágenes. Sin embargo, en los últimos años, el aprendizaje profundo ha surgido como una alternativa prometedora frente a los métodos tradicionales. En lugar de formular un problema de optimización, el aprendizaje profundo permite entrenar redes neuronales para predecir directamente el flujo óptico. Las metodologías actuales basadas en aprendizaje profundo han alcanzado un rendimiento comparable al de los mejores métodos convencionales, con la ventaja adicional de una mayor velocidad en la etapa de inferencia.

Entre las arquitecturas de redes neuronales convolucionales (CNN) empleadas para el cálculo del flujo óptico destacan las siguientes:

2.3.1. Transformaciones Recurrentes de Campos de Todos los Pares (RAFT)

RAFT es una arquitectura que ha demostrado resultados sobresalientes en la predicción de campos de desplazamiento, destacándose por su capacidad recurrente para capturar relaciones temporales, lo que mejora significativamente la precisión en la estimación del flujo óptico.

La arquitectura RAFT se compone de tres elementos principales:

1. **Codificador de Características:** Este módulo extrae un vector de características

para cada píxel, procesando la información detallada de ambas imágenes de entrada.

2. **Capa de Correlación:** Genera un volumen de correlación en 4D que abarca todas las combinaciones posibles de píxeles entre las dos imágenes. Posteriormente, se realiza un proceso de agrupación para reducir la resolución y facilitar el manejo eficiente de los datos.
3. **Operador de Actualización Recurrente basado en GRU (Gated Recurrent Unit):** Este operador utiliza unidades GRU para controlar el flujo de información a lo largo de las secuencias temporales. Toma los valores del volumen de correlación y realiza una actualización iterativa sobre un campo de flujo, inicialmente establecido en cero.

La Figura 2.12 proporciona una representación visual de la arquitectura de RAFT.

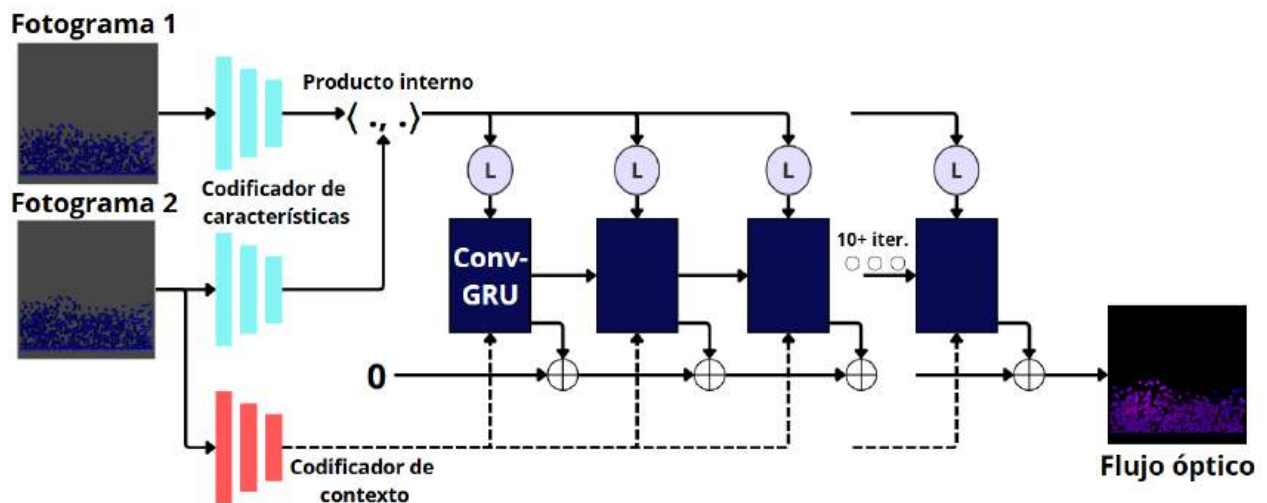


Figura 2.12: Componentes principales de RAFT: (1) Un codificador de características que extrae detalles píxel por píxel de ambas imágenes de entrada. (2) Una capa de correlación que crea un volumen de correlación 4D al comparar todas las combinaciones posibles de características. Las últimas dos dimensiones del volumen 4D se agrupan en varias escalas para obtener volúmenes de diferentes tamaños. (3) Un operador de actualización que mejora repetidamente el flujo óptico utilizando la estimación actual para buscar información en el conjunto de volúmenes de correlación. Adaptado de [10].

RAFT destaca por su precisión, logrando reducciones significativas en el error de estimación del flujo óptico en conjuntos de datos desafiantes como KITTI y Sintel. En KITTI,

RAFT alcanza un error F1-all del 5.10 %, representando una mejora del 16 % respecto a los mejores resultados anteriores. En Sintel, logra reducir el error de punto final a 2.855 píxeles, una mejora del 30 %. Además, RAFT demuestra una excepcional capacidad de generalización, manteniendo un bajo error incluso al utilizar datos sintéticos. En términos de eficiencia, RAFT procesa videos a 10 fotogramas por segundo en una GPU 1080Ti, y requiere 10 veces menos iteraciones de entrenamiento que otras arquitecturas. Incluso en su versión más compacta, logra una ejecución de 20 fotogramas por segundo en la base de datos Sintel [10].

2.3.2. RAFT-PIV: Aprendizaje Profundo Recurrente de Flujo Óptico para Imágenes de Velocimetría de Partículas

El modelo **RAFT-PIV** es una adaptación del exitoso método RAFT específicamente diseñado para mejorar la estimación de movimiento en imágenes de velocimetría de partículas (PIV). RAFT-PIV combina la precisión y eficiencia de la arquitectura RAFT original con modificaciones que permiten manejar las características únicas de los datos de PIV, como la densidad de partículas y las dinámicas de flujo de fluidos.

En la Figura 2.13, se muestra un esquema de la arquitectura general de RAFT-PIV, el cual detalla cómo los componentes clave del RAFT original han sido adaptados para abordar las particularidades de las imágenes de PIV.

Componentes Principales de RAFT-PIV

RAFT-PIV conserva los tres elementos clave de la arquitectura RAFT, con ajustes específicos para su aplicación en imágenes de PIV:

1. **Codificador de Características:** Este módulo extrae características a nivel de píxel de las imágenes de partículas, capturando información detallada sobre el desplazamiento de las partículas en el flujo. El codificador está adaptado para manejar la variabilidad en la distribución y densidad de las partículas.

2. **Capa de Correlación:** Similar al RAFT original, RAFT-PIV genera un volumen de correlación 4D que contiene todas las combinaciones posibles de características entre las partículas en las dos imágenes. Este volumen permite evaluar relaciones entre todas las partículas, asegurando una estimación precisa del flujo entre las imágenes.
3. **Operador de Actualización Recurrente:** RAFT-PIV emplea un operador recurrente basado en GRU (Gated Recurrent Unit) para realizar actualizaciones iterativas sobre el campo de flujo óptico. Este operador ajusta iterativamente el flujo estimado, utilizando la información del volumen de correlación para refinar la predicción del desplazamiento de las partículas en cada iteración.

Mejoras en RAFT-PIV para PIV

RAFT-PIV introduce varias mejoras para adaptarse mejor a las características particulares de las imágenes de PIV:

- **Estimación de Flujo Denso y Preciso:** RAFT-PIV permite la estimación de desplazamientos para cada píxel de la imagen, proporcionando un flujo óptico denso en lugar de los campos dispersos tradicionales obtenidos mediante técnicas de correlación cruzada.
- **Capacidad de Generalización en Datos Experimentales:** Aunque se entrena principalmente en datos sintéticos, RAFT-PIV muestra una excelente capacidad de generalización, adaptándose bien a condiciones experimentales reales de PIV, donde las partículas pueden tener distribuciones irregulares y flujos complejos.
- **Reducción del Ruido:** RAFT-PIV incluye mecanismos para reducir la sensibilidad al ruido en las imágenes, lo cual es crucial en los datos de PIV, donde las partículas pequeñas o dispersas pueden generar interferencias en la estimación del flujo.

RAFT-PIV logra resultados sobresalientes en la estimación de desplazamientos en imágenes de PIV. Comparado con técnicas tradicionales de correlación cruzada, RAFT-PIV

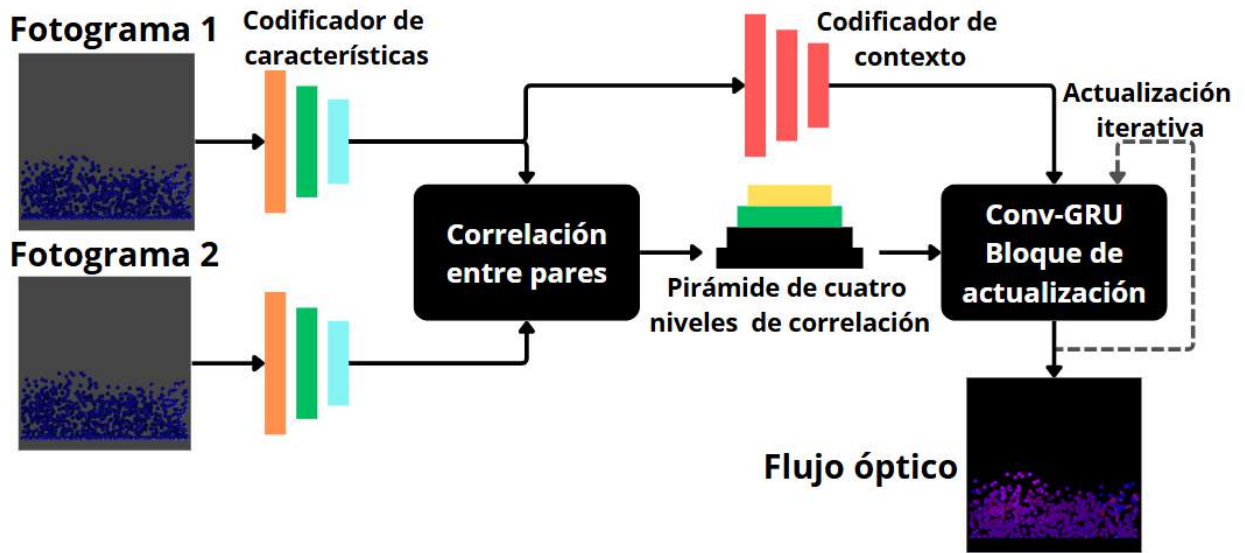


Figura 2.13: Esquema de RAFT-PIV adaptado a la estimación de flujo en imágenes de PIV. La arquitectura conserva el codificador de características y el volumen de correlación del RAFT original, con mejoras en la robustez frente a ruido y distribuciones irregulares de partículas. Adaptado de [11].

proporciona una mayor resolución espacial y precisión en la estimación de flujos turbulentos. Los experimentos muestran que RAFT-PIV es capaz de capturar dinámicas complejas de fluidos con una precisión notable, abriendo nuevas posibilidades para la implementación de estimación de flujo óptico en experimentos de mecánica de fluidos [11].

2.3.3. FlowNet

Otra arquitectura destacada en el cálculo de flujo óptico es **FlowNet**, la cual ha sido pionera en el uso de redes neuronales convolucionales (CNN) para esta tarea. FlowNet se enfoca en la eficiencia y en la capacidad de aprender representaciones de características relevantes para el movimiento entre imágenes.

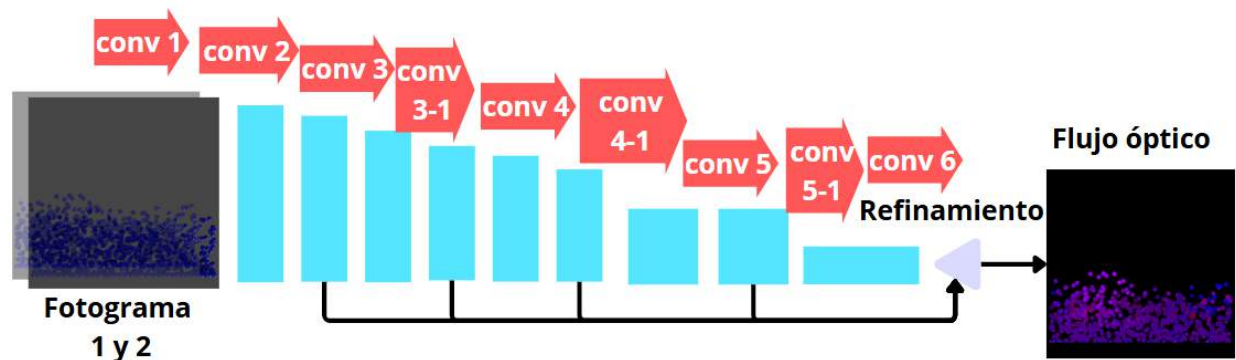
1. **Parte Contractiva:** En esta fase, una opción sencilla es apilar ambas imágenes de entrada y alimentarlas a través de una red convolucional genérica. Esta red tiene la capacidad de decidir cómo procesar el par de imágenes para extraer información de movimiento. Este enfoque se ilustra en la Figura 2.14 (a) y se denomina Flow-

NetSimple.

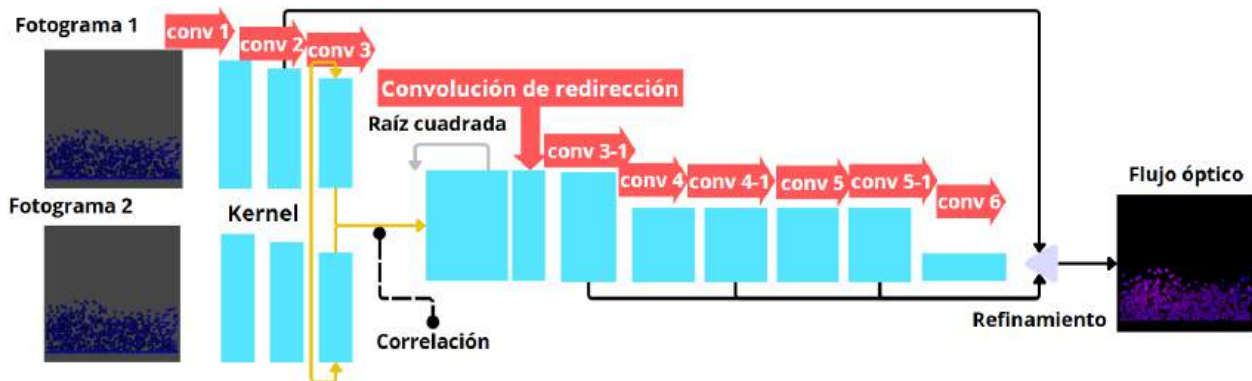
Otra aproximación implica la creación de dos flujos de procesamiento separados pero idénticos para las dos imágenes, combinándolos en una etapa posterior, como se muestra en la Figura 2.14 (b). En esta arquitectura, la red está limitada a producir representaciones significativas de las dos imágenes por separado antes de combinarlas a un nivel superior. Este enfoque se asemeja a la técnica tradicional de coincidencia, donde primero se extraen características de parches de ambas imágenes y luego se comparan esos vectores de características.

Para facilitar el proceso de coincidencia, se introduce una capa de correlación que realiza comparaciones multiplicativas de parches entre dos mapas de características. Esta capa permite a la red comparar cada parche de un mapa de características con cada parche del otro mapa.

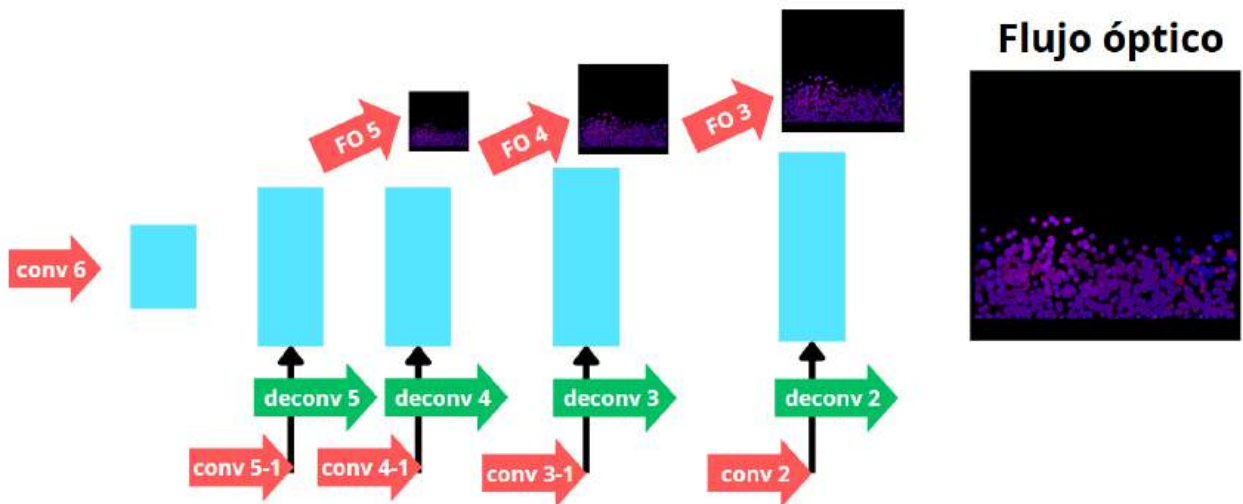
2. **Parte Expansiva:** El componente principal de la parte expansiva Figura 2.14 (c) consiste en las capas upconvolucionales, las cuales incluyen una operación de unpooling (extensión de los mapas de características) seguida de una convolución. Estas capas se aplican para el refinamiento y se concatenan con los mapas de características correspondientes de la parte contractiva de la red, junto con una predicción de flujo de menor resolución reescalada, si está disponible. Esto preserva tanto la información de alto nivel de los mapas de características más gruesos como la información local fina de las capas más bajas. En cada paso, la resolución se aumenta por un factor de dos. Este proceso se repite cuatro veces, resultando en un flujo predicho cuya resolución sigue siendo cuatro veces menor que la resolución de entrada.



(a) FlowNetSimple



(b) FlowNetCorr



(c) Refinamiento Expansivo

Figura 2.14: Dos arquitecturas de FlowNet: (a) FlowNetSimple y (b) FlowNetCorr. El cono verde representa la fase de refinamiento expansiva (c), como se ilustra en la figura. Estas redes, que incluyen la fase de refinamiento, se entrenan completamente de extremo a extremo. Adaptado de [12].

FlowNet se destaca por su eficiencia y alto rendimiento. Gracias a su implementación

optimizada en GPU, FlowNet puede predecir el flujo óptico en hasta 10 pares de imágenes por segundo, incluso en la resolución completa del conjunto de datos Sintel [12].

2.3.4. FlowNet 2.0

FlowNet 2.0 representa una evolución significativa respecto a su versión anterior en la estimación de flujo óptico basada en redes neuronales profundas, superando las limitaciones del FlowNet. Aunque FlowNet demostró que la estimación de flujo óptico podía abordarse como un problema de aprendizaje profundo, sus resultados aún podían ser mejorados, especialmente en desplazamientos pequeños y en datos del mundo real [13].

FlowNet 2.0 introduce varias mejoras clave para abordar estas deficiencias:

1. **Arquitectura Apilada:** FlowNet 2.0 implementa una arquitectura apilada que permite el refinamiento iterativo del flujo óptico estimado. Cada red en la pila recibe como entrada no solo las imágenes originales, sino también una estimación previa del flujo, que es refinada en cada paso mediante la operación de deformación (*warping*) de la segunda imagen. Este enfoque mejora la precisión de la estimación, especialmente en desplazamientos pequeños.
2. **Subred Especializada en Pequeños Movimientos:** Se introduce una subred dedicada a manejar desplazamientos pequeños. Esta subred se entrena con un conjunto de datos específico diseñado para mejorar la capacidad de la red en situaciones que involucran movimientos sutiles.
3. **Curriculum Learning:** Otra innovación importante es el uso de un programa de entrenamiento que alterna entre diferentes conjuntos de datos, comenzando con datos sintéticos simples y progresando hacia datos más complejos y realistas. Este enfoque, conocido como *curriculum learning*, permite que la red desarrolle primero una comprensión básica del flujo óptico antes de enfrentarse a escenarios más complejos, mejorando notablemente la generalización y el rendimiento en *benchmarks* como MPI Sintel y KITTI.

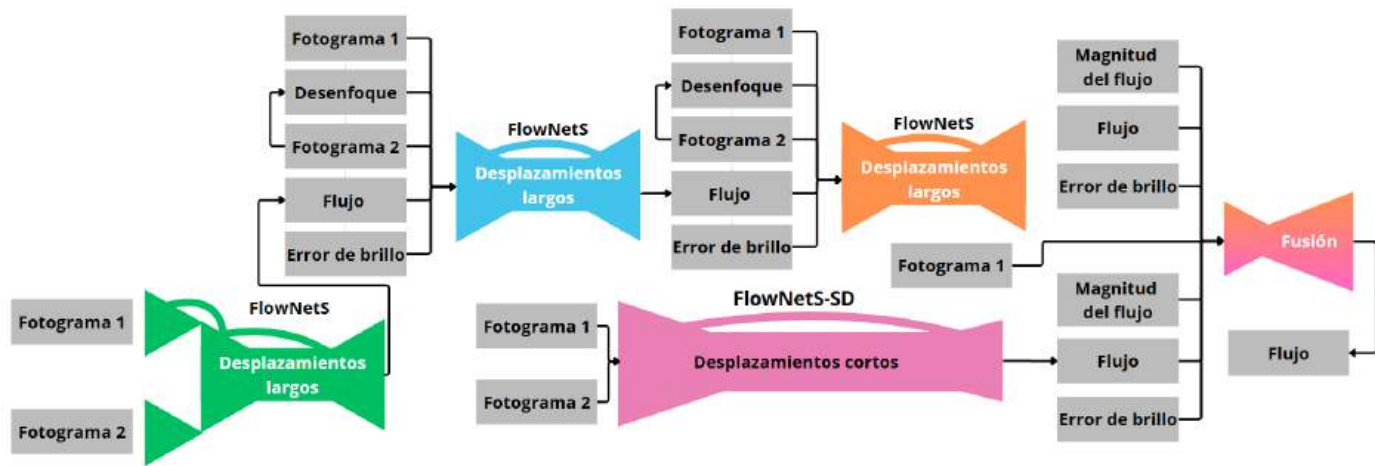


Figura 2.15: Esquema de la arquitectura de FlowNet 2.0 mostrando la estructura apilada y los refinamientos iterativos. Adaptado de [13].

FlowNet 2.0 logra reducir el error de estimación en más de un 50 % en comparación con su predecesor, alcanzando un rendimiento comparable al de los métodos más avanzados de su época, mientras mantiene velocidades de procesamiento que permiten su uso en aplicaciones interactivas [13]. Estas mejoras consolidan a FlowNet 2.0 como una herramienta poderosa y eficiente para la estimación de flujo óptico en aplicaciones del mundo real, como la segmentación de movimiento y el reconocimiento de acciones.

2.3.5. LiteFlowNet

LiteFlowNet surge como una alternativa a FlowNet, una red neuronal convolucional (CNN) que demanda más de 160 millones de parámetros para lograr una estimación precisa del flujo óptico. LiteFlowNet presenta un enfoque innovador que consigue un rendimiento similar al de FlowNet en conjuntos de datos desafiantes como Sintel y KITTI, mientras reduce significativamente el tamaño del modelo, siendo 30 veces más pequeño, y aumenta la velocidad de ejecución.

Las características clave de LiteFlowNet incluyen:

1. **Enfoque de Inferencia de Flujo Más Efectivo:** LiteFlowNet introduce un enfoque más eficiente para la inferencia de flujo en cada nivel de la pirámide mediante una red en cascada ligera. Este enfoque no solo mejora la precisión de la estimación

de flujo al realizar correcciones tempranas, sino que también facilita la integración fluida de la coincidencia que describen la red, permitiendo a LiteFlowNet mantener un alto rendimiento.

2. **Capa de Regularización de Flujo Innovadora:** LiteFlowNet presenta una nueva capa de regularización de flujo que aborda los problemas de valores atípicos y fronteras de flujo indefinidas mediante el uso de una convolución local impulsada por características. Esta capa es esencial para mantener la coherencia y precisión en la estimación del flujo en regiones complejas de las imágenes, como las áreas con oclusiones o movimientos rápidos.
3. **Extracción de Características Piramidales Efectiva:** LiteFlowNet emplea una estructura eficaz para la extracción de características piramidales y utiliza el deformado de características en lugar del deformado de imágenes, como en FlowNet. Este enfoque permite a la red mantener la integridad de la información visual durante el procesamiento, mejorando la precisión y estabilidad de la estimación del flujo óptico. Además, la arquitectura piramidal permite a LiteFlowNet capturar movimientos a diferentes escalas, lo que es especialmente útil en escenarios con variaciones de movimiento tanto grandes como pequeñas.
4. **Eficiencia Computacional y Aplicaciones en Tiempo Real:** Uno de los aspectos más destacados de LiteFlowNet es su eficiencia computacional. Gracias a su diseño optimizado, puede ejecutarse de manera más rápida y con menos recursos en comparación con modelos más grandes como FlowNet, lo que la convierte en una opción ideal para aplicaciones en tiempo real y en dispositivos con recursos limitados. Esto amplía su aplicabilidad a una variedad de entornos industriales y móviles, donde el equilibrio entre precisión y eficiencia es fundamental.

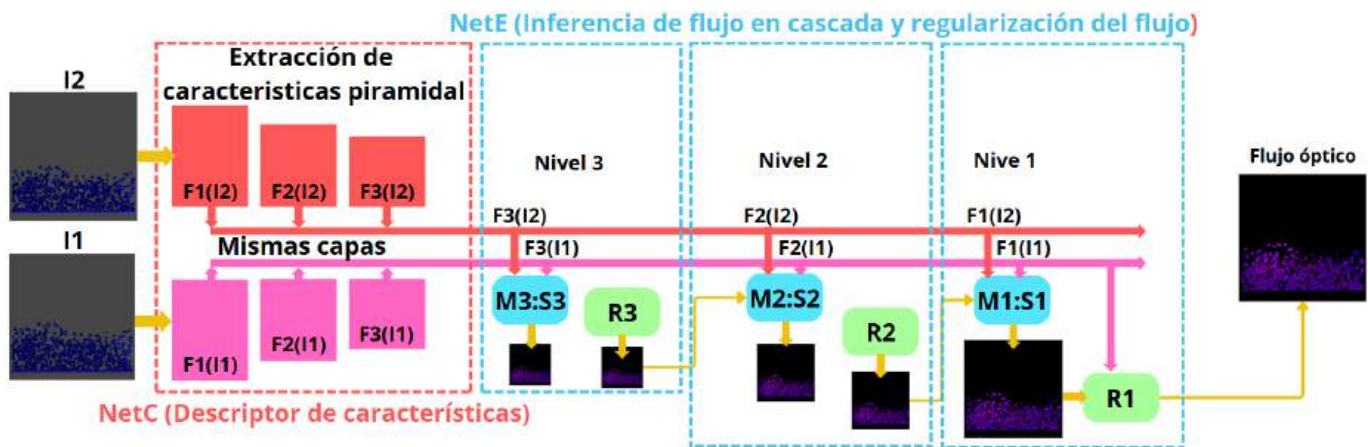


Figura 2.16: Esquema de la arquitectura de LiteFlowNet destacando su diseño optimizado y la estructura en cascada para la inferencia de flujo. Adaptado de [14].

En conjunto, LiteFlowNet representa una combinación única de eficiencia y rendimiento, lo que la convierte en una opción particularmente atractiva para aplicaciones en tiempo real y entornos con recursos limitados. Su diseño compacto y eficiente, junto con su capacidad para mantener un buen nivel de precisión, la posicionan como una herramienta poderosa en el campo de la estimación de flujo óptico [14].

2.3.6. PWC-Net

PWC-Net es una solución compacta pero altamente efectiva basada en redes neuronales convolucionales (CNN) para la estimación de flujo óptico. Su diseño se fundamenta en principios bien establecidos: procesamiento piramidal, deformación y la utilización de un volumen de coste.

En su formulación, PWC-Net implementa una pirámide de características que puede ser aprendida, aprovechando la estimación preliminar del flujo óptico para deformar las características de la segunda imagen procesadas por la CNN. A partir de estas características deformadas y las características de la primera imagen, se construye un volumen de coste que encapsula información crucial para la estimación del flujo óptico. Este volumen de coste se somete a procesamiento adicional mediante una CNN, que finalmente genera la estimación precisa del flujo óptico.

Una de las principales innovaciones de PWC-Net es su enfoque en la pirámide de características y el desenfoco. La pirámide permite trabajar con múltiples escalas, lo que facilita la captura de movimientos de diferentes magnitudes y mejora la robustez de la estimación. La deformación aplicada a las características de la segunda imagen en la pirámide utiliza la estimación preliminar del flujo para alinear mejor las características correspondientes entre ambas imágenes, reduciendo así la complejidad del problema de estimación del flujo óptico.

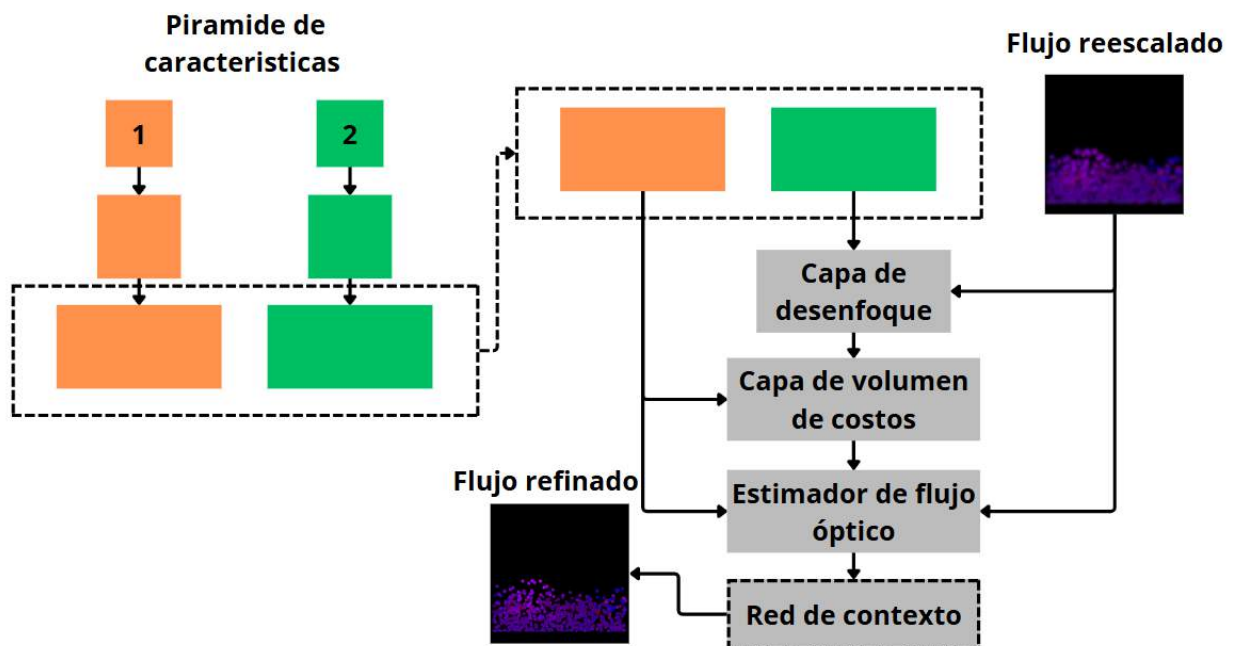


Figura 2.17: Esquema de la arquitectura de PWC-Net, destacando el uso de pirámides de características y la construcción del volumen de coste para la estimación de flujo óptico. Adaptado de [15].

El volumen de coste, generado a partir de las características deformadas y las originales, encapsula información esencial sobre las correspondencias entre los píxeles de ambas imágenes. Este volumen de coste es procesado por una CNN que refina progresivamente la estimación del flujo óptico, aprovechando tanto la estructura local como global de las características. Este enfoque de refinamiento iterativo permite a PWC-Net capturar detalles finos y mejorar la precisión del flujo estimado.

Una de las ventajas más destacadas de PWC-Net es su eficiencia en términos de tamaño y entrenamiento. En comparación con modelos como FlowNet, PWC-Net es significativa-

mente más compacto, con una complejidad 17 veces menor, lo que lo hace más manejable y menos costoso en términos computacionales. En cuanto a rendimiento, supera a todos los métodos de flujo óptico publicados en los benchmarks MPI Sintel final pass y KITTI 2015.

En entornos prácticos, PWC-Net demuestra su capacidad de ejecutarse a una velocidad notable, alcanzando aproximadamente 35 fotogramas por segundo en imágenes con resolución Sintel. Esta combinación de eficiencia, precisión y velocidad lo convierte en una opción atractiva para una amplia gama de aplicaciones en visión por computadora, tales como la navegación autónoma, la videovigilancia y la manipulación robótica, donde la estimación rápida y precisa del movimiento es crucial [15].

PWC-Net también ha influido en el desarrollo de arquitecturas subsecuentes para flujo óptico, como IRR y MaskFlownet, que han adoptado y extendido algunos de sus conceptos fundamentales, particularmente en el uso de pirámides de características y volúmenes de coste, mostrando la relevancia e impacto de esta arquitectura en la investigación continua del flujo óptico.

2.3.7. Refinamiento residual iterativo (IRR)

IRR se destaca por su enfoque iterativo en la refinación de estimaciones de flujo óptico. Inspirada en los métodos clásicos de minimización de energía, IRR aplica un esquema de refinamiento residual iterativo basado en la reutilización de pesos. Este enfoque no solo mejora la precisión en la estimación del flujo óptico, sino que también reduce significativamente la cantidad de parámetros del modelo, haciéndolo más eficiente en términos computacionales.

IRR se compone de los siguientes componentes principales:

1. **Refinamiento Residual Iterativo:** Este componente central toma la salida de una iteración previa como entrada y la refina de manera iterativa utilizando un único bloque de red neuronal con pesos compartidos. De este modo, se incrementa la precisión sin añadir parámetros adicionales.

2. **Estimación de Oclusiones:** IRR integra la estimación de oclusiones para mejorar la exactitud del flujo óptico mediante la predicción bidireccional del flujo y la identificación de regiones ocluidas. Esto permite una representación más precisa de las áreas complejas en las imágenes.
3. **Capas de Refinamiento Bilateral y de Upsampling:** Para abordar los problemas de desenfoco cerca de los bordes de movimiento, se utilizan filtros bilaterales aprendidos, que refinan tanto el flujo óptico como las oclusiones. Además, una capa de upsampling mejora la resolución de las oclusiones, permitiendo una mejor representación de los detalles finos.

La Figura 2.18 muestra una representación visual del diseño de IRR, destacando su proceso iterativo y los componentes clave que contribuyen a su alto rendimiento.

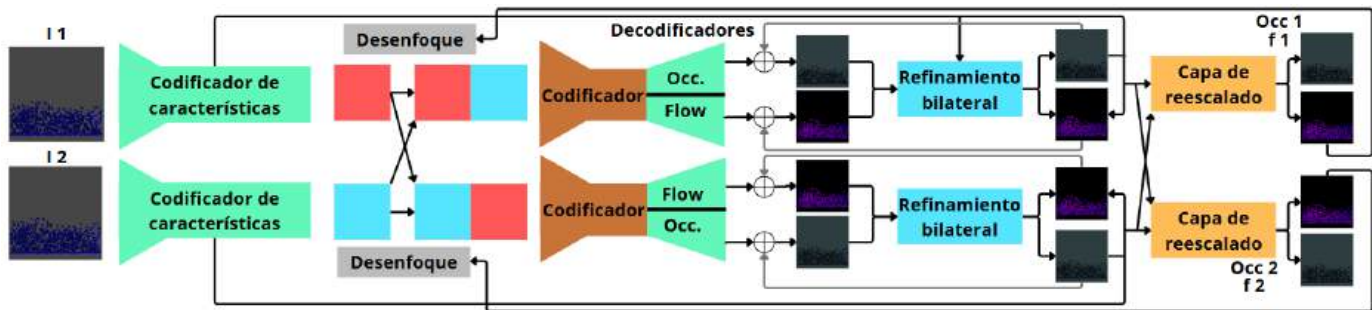


Figura 2.18: Diagrama esquemático de IRR: (1) Refinamiento residual iterativo que reutiliza el mismo bloque de red en múltiples iteraciones para mejorar las estimaciones de flujo. (2) Estimación de oclusiones integrada para mejorar la precisión general. (3) Capas de refinamiento bilateral y reescalado que mejoran la definición en los bordes de movimiento y oclusiones. Adaptado de [16].

IRR ha demostrado una notable mejora en la precisión del flujo óptico en diversos conjuntos de datos, superando a otros métodos con un menor número de parámetros. En pruebas realizadas en el conjunto de datos Sintel, IRR alcanzó un error de punto final de 3.32, lo que supone una mejora del 10 % en comparación con otros modelos de referencia. Además, en el conjunto de datos KITTI, logró una reducción del 20 % en el error de estimación con un 26 % menos de parámetros que el modelo base PWC-Net [16]. Esta combinación de precisión y eficiencia convierte a IRR en una arquitectura altamente competitiva para la estimación de flujo óptico.

2.3.8. MaskFlowNet

MaskFlowNet es una arquitectura avanzada para la estimación de flujo óptico que se destaca por su manejo eficiente de áreas ocluidas en secuencias de imágenes. Inspirada en la arquitectura PWC-Net [15], MaskFlowNet introduce el módulo *Asymmetric Occlusion-Aware Feature Matching* (AsymOFMM), que incorpora una máscara de oclusión aprendible. Este módulo tiene la capacidad de filtrar áreas ocluidas inmediatamente después de la deformación de características, mejorando significativamente la precisión del flujo óptico sin necesidad de supervisión explícita [17].

MaskFlowNet se estructura en dos etapas principales:

1. **MaskFlowNet-S:** La primera etapa, MaskFlowNet-S, hereda la arquitectura de PWC-Net, sustituyendo los módulos tradicionales de correspondencia de características por AsymOFMM. Este módulo permite generar una pirámide de características compartida a seis niveles, realizando predicciones de flujo óptico de manera iterativa, refinando el resultado desde un nivel grueso hasta uno más detallado. La introducción de AsymOFMM es clave para mitigar los efectos negativos de las oclusiones y mejorar la precisión general de la estimación [17].
2. **MaskFlowNet Completo:** La segunda etapa de la arquitectura es una extensión de MaskFlowNet-S, donde se introduce una pirámide de características consciente de las oclusiones. Esta etapa adicional permite combinar las pirámides de características originales con las nuevas pirámides conscientes de oclusiones, proporcionando refinamientos adicionales en la predicción del flujo óptico, especialmente en áreas complejas con grandes desplazamientos y múltiples oclusiones [17].

La arquitectura de MaskFlowNet, que se ilustra en la Figura 2.19, destaca por la integración del módulo AsymOFMM y las pirámides de características conscientes de oclusiones. Estos elementos son fundamentales para abordar los desafíos inherentes a la estimación precisa del flujo óptico en presencia de áreas ocluidas.

MaskFlowNet ha demostrado ser efectiva, superando a otros métodos de estimación de

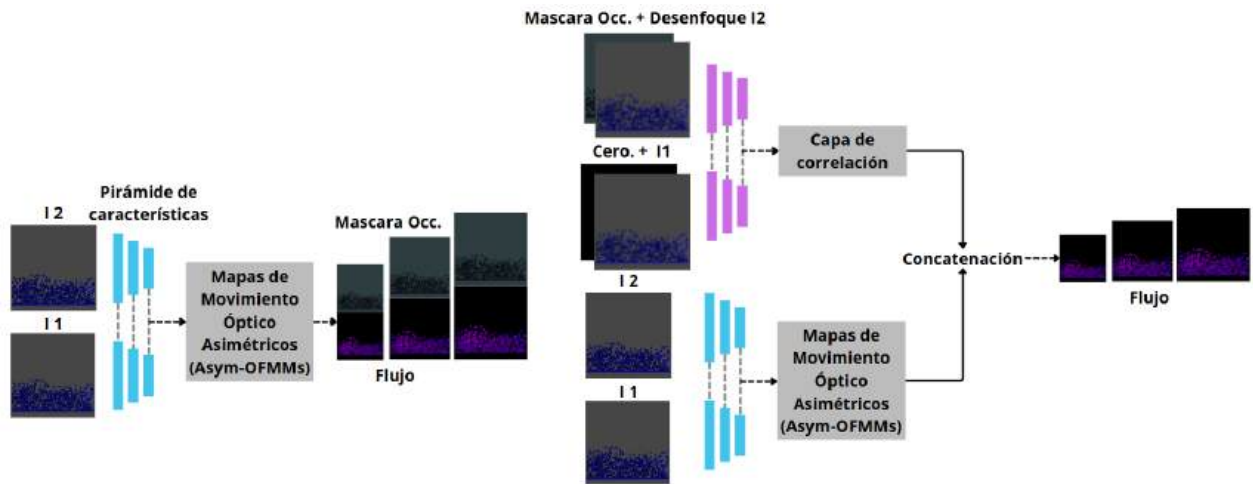


Figura 2.19: Esquema de la arquitectura de MaskFlowNet, destacando el módulo Asym-OFMM y la integración de pirámides de características conscientes de oclusiones para la estimación de flujo óptico. Adaptado de [17].

flujo óptico en benchmarks reconocidos como MPI Sintel y KITTI, estableciendo nuevos estándares en cuanto a precisión [48]. Además, la eficiencia computacional de MaskFlowNet la convierte en una solución práctica y robusta para aplicaciones en tiempo real, un aspecto crucial en campos como la visión por computadora y la robótica.

Es relevante destacar que MaskFlowNet comparte ciertos principios con la arquitectura IRR [6], que también busca mejorar la precisión del flujo óptico mediante refinamientos iterativos y la gestión de oclusiones. Sin embargo, IRR se basa en un esquema de refinamiento residual iterativo con reutilización de pesos, mientras que MaskFlowNet introduce mejoras significativas en la gestión de oclusiones mediante máscaras aprendibles y pirámides de características duales. Ambos enfoques subrayan la importancia de integrar la estimación de oclusiones para mejorar la precisión general en la estimación del flujo óptico, aunque difieren en los métodos y arquitecturas implementadas para alcanzar este objetivo.

2.3.9. Agregación Global de Movimiento (GMA)

GMA es una arquitectura avanzada de aprendizaje profundo diseñada para mejorar la precisión en la estimación de flujo óptico, especialmente en escenarios donde las oclusiones

y las variaciones de movimiento complejas presentan desafíos significativos.

Componentes y Estructura de GMA

1. **Agregación de Movimiento Global (GMA):** El núcleo de esta arquitectura es el módulo de Agregación de Movimiento Global, que utiliza mecanismos de atención inspirados en las redes transformer. Este módulo permite integrar información de diversas regiones de la imagen, compensando áreas donde las características de movimiento son más difíciles de estimar debido a la presencia de oclusiones o variaciones significativas en el flujo.
2. **Módulo de Auto-similitud:** GMA emplea un módulo de auto-similitud que compara la información de las regiones no ocluidas para inferir los flujos en áreas afectadas. Esta capacidad es crucial para mejorar la precisión en la estimación de flujos en áreas con variabilidad de movimiento o donde las oclusiones dificultan la medición directa.
3. **Optimización mediante Volumen de Correlación:** La red utiliza un volumen de correlación 4D, que encapsula las relaciones de movimiento entre píxeles, y un codificador de características de movimiento 2D que optimiza la extracción de información relevante para la estimación del flujo óptico. Estos pasos permiten que la red refuerce sus predicciones y mejore su precisión.
4. **Refinamiento Recurrente:** Tras la agregación de características de movimiento global, GMA concatena las características obtenidas y las introduce en una unidad recurrente GRU, que refina iterativamente la estimación del flujo, ajustando la salida de forma precisa en cada iteración.

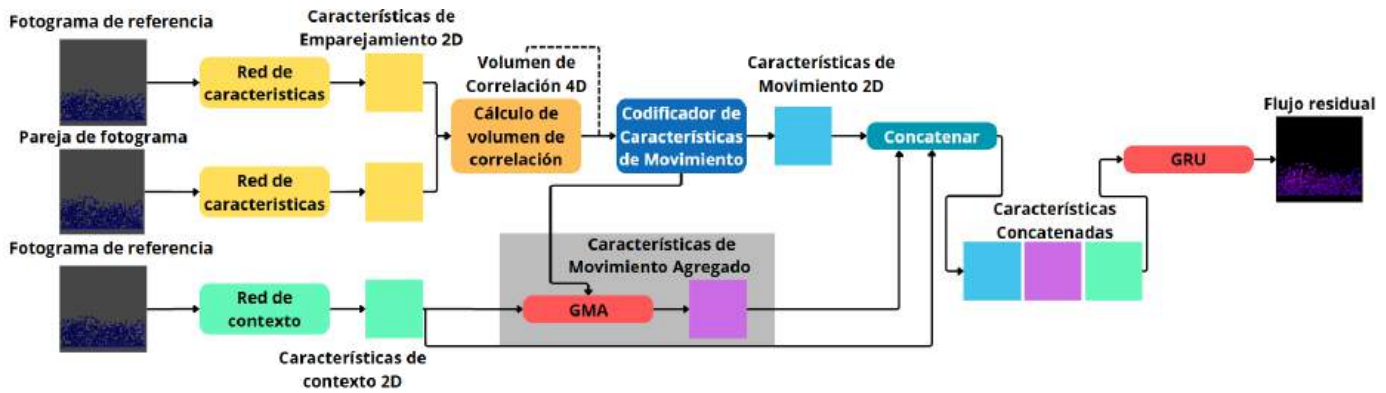


Figura 2.20: Diagrama de la arquitectura GMA. La red toma dos imágenes de referencia y coincidencia, pasa ambas por una red de características que genera un volumen de correlación 4D. Posteriormente, un codificador de características de movimiento extrae características 2D del movimiento, las cuales son refinadas mediante el módulo GMA, que agrega información de movimiento global. Finalmente, las características concatenadas se introducen en una unidad GRU para producir un flujo residual refinado. Adaptado de [18].

Esta estructura permite que GMA supere las limitaciones de otros modelos en la estimación de flujos complejos, mejorando tanto la precisión como la robustez en la predicción de flujo óptico en entornos con oclusiones y variaciones de movimiento significativas.

En este capítulo se describe la metodología empleada para el desarrollo y la evaluación de la arquitectura propuesta, Quetzal-Flow. Se presentan los detalles del diseño del modelo, la preparación de los datos, y la configuración del entrenamiento, incluyendo hiperparámetros y recursos computacionales. Además, se explican las métricas utilizadas para medir el desempeño del modelo y las configuraciones específicas de las pruebas realizadas. Por último, se mencionan las limitaciones encontradas durante el desarrollo y las estrategias implementadas para resolverlas.

3.1. Desarrollo del proyecto

El desarrollo de este proyecto involucró varias etapas, que se detallan en la Figura 3.1. A continuación se presentan las principales fases del trabajo, cada una diseñada para abordar un objetivo específico:

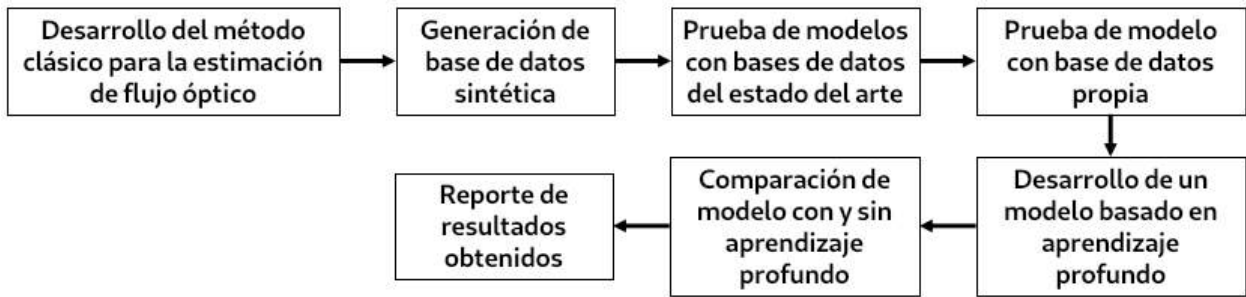


Figura 3.1: Diagrama de flujo de la metodología para el proyecto. El proceso incluye el desarrollo de un método clásico, la generación de una base de datos sintética, y pruebas de modelos utilizando bases de datos del estado del arte y una base de datos propia. A continuación, se desarrolla un modelo basado en aprendizaje profundo, seguido de una comparación de desempeño entre modelos con y sin aprendizaje profundo. Finalmente, se elabora el reporte de resultados obtenidos.

1. **Desarrollo del método clásico para la estimación de flujo óptico:** Implementación de Gunnar Farneback como referencia.
2. **Generación de base de datos sintética:** Creación de un conjunto de datos personalizado utilizando simulaciones controladas, para proporcionar un entorno de entrenamiento adecuado para los modelos.
3. **Prueba de modelos con bases de datos del estado del arte:** Evaluación inicial con bases de datos estándar para establecer un punto de comparación y estudio de las redes del estado del arte.
4. **Prueba de modelo con base de datos propia:** Evaluación del rendimiento de modelo creado en la base de datos sintética creada.
5. **Desarrollo de un modelo basado en aprendizaje profundo:** Entrenamiento de una red neuronal para la estimación de flujo óptico.
6. **Comparación de modelo con y sin aprendizaje profundo:** Implementación de ambos modelos con imágenes reales de reactores de placas paralelas, evaluando con métricas como error cuadrático medio, distancia coseno y distancia de correlación.

3.2. Hardware y Software

Para garantizar un entorno de desarrollo adecuado y obtener resultados precisos y reproducibles, se emplearon diferentes configuraciones de hardware y software en cada fase del proyecto:

- **Generación de la base de datos sintética:** Se utilizó Blender, complementado con Python 3.10 para scripts adicionales. El sistema contaba con una tarjeta NVIDIA 4070 Ti con 12 GB de memoria, un procesador Ryzen 7 5800X y 32 GB de RAM.
- **Método clásico de estimación de flujo óptico:** Se implementó el algoritmo de Gunnar Farnebäck en Python 3.10. Se ejecutó en una computadora con 24 GB de RAM y un procesador Intel Core i5 de 8 núcleos.
- **Entrenamiento y prueba de redes neuronales:** Se utilizó un servidor con 4 GPUs V100-SXM2, 631 GB de RAM y procesador Power9. Los modelos RAFT, FlowNet, FlowNet2, LiteFlowNet, LiteFlowNet2, PWC-Net, IRR, MaskFlowNet y GMA, disponibles en la biblioteca mmflow [49], fueron implementados utilizando Python y PyTorch. Por otro lado, Quetzal-Flow, la red propuesta en este trabajo, no pertenece a mmflow y también fue implementada en el mismo entorno.

3.3. Desarrollo del método clásico para la estimación de flujo óptico

En esta subsección se describe la implementación del método clásico de flujo óptico de Gunnar Farnebäck, empleado como referencia inicial antes de aplicar técnicas basadas en aprendizaje profundo, el flujo que se siguió se puede observar en la Figura 3.2.

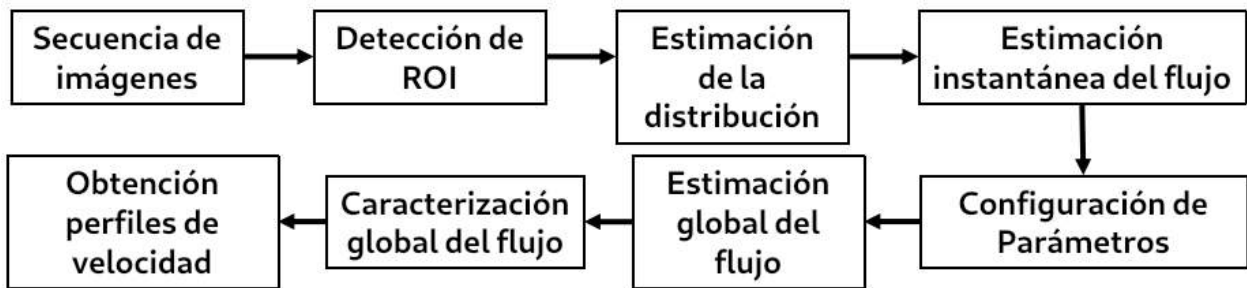


Figura 3.2: Diagrama de flujo del proceso de estimación y análisis del flujo óptico utilizando un modelo neuronal. El proceso abarca desde la detección de la región de interés (ROI) y la estimación de la distribución del flujo, hasta el diseño, entrenamiento y evaluación del modelo neuronal. Las flechas indican el flujo de las etapas, con íconos que destacan puntos de evaluación: rojo para corrección y verde para éxito.

3.3.1. Implementación del Método de Gunnar Farneback

La implementación del método de Gunnar Farneback se llevó a cabo utilizando la biblioteca OpenCV en Python, que proporciona una función optimizada para la aplicación de dicho algoritmo. La implementación se desarrolló a través de los siguientes pasos:

1. **Secuencia de imágenes:** Se procesaron secuencias de fotogramas consecutivos extraídos de videos capturados en un reactor de placas paralelas en CIDETEQ. Estas imágenes son flujos reales cuyo objetivo es analizar el desplazamiento del flujo a través del reactor.
2. **Detección ROI:** Se generó un algoritmo que permite seleccionar la zona de interés donde se desea realizar el análisis.
3. **Estimación de la distribución:** Se estimó la distribución en las imágenes para centrarla en el color azul, eliminando así artefactos no deseados y convirtiéndolas al espacio de escala de grises para facilitar el procesamiento. Esta conversión permite que el método de Farneback opere de manera más eficiente, ya que reduce la complejidad computacional al disminuir la cantidad de información a procesar. Como resultado, se acelera el cálculo y se optimiza el rendimiento del algoritmo, asegurando una mejor detección de patrones de movimiento.

4. Estimación instantánea del flujo:

Utilizando la función `cv2.calcOpticalFlowFarneback()`, se estimó el flujo óptico entre fotogramas consecutivos, como se muestra en la Figura 3.3(c). Esta implementación permite calcular el flujo óptico de manera eficiente, aunque el tiempo de procesamiento puede incrementarse en secuencias de alta resolución o con grandes volúmenes de datos. El resultado es un campo denso de vectores que describe el movimiento aparente de cada píxel entre las imágenes, proporcionando una representación detallada del comportamiento del flujo.

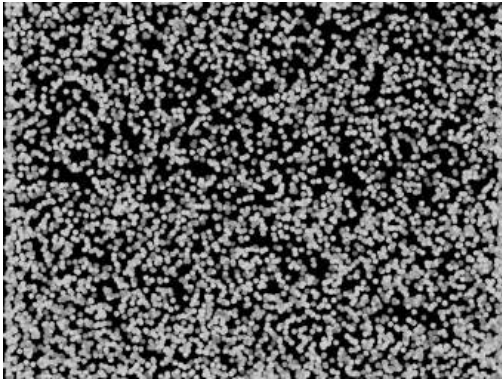
5. **Configuración de Parámetros:** Los parámetros del método de Farneback fueron ajustados empíricamente mediante un proceso iterativo en el que se probaban diferentes configuraciones y se evaluaba el rendimiento en términos de precisión del flujo y tiempo de cómputo. El equilibrio óptimo entre calidad y eficiencia se alcanzó al ajustar los niveles de la pirámide, el tamaño de la ventana y otros parámetros, evaluando su impacto en la estabilidad y la velocidad del algoritmo. Para la implementación sin GPU, se utilizó la función `cv2.calcOpticalFlowFarneback()` de OpenCV con los siguientes valores:

- `numLevels`: 4 (número de niveles en la pirámide).
- `pyrScale`: 0.5 (factor de escala entre niveles).
- `fastPyramids`: False (para una construcción de pirámide más precisa).
- `winSize`: 5 (tamaño de la ventana de búsqueda).
- `numIters`: 3 (número de iteraciones por nivel).
- `polyN`: 5 (tamaño del polinomio utilizado para la expansión).
- `polySigma`: 1.1 (desviación estándar para el filtro Gaussiano aplicado al polinomio).

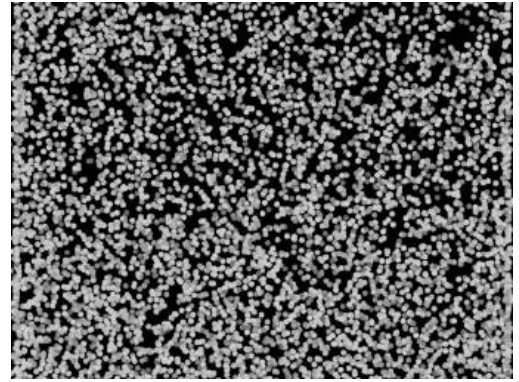
6. **Estimación global del flujo óptico:** Se consideraron las magnitudes de flujo óptico mayores en cada fotograma y se acumularon a lo largo de la secuencia de imágenes para obtener un perfil de velocidad global. Este proceso permitió identificar

las tendencias de movimiento predominantes y generar un perfil de velocidad que representa el comportamiento del flujo a través del tiempo. Al acumular las magnitudes mayores, se obtuvo una visión integral del flujo en la región de interés, lo cual es esencial para caracterizar patrones de movimiento y evaluar la estabilidad del sistema.

7. **Caracterización global del flujo:** Se utilizó el mapa de colores Jet para facilitar una interpretación visual intuitiva y amigable de los datos. Este esquema de color permite al usuario identificar rápidamente las variaciones en la magnitud, destacando las áreas de mayor y menor velocidad. La representación en colores facilita la comprensión de los patrones de movimiento y contribuye a un análisis más accesible y visualmente atractivo.
8. **Obtención de perfiles de velocidad:** Se digitalizó la imagen y se seleccionó la distancia de análisis para, posteriormente, graficar el perfil de velocidad y realizar la comparación con los resultados obtenidos a partir de la simulación CFD. Este proceso permitió evaluar la precisión de la estimación de flujo óptico en relación con el perfil de referencia generado por CFD.



(a) Frame 1



(b) Frame 2



(c) Flujo óptico

Figura 3.3: Flujo óptico estimado utilizando el método de Gunnar Farneback. En los incisos (a) y (b) se muestran los fotogramas de la base sintética creada, utilizados para calcular el flujo óptico; en (c) se presenta el flujo óptico calculado. Los colores representan la dirección y magnitud del flujo en cada píxel (Autoría propia).

Pseudocódigo del Método de Farneback

A continuación, se presenta el pseudocódigo del algoritmo de Farneback para la estimación de flujo óptico, con el objetivo de ilustrar la lógica detrás de cada paso del proceso. El método utiliza una pirámide Gaussiana para manejar grandes desplazamientos y un modelo polinomial para aproximar el movimiento en cada nivel de la pirámide.

Algorithm 1 Algoritmo de Flujo Óptico de Farneback (cv2.calcOpticalFlowFarneback)

Require: Imágenes en escala de grises I_1 e I_2 , parámetros del algoritmo (pyr_scale , $levels$, win_size , $iterations$, $poly_n$, $poly_sigma$, $flags$)

Ensure: Campo de flujo óptico Of con componentes de flujo horizontal u y vertical v

- 1: **Preprocesamiento:**
 - 2: Convertir imágenes I_1 e I_2 a escala de grises, si no lo están ya.
 - 3: Aplicar suavizado Gaussiano a I_1 e I_2 para reducir el ruido.
 - 4: **Construcción de pirámides Gaussianas:**
 - 5: Generar pirámides de imágenes para I_1 e I_2 con un factor de escala definido por pyr_scale .
 - 6: Dividir las imágenes en múltiples niveles de resolución ($levels$).
 - 7: **Estimación de modelos locales:**
 - 8: **for** cada píxel (x, y) en la imagen del nivel actual **do**
 - 9: Definir una vecindad N alrededor del píxel (x, y) con un tamaño definido por win_size .
 - 10: Ajustar un polinomio cuadrático en la vecindad de I_1 usando $poly_n$ y $poly_sigma$.
 - 11: Repetir el ajuste para la vecindad correspondiente en I_2 .
 - 12: Calcular el desplazamiento horizontal (u) y vertical (v) resolviendo $f_2(x, y) \approx f_1(x + u, y + v)$ utilizando mínimos cuadrados.
 - 13: Almacenar u y v en el campo de flujo Of para la posición (x, y) .
 - 14: **end for**
 - 15: **Interpolación de flujos:**
 - 16: **for** cada nivel de la pirámide desde el más grueso al más fino **do**
 - 17: Remuestrear I_2 utilizando el flujo estimado (u, v) del nivel anterior.
 - 18: Refinar la estimación del flujo Of aplicando transformaciones locales y ajustando u y v en cada iteración.
 - 19: **end for**
 - 20: **Refinamiento:**
 - 21: Realizar iteraciones de refinamiento ($iterations$) para mejorar la precisión del flujo.
 - 22:
 - 23: **Retornar** el campo de flujo óptico Of , con u y v en cada píxel (x, y) .
-

3.4. Generación de base de datos sintética

El desarrollo de una base de datos sintética personalizada fue necesario para abordar las limitaciones de los conjuntos de datos existentes, tales como la falta de escenarios específicos y controlados que representen de manera precisa el comportamiento de fluidos en condiciones experimentales. Esta base de datos permite entrenar redes neuronales para la estimación de flujo óptico en contextos particulares, simulando patrones de movimiento

de partículas que representan flujos de fluidos en ambientes industriales.

El objetivo principal de esta base de datos fue proporcionar un entorno de entrenamiento bien definido y con un *ground truth* preciso de los desplazamientos de las partículas. Para ello, se utilizó el software de modelado y simulación Blender, que, junto con el addon BlenderProc y scripts personalizados en Python, permitió generar de forma automatizada pares de imágenes con movimiento controlado y archivos de flujo óptico que describen el desplazamiento de cada píxel en las direcciones u y v .

Parámetros de Simulación

Los parámetros de simulación fueron seleccionados cuidadosamente para capturar una amplia gama de comportamientos de partículas en flujo, abarcando tanto movimientos lineales como trayectorias no lineales. En cada simulación se incluyeron de 9,000 a 10,000 esferas por escena, con variación en el tamaño y la velocidad para representar diferentes regímenes de flujo. Las trayectorias fueron modeladas para incluir tanto movimientos suaves como colisiones, lo que permitió generar flujos complejos y detallados.

El motor de físicas de Blender fue configurado para controlar propiedades como la gravedad, la fricción y la elasticidad de las esferas, influyendo directamente en el comportamiento del sistema simulado. Las escenas se renderizaron a una resolución de 512x384 píxeles, con dos imágenes generadas para cada simulación: una correspondiente al estado inicial (*Frame 1*) y otra al estado final (*Frame 2*). Además, se generó un archivo en formato `.flo` que almacena los desplazamientos exactos en las direcciones u y v , proporcionando así la "verdad absoluta" (*ground truth*) necesaria para el entrenamiento de los modelos de flujo óptico.

Preprocesamiento de Datos

Antes de ser utilizados para el entrenamiento de los modelos, los datos generados pasaron por un proceso de preprocesamiento similar al utilizado en conjuntos de datos como FlyingChairs. Este proceso incluyó varios pasos clave para asegurar la robustez del con-

junto de datos y mejorar la capacidad de generalización de las redes neuronales:

- **Normalización de imágenes:** Los valores de los píxeles en las imágenes fueron escalados a un rango de $[0, 1]$, asegurando que las entradas al modelo estuvieran dentro de un intervalo adecuado para el entrenamiento de redes neuronales profundas.
- **Normalización de vectores de desplazamiento:** Los vectores de desplazamiento en las direcciones u y v almacenados en los archivos `.flo` también fueron normalizados para que el modelo pudiera aprender de manera efectiva la magnitud y dirección del movimiento.
- **Aumento de datos:** Se aplicaron técnicas de aumento de datos, tales como rotaciones aleatorias y escalados, con el fin de incrementar la diversidad del conjunto de entrenamiento y evitar el sobreajuste. Estas transformaciones permitieron simular diferentes ángulos y tamaños de las esferas en movimiento, mejorando la capacidad del modelo para generalizar a nuevas condiciones.

El preprocesamiento de los datos es fundamental para preparar la base de datos sintética, garantizando que los modelos puedan aprender de manera eficiente y ser robustos frente a variaciones no vistas durante el entrenamiento.

Configuración de las Partículas

El conjunto de datos sintético se diseñó para representar el movimiento de partículas distribuidas en un espacio tridimensional, emulando el comportamiento de flujos de fluidos en un entorno cerrado. Para cada escena generada, se definieron las siguientes configuraciones:

- **Número de partículas:** Se emplearon entre 9,000 y 10,000 partículas por escena, distribuidas de manera uniforme dentro del espacio de simulación para proporcionar una densidad adecuada de elementos en movimiento.

- **Tamaño de partículas:** El tamaño de cada esfera varió entre 1 y 3 píxeles en la imagen renderizada, lo que permitió capturar diferentes niveles de detalle en los desplazamientos. Esta variación es crucial para entrenar a las redes neuronales en la detección de movimientos sutiles y cambios pequeños en el entorno.
- **Velocidad y dirección:** Las velocidades de las partículas fueron seleccionadas aleatoriamente dentro de un rango controlado para generar una variedad de comportamientos. Además, se incluyeron movimientos tanto lineales como no lineales, lo que aumentó la complejidad de las trayectorias y ayudó a que los modelos aprendieran a manejar flujos irregulares.
- **Interacciones de colisión:** Para simular colisiones entre partículas, se activaron propiedades de física como fricción y elasticidad. La fricción se configuró en un rango de 0.1 a 0.3, mientras que la elasticidad varió entre 0.7 y 0.9. Estos ajustes permitieron generar interacciones realistas entre las partículas, replicando comportamientos observados en flujos de fluidos reales.

Propiedades del Entorno

El entorno de simulación se configuró para representar un espacio cerrado con características homogéneas, que permitiera capturar de manera precisa el movimiento de las partículas sin interferencias externas. Entre las propiedades del entorno se encuentran:

- **Tamaño del espacio de simulación:** El espacio de simulación se estableció con un límite cúbico de 10x10x10 unidades en Blender, lo que aseguró que las partículas se mantuvieran dentro del rango visible de la cámara y que el movimiento se registrara completamente en cada par de imágenes generado.
- **Iluminación y sombras:** La configuración de iluminación fue homogénea, evitando variaciones significativas en la luminosidad entre fotogramas. Esto aseguró que los cambios observados en las imágenes se debieran únicamente al desplazamiento de las partículas y no a factores externos de iluminación.

- **Cámara fija:** La cámara se mantuvo estática durante toda la simulación para evitar la introducción de movimientos adicionales que pudieran afectar la precisión de los vectores de desplazamiento generados. La posición de la cámara se estableció de manera que cada partícula fuera visible y su movimiento pudiera ser capturado con precisión.

Generación de Secuencias

Cada simulación generó un par de imágenes que representan el estado inicial (`_img1.png`) y el estado final (`_img2.png`) del movimiento de las partículas. Adicionalmente, se creó un archivo en formato `.flo` que almacena los desplazamientos de cada píxel en las direcciones u y v . La generación de estos archivos se realizó siguiendo un protocolo automatizado, lo que permitió asegurar la consistencia en la estructura de los datos generados y facilitó su integración en el entrenamiento de modelos de aprendizaje profundo.

Los desplazamientos en u y v se calcularon con alta precisión, permitiendo que los archivos `.flo` generados sirvan como *ground truth* para la evaluación de las redes neuronales. Este enfoque asegura que los modelos tengan acceso a una representación precisa de los movimientos reales, mejorando la capacidad de las redes para aprender a predecir el flujo óptico con alta exactitud.

Rendimiento y Tiempo de Simulación

El proceso de generación de la base de datos se realizó utilizando hardware avanzado, lo que permitió llevar a cabo simulaciones detalladas y precisas. La computadora empleada contaba con una tarjeta gráfica NVIDIA 4070 Ti con 12 GB de memoria dedicada, un procesador Ryzen 7 5800X y 32 GB de memoria RAM. Esta configuración fue clave para simular de manera eficiente y realista los experimentos, generando cada par de imágenes en aproximadamente 5 segundos, incluyendo el cálculo de trayectorias y la generación del archivo `.flo`.

El diseño de cada ambiente, junto con la simulación del movimiento de partículas y el ren-

derizado, tomó aproximadamente 4 horas por caso. Este enfoque permitió recrear escenarios complejos con un alto nivel de detalle y fidelidad, asegurando que los datos obtenidos fueran representativos y útiles para el entrenamiento de redes neuronales especializadas en la estimación de flujo óptico.

La viabilidad del proceso radica en la capacidad de Blender para modelar y simular condiciones experimentales con un alto grado de control y precisión, permitiendo generar bases de datos sintéticas robustas y confiables. Esto garantiza datos consistentes y detallados, fundamentales para la evaluación y mejora de algoritmos en contextos complejos.

3.5. Prueba de modelos con bases de datos del estado del arte

En esta etapa del trabajo, se evaluaron múltiples arquitecturas avanzadas de flujo óptico utilizando bases de datos del estado del arte ampliamente reconocidas en la comunidad científica. Las redes analizadas incluyen FlowNet, FlowNet2, PWC-Net, LiteFlowNet2, IRR-PWC, MaskFlowNet, RAFT y GMA. Estas arquitecturas fueron seleccionadas por su relevancia en la literatura y su diseño específico para la tarea de estimación de flujo óptico.

Los modelos fueron entrenados y evaluados con distintas combinaciones de conjuntos de datos, como FlyingChairs, FlyingThings3D, Sintel, KITTI2012, KITTI2015 y HD1K. Cada conjunto de datos permitió analizar el rendimiento de las redes en escenarios variados, desde configuraciones controladas hasta entornos más complejos. Las pruebas incluyeron configuraciones específicas de hiperparámetros para garantizar que los resultados fueran representativos del desempeño típico de cada modelo.

Para evaluar el rendimiento, se emplearon métricas estándar como el Error de Punto Final (**EPE**) y el porcentaje de píxeles con error (**Fl-all**). Estas métricas proporcionaron una base cuantitativa para comparar la precisión y robustez de las redes bajo diferentes configuraciones de entrenamiento y validación. Adicionalmente, se analizó la eficiencia computacional de cada modelo, considerando los tiempos de entrenamiento e inferencia.

Los resultados obtenidos en esta etapa permitieron identificar patrones de comportamiento entre las diferentes arquitecturas, destacando las ventajas de los modelos más recien-

tes, como RAFT y GMA, en comparación con redes más antiguas, como FlowNet. Este análisis sirvió como base para establecer una comparación fundamentada entre las arquitecturas probadas y comprender su aplicabilidad en diferentes escenarios y de esta forma generar ideas para el desarrollo de la red propia.

El entendimiento profundo de estas diferencias no solo permitió evaluar el desempeño de cada modelo bajo diversas condiciones, sino también generó ideas clave para el diseño de *Quetzal-Flow*. A través de este proceso, se identificaron componentes y estrategias arquitectónicas relevantes, como el uso de correlación, generar múltiples escalas y actualización iterativa.

3.5.1. Configuraciones de Entrenamiento

Las redes se entrenaron con distintas combinaciones de conjuntos de datos, incluyendo FlyingChairs, FlyingThings3D, Sintel, KITTI2012, KITTI2015 y HD1K. Algunas configuraciones incluyeron etapas de *fine-tuning* (**ft**) en conjuntos de datos específicos para mejorar el rendimiento en escenarios más complejos.

A continuación, se presentan las combinaciones principales utilizadas en los entrenamientos:

1. **FlyingChairs**: Dataset sintético diseñado para escenarios controlados, principalmente basado en imágenes de sillas en movimiento sobre fondos variados.
2. **FlyingThing3D**: Dataset tridimensional que simula objetos complejos en movimiento, introduciendo desafíos como oclusión y diferentes perspectivas.
3. **ChairsSDHom**: Dataset sintético que enfoca el movimiento homogéneo de objetos (principalmente sillas) para el entrenamiento de modelos en escenarios simplificados con transformaciones estructurales.
4. **FlyingChairs + Sintel**: Combinación que introduce entornos más complejos y realistas, con variaciones de iluminación y escenas animadas inspiradas en la película Sintel.

5. **FlyingChairs + FlyingThings3D**: Dataset tridimensional con movimientos complejos y objetos dinámicos, enriquecido con mayor diversidad y realismo.
6. **FlyingChairs + FlyingThings3D + ChairsSDHom**: Combinación de datasets sintéticos que une FlyingChairs, FlyingThings3D y ChairsSDHom, permitiendo entrenar modelos en un rango diverso de movimientos homogéneos, tridimensionales y transformaciones estructurales.
7. **Mixed Dataset**: Conjunto que incluye FlyingChairs, FlyingThings3D, Sintel, KITTI2015 y HD1K para una mayor generalización, abarcando escenarios desde sintéticos hasta reales con diferentes tipos de movimiento.

3.6. Resultados de las Redes de Flujo Óptico

En esta sección se presentan los resultados obtenidos para cada red de flujo óptico evaluada, organizados por arquitectura. Cada subsección incluye los resultados obtenidos en diversas configuraciones de entrenamiento y validación. Al final, se presenta una comparación global para facilitar el análisis comparativo entre las redes.

3.6.1. FlowNet

La arquitectura **FlowNet** marcó un punto de partida significativo en la estimación de flujo óptico utilizando redes neuronales profundas. Diseñada inicialmente para procesar datos sintéticos, esta red implementa una arquitectura de tipo *encoder-decoder*, permitiendo la extracción de características y la predicción de mapas de flujo óptico. Las variantes de FlowNet, como **FlowNetC** y **FlowNetS**, incluyen modificaciones específicas para mejorar la precisión en escenarios controlados y complejos.

En la Tabla 3.1 se presentan los resultados obtenidos con FlowNet y sus variantes bajo diferentes configuraciones de entrenamiento. Los conjuntos de datos utilizados incluyen **FlyingChairs**, **Sintel (training)** y **KITTI2012**, permitiendo evaluar su rendimiento en escenarios variados.

Models	Training datasets	FlyingChairs	Sintel (training)		KITTI2012 (EPE)
			clean	final	
FlowNetC	1	1.78	3.6	4.93	7.55
FlowNetC	5	2.57	2.74	4.52	5.42
FlowNetC+ft	4	2.8	1.73	2.09	4.78
FlowNetS	1	2.03	4.25	5.64	8.81
FlowNetS+ft	4	3.06	1.93	2.12	6.83

Tabla 3.1: Resultados de FlowNet: Métricas obtenidas en diversos conjuntos de datos. Las métricas incluyen el EPE, evaluado en los conjuntos FlyingChairs, Sintel (entrenamiento limpio y final) y KITTI2012. Los valores más bajos indican un mejor desempeño del modelo.

Rendimiento en FlyingChairs: FlowNetC y FlowNetS muestran un **EPE** de 1.78 y 2.03, respectivamente, reflejando un desempeño sólido en un dataset sintético diseñado para escenarios controlados. El *fine-tuning* (**ft**) sobre otros datasets mejora significativamente los resultados, especialmente en FlowNetC+ft, con un **EPE** de 2.8.

Rendimiento en Sintel (training): FlowNetC logra un **EPE** de 3.6 (**clean**) y 4.93 (**final**) en el dataset Sintel. Sin embargo, tras el *fine-tuning*, FlowNetC+ft mejora a 1.73 (**clean**) y 2.09 (**final**). La variante FlowNetS muestra resultados comparables pero ligeramente inferiores, con un **EPE** de 4.25 (**clean**) y 5.64 (**final**).

Rendimiento en KITTI2012: En KITTI2012, FlowNetC alcanza un **EPE** de 7.55, mientras que FlowNetS registra un valor mayor, de 8.81. Las versiones con *fine-tuning* (e.g., FlowNetC+ft y FlowNetS+ft) destacan al reducir el error, con FlowNetS+ft alcanzando un **EPE** de 6.83, el mejor resultado en este dataset para la arquitectura FlowNet.

3.6.2. FlowNet2

FlowNet2 es una evolución de la arquitectura original FlowNet, diseñada para mejorar la estimación de flujo óptico en escenarios más complejos. Esta red refina progresivamente las predicciones del flujo, aumentando la precisión en comparación con su predecesor. Entre sus variantes se incluyen FlowNet2CS, FlowNet2CSS, y FlowNet2CSS-sd, que introducen configuraciones específicas para optimizar el rendimiento en distintos conjuntos de datos.

A continuación, se presentan los resultados obtenidos para FlowNet2 y sus variantes, organizados por los conjuntos de datos de entrenamiento y validación empleados. Los resultados abarcan datasets como **FlyingChairs**, **Sintel (training)**, **KITTI2012** y **KITTI2015**, además de combinaciones con **ChairsSDHom**.

Models	Training datasets	FlyingChairs	Sintel (training)	
			clean	final
FlowNet2CS	1	1.59	-	-
FlowNet2CS	5	-	1.96	3.69
FlowNet2CSS	1	1.55	-	-
FlowNet2CSS	5	-	1.85	3.57
FlowNet2CSS-sd	6	-	1.81	3.69
FlowNet2	1	1.15	-	-
FlowNet2	2	-	1.78	3.31
FlowNet2sd	3	-	-	-

Models	KITTI2012 (EPE)	KITTI2015 (training)		ChairsSDHom EPE
		FI-all	EPE	
FlowNet2CS	-	-	-	-
FlowNet2CS	3.5	28.28 %	8.23	-
FlowNet2CSS	-	-	-	-
FlowNet2CSS	3.13	25.76 %	7.72	-
FlowNet2CSS-sd	2.98	25.66 %	7.99	-
FlowNet2	-	-	-	-
FlowNet2	3.02	25.18 %	8.02	-
FlowNet2sd	-	-	-	0.37

Tabla 3.2: Resultados de FlowNet2: Métricas obtenidas en diversos conjuntos de datos. Las métricas incluyen EPE, FI-all para KITTI2015, y ChairsSDHom. Los valores más bajos indican un mejor desempeño del modelo.

Rendimiento en FlyingChairs: Las variantes FlowNet2CS y FlowNet2CSS muestran valores de **EPE** de 1.59 y 1.55, respectivamente, al entrenarse en el dataset FlyingChairs (**1**). FlowNet2 alcanza su mejor valor en este conjunto con 1.15 en EPE.

Rendimiento en Sintel (training): En el conjunto Sintel, FlowNet2CS entrenado con FlyingChairs + FlyingThings3D (**5**) registra un **EPE** de 1.96 (**clean**) y 3.69 (**final**), mientras que FlowNet2CSS-sd logra 1.81 (**clean**) y 3.69 (**final**) al incluir ChairsSDHom (**6**) en el entrenamiento.

Rendimiento en KITTI2012 y KITTI2015: Para KITTI2012, FlowNet2CSS-sd alcanza un

EPE de 2.98, destacándose entre las variantes. En KITTI2015, los mejores resultados en **FI-all** (25.18 %) y **EPE** (7.99) corresponden a FlowNet2 y FlowNet2CSS-sd respectivamente.

3.6.3. PWC-Net

PWC-Net es una arquitectura optimizada para la estimación de flujo óptico que utiliza un enfoque basado en pirámides de características, warping (*deformación*) y procesamiento basado en costos. Esta estructura modular permite reducir significativamente los parámetros del modelo, mejorando la eficiencia computacional sin sacrificar precisión.

En la Tabla 3.3, se presentan los resultados obtenidos para PWC-Net y sus variantes. Las configuraciones de entrenamiento incluyeron datasets como **FlyingChairs**, **FlyingThings3D**, **Sintel**, y **KITTI**, además de combinaciones con **ChairsSDHom**.

Models	Training datasets	FlyingChairs	Sintel (training)	
			clean	final
PWC-Net	1	1.51	3.52	4.81
PWC-Net	5	-	2.26	3.79
PWC-Net-ft	4	-	1.5	2.06
PWC-Net-ft-final	4	-	1.82	1.78
PWC-Net-ft	6	-	-	-
PWC-Net+	7	-	1.9	2.39

Models	KITTI2012 (training) EPE	KITTI2015 (training)		ChairsSDHom EPE
		FI-all	EPE	
PWC-Net	-	-	-	-
PWC-Net	3.66	29.85 %	9.49	-
PWC-Net-ft	-	-	-	-
PWC-Net-ft-final	-	-	-	-
PWC-Net-ft	1.07	6.09 %	1.64	-
PWC-Net+	-	-	-	-

Tabla 3.3: Resultados de PWC-Net: Métricas obtenidas en diversos conjuntos de datos.

Rendimiento en FlyingChairs: El modelo base PWC-Net logra un **EPE** de 1.51 en FlyingChairs (**1**), mostrando un desempeño sólido en esta base de datos sintética.

Rendimiento en Sintel (training): Para Sintel (training), PWC-Net entrenado con FlyingChairs + FlyingThings3D (**5**) obtiene un **EPE** de 2.26 (**clean**) y 3.79 (**final**), y tras aplicar *fine-tuning* con Sintel (**4**), las variantes PWC-Net-ft y PWC-Net-ft-final mejoran los valores a 1.5 y 1.82 (**clean**), y 2.06 y 1.78 (**final**), respectivamente.

Rendimiento en KITTI: En KITTI2012 y KITTI2015, los resultados indican que la variante PWC-Net-ft entrenada con FlyingChairs + FlyingThings3D + ChairsSDHom (**6**) alcanza valores de **EPE** de 1.07 (KITTI2012) y 1.64 (KITTI2015). Además, se registra un **FI-all** del 6.09 % en KITTI2015.

3.6.4. LiteFlowNet

LiteFlowNet es una arquitectura diseñada para ser compacta y eficiente en la estimación de flujo óptico. Utiliza una estructura de redes neuronales ligeras, lo que reduce el número de parámetros y acelera tanto el entrenamiento como la inferencia. Esta arquitectura se presenta en varias variantes, que incluyen configuraciones optimizadas para diferentes escenarios y conjuntos de datos.

En la Tabla 3.4, se presentan los resultados obtenidos para LiteFlowNet y sus variantes bajo distintas configuraciones de entrenamiento. Los conjuntos de datos utilizados incluyen **FlyingChairs**, **FlyingThings3D**, **Sintel (training)**, **KITTI2012**, y **KITTI2015**, así como combinaciones con **ChairsSDHom**.

Models	Training datasets	FlyingChairs	Sintel (training)	
			clean	final
LiteFlowNet-pre-M6S6	1	4.43	-	-
LiteFlowNet-pre-M6S6R6	1	4.07	-	-
LiteFlowNet-pre-M5S5R5	1	2.98	-	-
LiteFlowNet-pre-M4S4R4	1	2.20	-	-
LiteFlowNet-pre-M3S3R3	1	1.71	-	-
LiteFlowNet-pre-M2S2R2	1	1.38	2.74	4.52
LiteFlowNet	5	-	2.47	4.3
LiteFlowNet-ft	4	-	1.47	2.06
LiteFlowNet-ft	6	-	-	-

Models	KITTI2012 (EPE)	KITTI2015 (training)		ChairsSDHom EPE
		FI-all	EPE	
LiteFlowNet-pre-M6S6	-	-	-	-
LiteFlowNet-pre-M6S6R6	-	-	-	-
LiteFlowNet-pre-M5S5R5	-	-	-	-
LiteFlowNet-pre-M4S4R4	-	-	-	-
LiteFlowNet-pre-M3S3R3	-	-	-	-
LiteFlowNet-pre-M2S2R2	6.49	37.99 %	15.41	-
LiteFlowNet	5.42	32.86 %	13.5	-
LiteFlowNet-ft	-	-	-	-
LiteFlowNet-ft	1.07	5.45 %	1.45	-

Tabla 3.4: Resultados de LiteFlowNet: Métricas obtenidas en diversos conjuntos de datos.

Rendimiento en FlyingChairs: Las configuraciones iniciales de LiteFlowNet (1) muestran una reducción progresiva en el **EPE**, comenzando con 4.43 (M6S6) y alcanzando un valor mínimo de 1.38 (M2S2R2).

Rendimiento en Sintel (training): En Sintel (training), las variantes LiteFlowNet-pre-M2S2R2 (1) y LiteFlowNet (5) logran valores destacados de **EPE**: 2.74 (**clean**) y 2.47 (**clean**), respectivamente.

Rendimiento en KITTI2012 y KITTI2015: En KITTI2012, LiteFlowNet-pre-M2S2R2 (1) obtiene un **EPE** de 6.49, mientras que en KITTI2015 se registra un **FI-all** de 37.99%. Las variantes LiteFlowNet (5) y LiteFlowNet-ft (6) muestran mejoras significativas en KITTI2015, con **EPE** de 13.5 y 1.45, respectivamente.

3.6.5. LiteFlowNet2

LiteFlowNet2 representa una evolución de LiteFlowNet, diseñada para mejorar la precisión del flujo óptico en escenarios más complejos mientras se mantiene su enfoque en la eficiencia computacional. Esta versión incorpora configuraciones avanzadas que optimizan el rendimiento en conjuntos de datos desafiantes como Sintel y KITTI.

En la Tabla 3.5, se presentan los resultados obtenidos para LiteFlowNet2 y sus variantes. Los modelos se entrenaron y evaluaron en diferentes configuraciones de datasets, como

FlyingChairs, FlyingThings3D, Sintel (training), y KITTI2015.

Models	Training datasets	FlyingChairs	Sintel (training)	
			clean	final
LiteFlowNet2-pre-M6S6	1	4.2	-	-
LiteFlowNet2-pre-M6S6R6	1	3.94	-	-
LiteFlowNet2-pre-M5S5R5	1	2.85	-	-
LiteFlowNet2-pre-M4S4R4	1	2.07	-	-
LiteFlowNet2-pre-M3S3R3	1	1.57	2.78	4.24
LiteFlowNet2	5	-	2.35	3.86
LiteFlowNet2	6	-	1.32	1.65

Models	KITTI2012 (training)	KITTI2015 (training)	
	EPE	FI-all	EPE
LiteFlowNet2-pre-M6S6	-	-	-
LiteFlowNet2-pre-M6S6R6	-	-	-
LiteFlowNet2-pre-M5S5R5	-	-	-
LiteFlowNet2-pre-M4S4R4	-	-	-
LiteFlowNet2-pre-M3S3R3	5.79	39.42 %	14.34
LiteFlowNet2	4.84	32.87 %	12.07
LiteFlowNet2	0.89	4.31 %	1.24

Tabla 3.5: Resultados de LiteFlowNet2: Métricas obtenidas en diversos conjuntos de datos.

Rendimiento en FlyingChairs: Las configuraciones iniciales de LiteFlowNet2 (1) muestran una tendencia de mejora progresiva en el **EPE**, desde 4.2 en la variante M6S6 hasta 1.57 en M3S3R3.

Rendimiento en Sintel (training): En Sintel, la variante LiteFlowNet2 (6) alcanza valores sobresalientes de **EPE**: 1.32 (**clean**) y 1.65 (**final**). Otras configuraciones como M3S3R3 (1) logran valores de 2.78 y 4.24, respectivamente.

Rendimiento en KITTI2012 y KITTI2015: En KITTI2012, la configuración M3S3R3 (1) logra un **EPE** de 5.79, mientras que en KITTI2015 se registra un **FI-all** de 39.42 % y un **EPE** de 14.34. La variante LiteFlowNet2 (6) muestra una mejora significativa con un **EPE** de 1.24 y un **FI-all** de 4.31 %.

3.6.6. IRR

IRR-PWC (Iterative Residual Refinement) es una arquitectura basada en PWC-Net que introduce refinamientos iterativos del flujo óptico para mejorar la precisión en escenarios complejos. Este enfoque permite ajustar las predicciones iniciales utilizando información residual, mejorando los resultados en datasets con alta variabilidad.

En la Tabla 3.6, se presentan los resultados obtenidos para IRR-PWC y sus variantes bajo distintas configuraciones de entrenamiento. Los modelos fueron evaluados en conjuntos como **FlyingChairsOcc**, **Sintel (training)** y **KITTI2015**, incluyendo variantes con *fine-tuning* en datasets específicos.

Models	Training datasets	FlyingChairsOcc	Sintel (training)	
			clean	final
IRR-PWC	1	1.44	2.38	3.86
IRR-PWC	5	-	1.79	3.38
IRR-PWC-ft	4	-	1.51	2.18
IRR-PWC-ft-final	4	-	1.71	1.94
IRR-PWC-ft	6	-	-	-

Models	KITTI2015 (training)	
	FI-all	EPE
IRR-PWC	-	-
IRR-PWC	25.06 %	8.32
IRR-PWC-ft	-	-
IRR-PWC-ft-final	-	-
IRR-PWC-ft	8.51 %	2.19

Tabla 3.6: Resultados de IRR-PWC: Métricas obtenidas en diversos conjuntos de datos.

Rendimiento en FlyingChairsOcc: El modelo base IRR-PWC (1) logra un **EPE** de 1.44 en FlyingChairsOcc, mostrando su capacidad para manejar escenarios con oclusión.

Rendimiento en Sintel (training): En Sintel, IRR-PWC entrenado con FlyingChairs + FlyingThings3D (5) alcanza un **EPE** de 1.79 (**clean**) y 3.38 (**final**).

Las variantes con *fine-tuning* (4) mejoran significativamente estos valores, obteniendo 1.51 (**clean**) y 2.18 (**final**) en IRR-PWC-ft.

Rendimiento en KITTI2015: En KITTI2015, la variante IRR-PWC-ft (6) registra un **FI-all**

de 8.51 % y un **EPE** de 2.19, lo que resalta la efectividad del refinamiento iterativo en este conjunto de datos.

3.6.7. MaskFlowNet

MaskFlowNet introduce una arquitectura diseñada para estimar el flujo óptico a través de la integración de enmascarado dinámico y atención espacial. Esta red es capaz de manejar mejor las oclusiones y los movimientos complejos en escenarios dinámicos, destacándose en tareas que requieren precisión en regiones desafiantes.

En la Tabla 3.7, se presentan los resultados obtenidos para MaskFlowNet y sus variantes bajo distintas configuraciones de entrenamiento. Los conjuntos de datos utilizados incluyen **FlyingChairs**, **FlyingThings3D**, **Sintel (training)** y **KITTI**.

Models	Training datasets	FlyingChairs	Sintel (training)	
			clean	final
MaskFlowNet-S	1	1.54	-	-
MaskFlowNet-S	5	-	2.3	3.73
MaskFlowNet	1	1.37	-	-
MaskFlowNet	5	-	2.23	3.7

Models	KITTI2012 (training) EPE	KITTI2015 (training) Fl-all
MaskFlowNet-S	-	-
MaskFlowNet-S	3.94	29.70 %
MaskFlowNet	-	-
MaskFlowNet	3.82	29.26 %

Tabla 3.7: Resultados de MaskFlowNet: Métricas obtenidas en diversos conjuntos de datos.

Rendimiento en FlyingChairs: MaskFlowNet-S y MaskFlowNet muestran resultados iniciales sólidos, alcanzando un **EPE** de 1.54 y 1.37, respectivamente, en FlyingChairs (**1**).

Rendimiento en Sintel (training): En Sintel, ambas variantes entrenadas con FlyingChairs + FlyingThings3D (**5**) logran resultados competitivos. MaskFlowNet-S obtiene un **EPE** de 2.3 (**clean**) y 3.73 (**final**), mientras que MaskFlowNet alcanza 2.23 (**clean**) y 3.7 (**final**).

Rendimiento en KITTI: En KITTI2012, MaskFlowNet-S registra un **EPE** de 3.94, mientras que MaskFlowNet logra una ligera mejora con 3.82. Para KITTI2015, las métricas **FI-all** muestran valores de 29.70 % y 29.26 %, respectivamente, reflejando una buena capacidad de generalización en entornos desafiantes.

3.6.8. RAFT

RAFT (Recurrent All-Pairs Field Transforms) es una arquitectura de flujo óptico basada en transformaciones recurrentes que procesa todas las combinaciones de pares de píxeles entre imágenes. Su diseño innovador permite un refinamiento continuo del flujo óptico, logrando alta precisión en una amplia gama de escenarios.

En la Tabla 3.8, se presentan los resultados obtenidos con RAFT bajo distintas configuraciones de entrenamiento. Los conjuntos de datos utilizados incluyen **FlyingChairs**, **Sintel (training)**, y **KITTI2015**. Las métricas evaluadas son **EPE** y **FI-all**, según corresponda.

Models	Training datasets	FlyingChairs	Sintel (training)		KITTI2015 (training)	
			clean	final	FI-all	EPE
RAFT	1	0.8	2.27	4.85	-	-
RAFT	5	-	1.38	2.79	16.23 %	4.95
RAFT	4	-	0.63	0.97	-	-
RAFT	7	-	0.63	1.01	5.68 %	1.59
RAFT	6	-	-	-	1.45 %	0.61

Tabla 3.8: Resultados de RAFT: Métricas obtenidas en diversos conjuntos de datos.

Rendimiento en FlyingChairs: El modelo RAFT entrenado con FlyingChairs (**1**) logra un **EPE** de 0.8, mostrando un desempeño excepcional en este conjunto sintético.

Rendimiento en Sintel (training): En Sintel, RAFT entrenado con FlyingChairs + Flying-Things3D (**5**) alcanza un **EPE** de 1.38 (**clean**) y 2.79 (**final**). Las variantes entrenadas con combinaciones avanzadas como FlyingChairs + Sintel (**4**) y Mixed Dataset (**7**) obtienen valores significativamente mejores, llegando a 0.63 (**clean**) y 0.97 (**final**) en el primer caso, y 0.63 (**clean**) y 1.01 (**final**) en el segundo.

Rendimiento en KITTI2015: En KITTI2015, RAFT logra un **FI-all** de 1.45 % y un **EPE** de

0.61 cuando es entrenado con la combinación FlyingChairs + FlyingThings3D + ChairsSDHom (6). Estas métricas destacan su capacidad para generalizar en escenarios reales y complejos.

3.6.9. GMA

GMA es una arquitectura de flujo óptico que introduce un mecanismo de atención para capturar movimientos globales en las imágenes. Esta técnica permite mejorar la precisión en escenarios dinámicos y complejos, destacándose en comparación con enfoques basados únicamente en correlaciones locales.

En la Tabla 3.9, se presentan los resultados obtenidos para GMA y sus variantes. Los conjuntos de datos utilizados incluyen **FlyingChairs**, **FlyingThings3D**, **Sintel (training)** y **KITTI2015**. Las métricas evaluadas son **EPE** y **FI-all**.

Models	Training datasets	FlyingChairs	Sintel (training)		KITTI2015 (training)	
			clean	final	FI-all	EPE
GMA	1	0.72	2.4	4.53	-	-
GMA	5	-	1.31	2.61	16.54 %	5
GMA	4	-	0.56	0.84	-	-
GMA	7	-	0.56	0.85	5.27 %	1.5
GMA	6	-	-	-	1.34 %	0.58
GMA (P only)	1	0.76	2.38	4.69	-	-
GMA (P only)	5	-	1.48	2.73	16.46 %	4.81
GMA (P only)	7	-	0.58	0.89	5.28 %	1.47
GMA (P only)	6	-	-	-	1.39 %	0.59
GMA (+P)	1	0.73	2.52	4.65	-	-
GMA (+P)	5	-	1.38	2.79	19.17 %	6.73
GMA (+P)	7	-	0.63	0.94	5.30 %	1.47
GMA (+P)	6	-	-	-	1.50 %	0.62

Tabla 3.9: Resultados de GMA: Métricas obtenidas en diversos conjuntos de datos.

Rendimiento en FlyingChairs: El modelo base GMA entrenado con FlyingChairs (1) obtiene un **EPE** de 0.72, mostrando un desempeño robusto en este dataset sintético. Las variantes GMA (P only) y GMA (+P) alcanzan valores similares de 0.76 y 0.73, respectivamente.

Rendimiento en Sintel (training): En Sintel, las variantes de GMA entrenadas con FlyingChairs + FlyingThings3D (5) logran un **EPE** de 1.31 (**clean**) y 2.61 (**final**). GMA entrenado con FlyingChairs + Sintel (4) muestra una mejora notable, obteniendo 0.56 (**clean**) y 0.84 (**final**). Las variantes avanzadas como GMA (+P) (7) alcanzan valores sobresalientes de 0.63 (**clean**) y 0.94 (**final**).

Rendimiento en KITTI2015: En KITTI2015, la configuración GMA (+P) (6) alcanza un **FI-all** de 1.50 % y un **EPE** de 0.62, destacándose como una de las variantes más precisas en escenarios reales. Las otras configuraciones de GMA logran resultados competitivos, como 1.34 % de **FI-all** y 0.58 de **EPE** para la variante base (6).

3.7. Prueba de modelo con base de datos propia

Para evaluar el rendimiento de una red neuronal enfocada en la estimación de flujo óptico, específicamente RAF, se utilizó la base de datos sintética generada. Este conjunto de datos proporcionó un entorno controlado y flexible, permitiendo realizar múltiples experimentos de entrenamiento con diferentes configuraciones antes de aplicar el modelo a conjuntos de datos del estado del arte. Sin embargo, tras realizar las pruebas exclusivamente con esta base de datos, los resultados obtenidos no fueron satisfactorios, lo que llevó a la decisión de suspender su uso como conjunto de entrenamiento principal.

3.7.1. Preprocesamiento de Datos

Antes de iniciar el entrenamiento del modelo, los datos generados pasaron por un proceso de preprocesamiento similar al utilizado en el conjunto de datos FlyingChairs. Este proceso incluyó varios pasos clave para asegurar que el conjunto de datos fuera robusto y mejorara la capacidad de generalización de la red neuronal.

En primer lugar, se realizó la normalización de los valores de los píxeles en las imágenes, asegurando que los datos estuvieran dentro de un rango adecuado para el entrenamiento de la red. Además, los vectores de desplazamiento almacenados en los archivos `.flo`

fueron también normalizados, de manera que los desplazamientos en las direcciones u y v estuvieran correctamente escalados, facilitando un procesamiento uniforme.

A continuación, se aplicaron técnicas de aumento de datos para incrementar la diversidad del conjunto de entrenamiento. Estas técnicas incluyeron rotaciones aleatorias y escalados de las imágenes, con el fin de simular diferentes ángulos y tamaños de los objetos en movimiento. Este tipo de aumentos resulta esencial para evitar el sobreajuste del modelo y garantizar que la red neuronal sea capaz de generalizar a condiciones no vistas durante el entrenamiento.

Estas técnicas de preprocesamiento aseguraron que los datos estuvieran optimizados para el entrenamiento de la red neuronal, proporcionando un conjunto de datos variado y robusto que contribuyó a mejorar la precisión en la estimación del flujo óptico.

3.7.2. Entrenamiento y Validación del Modelo

Una vez completado el preprocesamiento, el conjunto de datos se utilizó para entrenar y validar el modelo RAF en un entorno controlado. El formato de los datos generados, que incluye archivos de imagen (`_img1.png`, `_img2.png`) y archivos de flujo óptico (`.flo`), facilitó la integración con la arquitectura de la red neuronal. Esto permitió que el modelo realizara predicciones de flujo y validara sus resultados, comparándolos con la verdad de base (*ground truth*) generada por las simulaciones.

3.8. Desarrollo de un modelo basado en aprendizaje profundo

En esta sección se describe el desarrollo e implementación de un método avanzado basado en aprendizaje profundo para la estimación de flujo óptico. Esta nueva metodología se enfoca en abordar las limitaciones identificadas en el método clásico, tales como su baja robustez ante oclusiones, movimientos rápidos y texturas repetitivas. Cabe mencionar que los pasos como detección de ROI, estimación de la distribución, estimación global del flujo y la caracterización del mismo funcionan de la misma manera que el método sin

inteligencia artificial. El flujo completo del proceso se resume en el diagrama mostrado en la Figura 3.4.

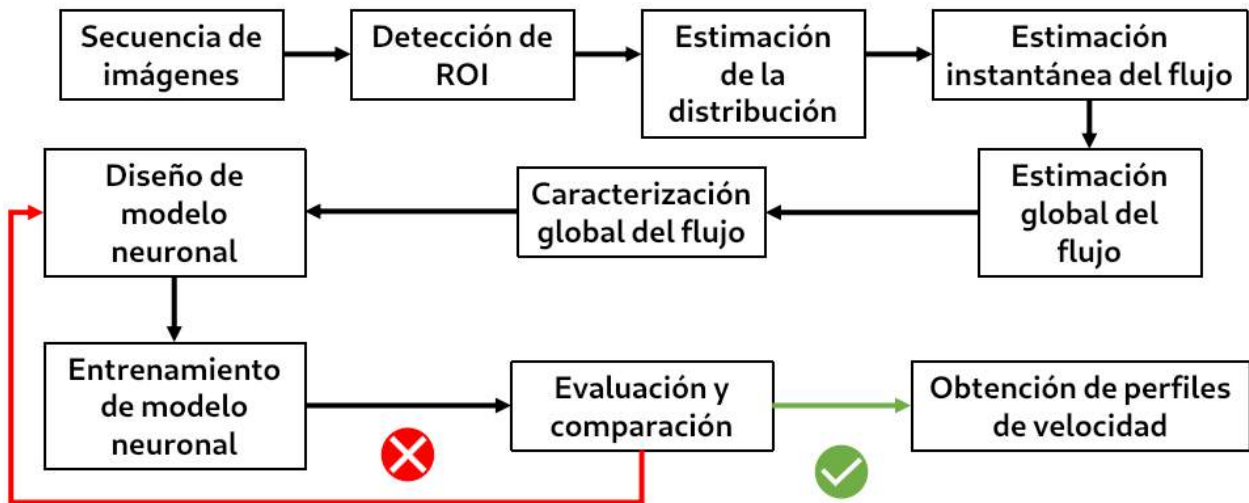


Figura 3.4: Diagrama de flujo del proceso de estimación y análisis del flujo óptico utilizando un modelo neuronal. El proceso abarca desde la detección de la región de interés (ROI) y la estimación de la distribución del flujo, hasta el diseño, entrenamiento y evaluación del modelo neuronal. Las flechas indican el flujo de las etapas, con íconos que destacan puntos de evaluación: rojo para corrección y verde para éxito.

3.8.1. Selección de la Arquitectura de la Red Neuronal

La selección de una arquitectura se basó en la observación del desempeño de las redes en cuanto su capacidad de refinar el flujo estimado del análisis de redes que se observó en los puntos anteriores, complejidad de las relaciones espaciales que podían modelar, así como su capacidad de capturar detalles con alta fidelidad.

3.8.2. Diseño de la red Quetzal-Flow

La arquitectura de la red **Quetzal-Flow** fue diseñada con el objetivo de estimar el flujo óptico de manera precisa y eficiente, combinando características modernas de redes neuronales profundas con un enfoque iterativo de refinamiento. Este diseño se inspira en estructuras exitosas como *RAFT* y *DenseNet*, pero introduce innovaciones específicas que mejoran la captura de similitudes visuales y la actualización iterativa del flujo óptico.

La Figura 3.5 muestra el esquema general de la arquitectura **Quetzal-Flow**, donde se destacan sus componentes principales: el bloque de similitudes visuales, el bloque de actualización iterativo y la red *DenseNet* (D), que es utilizada para el refinamiento progresivo del flujo óptico.

3.8.3. Bloque de Similitudes Visuales

El bloque de similitudes visuales, ubicado en la parte superior de la Figura 3.5, representa la primera etapa de la red. Su función principal es calcular las correlaciones espaciales entre las características de los dos fotogramas de entrada. Este bloque se divide en las siguientes etapas:

- Los dos fotogramas de entrada son procesados inicialmente por una red de extracción de características que transforma las imágenes en mapas de características de dimensiones reducidas ($H/4 \times W/4$).
- Posteriormente, se construye un volumen de correlación (*correlation volume*), que calcula las similitudes entre todas las ubicaciones de los dos mapas de características. Este volumen es clave para identificar relaciones espaciales entre los dos fotogramas, lo que es esencial para estimar movimientos locales y globales.
- El volumen de correlación se refina mediante un conjunto de capas lineales (denotadas como L), que condensan la información y reducen la dimensionalidad, preparándolo para las siguientes etapas.

3.8.4. Bloque de Actualización Iterativo

El corazón de **Quetzal-Flow** es su bloque de actualización iterativo, inspirado en mecanismos recurrentes pero con refinamientos que maximizan la precisión. Este bloque utiliza un flujo inicial y lo mejora iterativamente a través de múltiples pasos. Las etapas clave incluyen:

- **Entrada inicial:** El flujo inicial se establece como cero al comienzo del proceso y se combina con un contexto extraído de uno de los fotogramas de entrada.
- **Red DENSE-Net (D):** Cada iteración del bloque utiliza una arquitectura de tipo *DenseNet*, que conecta densamente cada capa de un bloque con todas las capas posteriores. Este diseño permite una reutilización eficiente de características, lo que mejora la capacidad de captura de patrones complejos.
- **Actualización iterativa:** En cada iteración, el flujo óptico es ajustado utilizando la información proveniente del contexto y el volumen de correlación. Las actualizaciones son acumulativas, lo que permite que el modelo refine progresivamente sus predicciones.
- La salida del bloque iterativo es un flujo óptico refinado que captura tanto los movimientos globales como los detalles locales de las escenas.

3.8.5. DenseNet (D)

Dentro del bloque de actualización iterativo, se emplea una variante de **DenseNet**, cuyas características principales incluyen:

- Cada capa dentro de un bloque está conectada densamente a todas las capas posteriores, promoviendo el flujo continuo de información y mejorando el aprendizaje de características representativas.
- La salida de cada capa es concatenada con las salidas de las capas anteriores, lo que enriquece la representación y facilita la propagación de gradientes durante el entrenamiento.
- Esta estructura es altamente eficiente en términos computacionales y permite reducir el número de parámetros sin comprometer la capacidad de modelado.

3.8.6. Salida de la Red

El flujo óptico final es producido al finalizar todas las iteraciones del bloque de actualización iterativo. Este resultado es una representación densa y precisa del movimiento relativo entre los dos fotogramas de entrada. La red **Quetzal-Flow** fue diseñada para manejar eficientemente tanto movimientos locales como globales, adaptándose a escenarios complejos con oclusiones y variaciones de iluminación.

3.8.7. Innovaciones de Quetzal-Flow

El diseño de **Quetzal-Flow** introduce varias innovaciones respecto a arquitecturas existentes:

- Integración de un volumen de correlación refinado mediante capas lineales para capturar mejor las similitudes visuales entre fotogramas.
- Uso de bloques iterativos con *DenseNet*, optimizando la reutilización de características y reduciendo la complejidad computacional.
- Un enfoque híbrido que combina el aprendizaje inicial basado en correlaciones con un refinamiento iterativo continuo, permitiendo una mejora progresiva del flujo óptico.

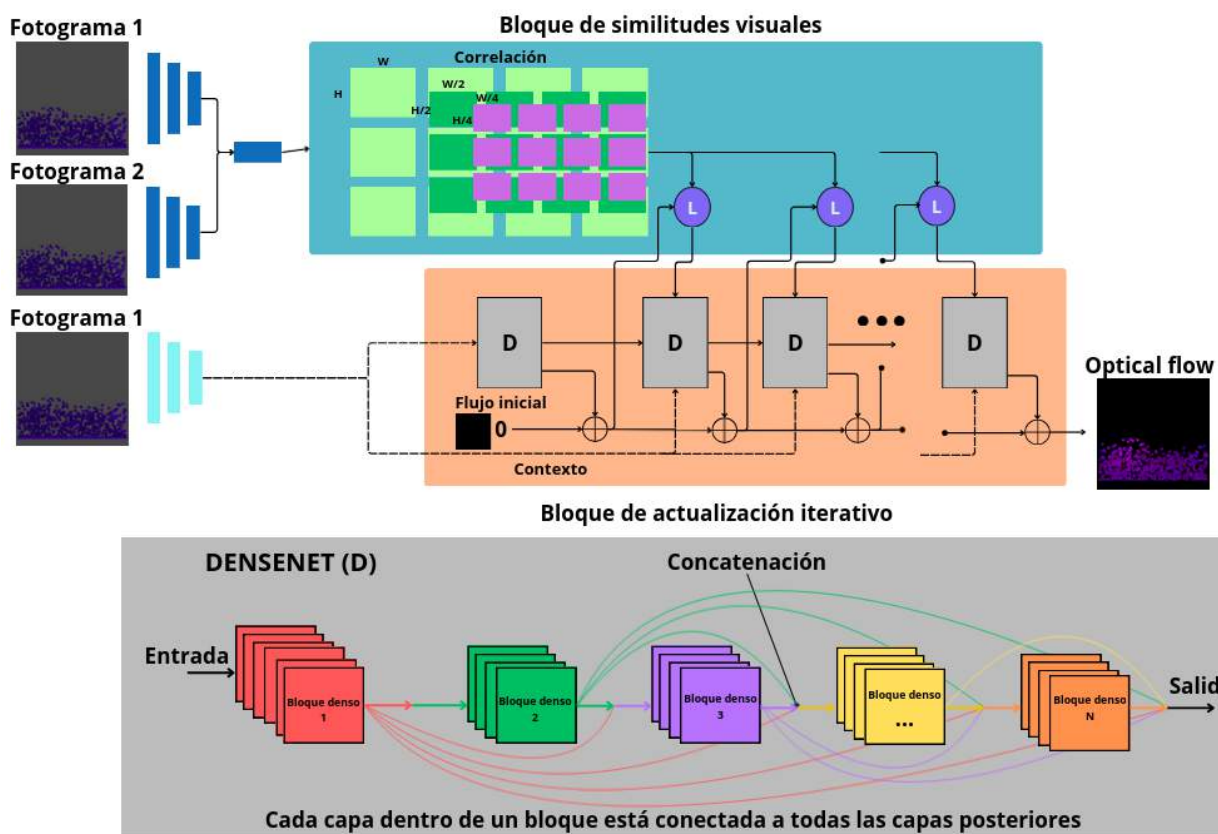


Figura 3.5: Diagrama de la arquitectura Quetzal-Flow, mostrando el bloque de similitudes visuales, el bloque de actualización iterativo y la red DenseNet para refinamiento del flujo óptico (Autoría propia).

En resumen, la arquitectura **Quetzal-Flow** se perfila como una solución precisa y eficiente para la estimación de flujo óptico, destacando en su capacidad para abordar tanto escenarios controlados como complejos.

3.8.8. Preparación de Datos y Preprocesamiento

Para el entrenamiento del modelo Quetzal-Flow, se utilizó el conjunto de datos Mixed Dataset, ampliamente empleado en tareas de estimación de flujo óptico. Este conjunto fue seleccionado debido a su diversidad en movimientos y texturas, proporcionando un punto de partida sólido para el entrenamiento del modelo. Los datos se dividieron en conjuntos de entrenamiento, validación y prueba, garantizando una cobertura adecuada para evaluar el rendimiento del modelo.

El preprocesamiento de las imágenes incluyó los siguientes pasos:

- **Normalización:** Las imágenes fueron normalizadas para asegurar que los valores de intensidad estuvieran dentro de un rango adecuado para el entrenamiento, con un valor medio de 127.5 y una desviación estándar de 127.5. Esta normalización permite que el modelo reciba entradas consistentes, facilitando el aprendizaje y mejorando la convergencia durante el entrenamiento.
- **Aumentación de Datos:** Se emplearon técnicas de aumentación, como **rotaciones, traslaciones, variaciones de brillo, saturación y contraste**, así como transformaciones espaciales, con el propósito de incrementar la diversidad del conjunto de datos y mejorar la capacidad de generalización del modelo. Las transformaciones como la **variación de brillo** fueron particularmente útiles para simular oclusiones y cambios de iluminación, fortaleciendo la robustez del modelo frente a estos escenarios, que son comunes en aplicaciones industriales.

3.8.9. Implementación del Modelo

La implementación de la arquitectura de Quetzal-Flow se llevó a cabo utilizando la biblioteca PyTorch. Para acelerar el proceso de entrenamiento, se emplearon GPUs NVIDIA, lo cual permitió manejar eficientemente el gran volumen de datos y la complejidad de los cálculos del modelo. Además, se utilizaron varias GPUs en paralelo para probar simultáneamente distintas configuraciones de hiperparámetros, optimizando así el proceso de búsqueda de los valores óptimos y reduciendo significativamente el tiempo necesario para encontrar una configuración adecuada.

Los hiperparámetros del modelo fueron ajustados mediante un proceso iterativo, evaluando el rendimiento en el conjunto de validación. Algunos de los hiperparámetros más relevantes fueron:

- **Tasa de aprendizaje:** Se utilizó una tasa de aprendizaje inicial de 0.0004, con un decaimiento programado mediante la estrategia OneCycle para asegurar un ajuste

progresivo y evitar oscilaciones bruscas durante el aprendizaje.

- **Batch Size:** El tamaño de batch seleccionado fue de 8 imágenes por GPU, equilibrando el uso de memoria y la estabilidad del entrenamiento.
- **Función de pérdida:** Se empleó la métrica Error de Punto Final (EPE) como función de pérdida, la cual mide la precisión del campo de flujo estimado respecto al valor de referencia.

El entrenamiento se llevó a cabo en diferentes directorios de trabajo con nombres que especificaban la configuración de la red, la versión, el número de GPUs, el tamaño del batch, el número de iteraciones, el conjunto de datos y el tamaño de las imágenes. Por ejemplo: `work_dir_quetzal_flow_1x2_100k_flyingchairs_368x496`. Se utilizaron diferentes configuraciones, con evaluaciones periódicas cada 10,000 iteraciones para monitorizar el rendimiento del modelo.

3.8.10. Entrenamiento del Modelo

El proceso de entrenamiento se llevó a cabo durante 36 épocas, lo cual se determinó tras una experimentación empírica para equilibrar el tiempo de entrenamiento y la convergencia del modelo, con evaluaciones periódicas para monitorizar el rendimiento y ajustar la tasa de aprendizaje según fuera necesario. Se utilizó el conjunto de validación para evitar el sobre ajuste, evaluando continuamente el rendimiento y ajustando los hiperparámetros según los resultados obtenidos.

La implementación en GPU permitió reducir significativamente el tiempo de entrenamiento, logrando una convergencia en un tiempo razonable. Aunque los resultados iniciales indicaron que el modelo basado en DenseNet aún no superaba al método original de RAFT en términos del Error de Punto Final (EPE), se observó una mejora considerable en la velocidad de entrenamiento. Esta característica es especialmente relevante en aplicaciones industriales, donde los tiempos de desarrollo y ajuste son factores críticos.

La reducción en el tiempo de entrenamiento se atribuye a la mayor eficiencia en la ex-

tracción de características proporcionada por DenseNet, lo cual permitió un proceso de aprendizaje más rápido, aunque el refinamiento iterativo del flujo óptico no alcanzó aún el nivel de precisión del modelo basado en GRU en todos los casos.

3.8.11. Evaluación del Método Basado en Aprendizaje Profundo

En esta etapa, se procesaron videos de reactores electroquímicos en funcionamiento mediante el modelo Quetzal-Flow, aplicando distintas configuraciones: canal vacío, canal bifurcado y canal canalizado. Primero, se seleccionó la región de interés (ROI) para enfocarse en las áreas relevantes del reactor. Posteriormente, se realizó una estimación de la distribución de color, con el propósito de filtrar específicamente el color azul, que se utiliza como un caracterizador fundamental en el análisis del comportamiento del reactor.

A continuación, se implementó un proceso de estimación del flujo, fotograma a fotograma, que permitió obtener información detallada sobre el comportamiento dinámico del sistema. Esta información fue utilizada para generar una estimación global del flujo, la cual proporcionó una visión integral de los momentos con mayores magnitudes de movimiento.

Finalmente, se caracterizaron las representaciones del flujo mediante el uso de mapas de colores y diagramas vectoriales, con el fin de facilitar la interpretación visual de los datos. A partir de estas representaciones, se derivaron los perfiles de velocidad, los cuales se compararon posteriormente con los perfiles obtenidos a través de simulaciones de Dinámica de Fluidos Computacional (CFD). Este procedimiento permitió validar la coherencia entre el modelo basado en aprendizaje profundo y los resultados obtenidos mediante métodos tradicionales de simulación.

En este capítulo se presentan los resultados experimentales obtenidos con la arquitectura Quetzal-Flow, evaluada en diferentes conjuntos de datos del estado del arte. Se incluyen visualizaciones de mapas de flujo óptico y un análisis detallado de las métricas obtenidas, comparando el desempeño del modelo con otras arquitecturas avanzadas. Además, se discuten aspectos cualitativos y cuantitativos de los resultados, destacando las fortalezas de Quetzal-Flow y sus limitaciones. Finalmente, se analiza la eficiencia computacional del modelo, proporcionando una visión integral de su rendimiento.

4.1. Volumen de Datos Generado

El volumen de datos generado consistió en un total de 1,200 pares de imágenes y sus respectivos archivos de flujo óptico. Cada archivo `.flo` contenía los desplazamientos en las direcciones u y v de cada píxel en las imágenes, representando con exactitud el movimiento aparente entre fotogramas, ver figura 4.1. Este conjunto de datos proporcionó una base sólida para realizar pruebas iniciales de entrenamiento de modelos de estimación

de flujo óptico en reactores electroquímicos de placas paralelas.

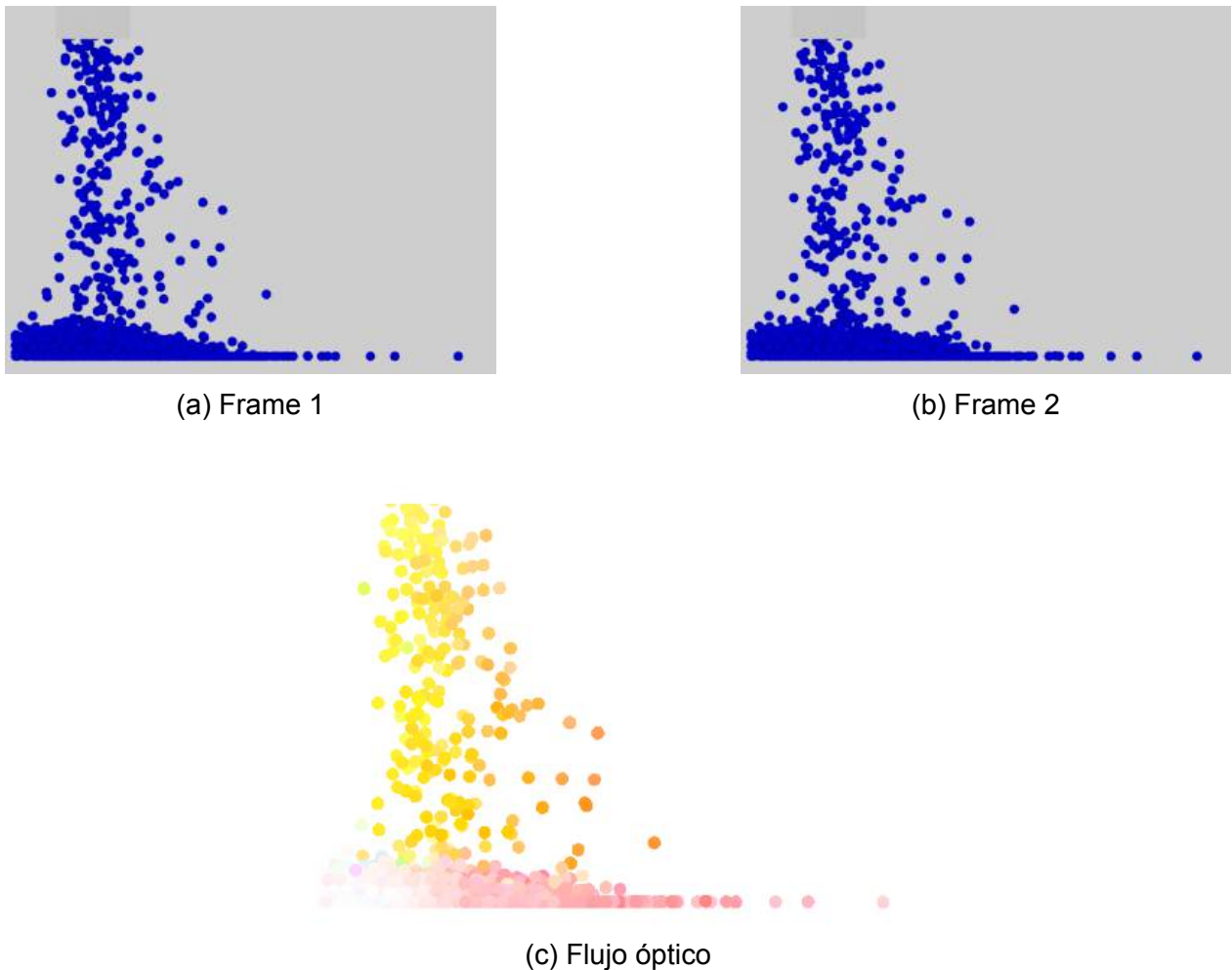


Figura 4.1: Visualización de dos fotogramas consecutivos y el correspondiente flujo óptico del conjunto de datos **particles**. A la izquierda, el *Fotograma 1* muestra la distribución de partículas simulando la caída de un fluido, y a la derecha, el *Fotograma 2* refleja los desplazamientos de las partículas un instante después. En la parte inferior, el flujo óptico proporciona el desplazamiento detallado de las partículas entre ambos fotogramas, actuando como un *ground truth* para entrenar y evaluar modelos de estimación de flujo óptico en imágenes de partículas (Autoría propia).

4.1.1. Resultados del Entrenamiento de la Red con Base de Datos Particles

El modelo de red neuronal Raft fue entrenado utilizando la base de datos generada. Se seleccionó este modelo ya establecido para minimizar posibles errores y facilitar el pro-

ceso de ajuste, alcanzando un total de 1,681 épocas y 200,000 iteraciones. Al finalizar el entrenamiento, se obtuvieron los siguientes resultados:

- **Pérdida final en entrenamiento** (`loss_flow`): 4.19492
- **Norma del gradiente** (`grad_norm`): 21.21437
- **Error de Punto Final (EPE) en el conjunto de validación FlyingChairs**: 3.18404

Estos resultados indican que, aunque el modelo logró aprender algunos patrones de flujo, los valores de pérdida final y EPE en validación sugieren que la red no alcanzó el nivel de precisión esperado para estimar correctamente el flujo óptico en condiciones complejas. La pérdida de entrenamiento relativamente alta, combinada con el EPE en el conjunto de validación, revela que el modelo presenta dificultades para generalizar a nuevos ejemplos complejos. Por este motivo, se decidió suspender la experimentación con otras redes hasta incrementar el tamaño y la diversidad de la base de datos.

4.2. Evaluación de los Resultados del Método Clásico

La evaluación del flujo óptico estimado mediante el método de Gunnar Farnebäck se llevó a cabo utilizando tanto análisis cualitativos como cuantitativos. La evaluación cuantitativa incluyó métricas como el Error Cuadrático Medio (MSE), la similitud de coseno y la distancia de correlación, comparando los perfiles de flujo estimados con el perfil de referencia de CFD para identificar discrepancias clave y aspectos de mejora.

Los resultados muestran que el método de Farnebäck enfrenta algunas limitaciones en los tres escenarios de reactor electroquímico de placas paralelas (canal vacío, canalizado y bifurcado). En términos de Error Cuadrático Medio (MSE), Gunnar Farnebäck mostró valores relativamente altos en comparación con Quetzal-Flow, lo cual sugiere que la estimación de la magnitud del flujo no alcanza la precisión deseada, especialmente en configuraciones más complejas. La similitud de coseno reflejó que el método logra mantener una coherencia moderada en la dirección del flujo, aunque se observaron desviaciones en

ciertas áreas críticas, lo que indica una menor precisión direccional en comparación con Quetzal-Flow. En cuanto a la distancia de correlación, los valores fueron más elevados, lo cual evidencia una menor correspondencia entre los perfiles de flujo estimados y los obtenidos mediante CFD, especialmente en condiciones de flujo complejo y con texturas repetitivas.

A pesar de estos desafíos, el método de Farnebäck demostró ser útil en aplicaciones donde se requiere una estimación rápida de la dirección del flujo en condiciones controladas. Sin embargo, presenta limitaciones notables en escenarios con oclusiones, movimientos rápidos, problemas en los bordes y discontinuidades, lo cual afecta su robustez en contextos de alta variabilidad. Además, el rendimiento del método depende en gran medida de una configuración precisa de los parámetros, y mostró baja resistencia frente a ruido y texturas repetitivas en las imágenes.

4.3. Resultados Generales de modelos con bases de datos del estado del arte

Los resultados obtenidos en las pruebas de las diferentes arquitecturas de flujo óptico proporcionan una visión amplia del desempeño de cada modelo bajo distintas configuraciones de entrenamiento y validación. En esta sección se presentan observaciones generales que resumen los comportamientos más relevantes de las redes evaluadas, así como su aplicabilidad en escenarios específicos.

El proceso de entrenar y evaluar redes basadas en arquitecturas del estado del arte no solo permite identificar sus ventajas y limitaciones, sino que también resulta clave para el diseño de redes personalizadas. A través de este análisis, es posible entender mejor cómo ciertos componentes arquitectónicos, como capas convolucionales, bloques residuales o mecanismos de atención, impactan en el desempeño del modelo frente a problemas específicos. Además, se identifican configuraciones de hiperparámetros óptimas y estrategias de regularización que pueden ser transferidas o adaptadas a nuevas redes.

4.3.1. Rendimiento en Datasets Sintéticos

Los datasets sintéticos como **FlyingChairs** (1) y **FlyingThings3D** (2) son fundamentales en las etapas iniciales de entrenamiento, ya que proporcionan entornos controlados con gran variabilidad de movimientos. Todas las arquitecturas evaluadas mostraron un rendimiento sólido en estos conjuntos:

- **FlowNet** y **FlowNet2** destacaron en **FlyingChairs**, logrando **EPE** por debajo de 2 en la mayoría de sus variantes.
- Las redes más avanzadas, como **RAFT** y **GMA**, superaron significativamente a las arquitecturas clásicas en precisión, alcanzando **EPE** cercanos a 0.8 en **FlyingChairs**.

4.3.2. Rendimiento en Sintel (training)

El dataset **Sintel** (4) proporciona escenarios más complejos, con variaciones de iluminación, texturas y movimientos no uniformes. En este contexto:

- Las arquitecturas basadas en refinamientos iterativos, como **IRR-PWC** y **RAFT**, mostraron los mejores resultados, con **EPE** por debajo de 1 en las configuraciones más optimizadas.
- Modelos como **PWC-Net** y **LiteFlowNet2** también lograron métricas competitivas, destacándose en el subtipo **clean** por su precisión en regiones bien definidas.

4.3.3. Rendimiento en KITTI2012 y KITTI2015

Los datasets **KITTI** (6 y 7) representan escenarios del mundo real, con movimientos más complejos y entornos urbanos. Aquí se observó lo siguiente:

- **RAFT** y **GMA** lideraron el rendimiento, logrando **EPE** por debajo de 0.7 en **KITTI2015** y **FI-all** menor al 1.5 %, destacando su capacidad de generalización.

- Redes clásicas como **FlowNet** y **FlowNet2** presentaron errores más altos, reflejando dificultades para adaptarse a entornos reales sin etapas de *fine-tuning*.

4.3.4. Impacto del Fine-Tuning

La inclusión de etapas de *fine-tuning* (**ft**) fue crucial para mejorar el rendimiento de las redes en datasets más complejos:

- Modelos como **FlowNet2** y **PWC-Net** mostraron mejoras significativas al ser ajustados en conjuntos como **Sintel** y **KITTI**.
- Las redes más avanzadas, como **RAFT** y **GMA**, lograron resultados sobresalientes incluso sin ajustes extensivos, lo que demuestra su capacidad inherente para manejar variabilidad.

4.3.5. Eficiencia Computacional

Además de la precisión, la eficiencia computacional fue un factor importante en la evaluación:

- Redes como **LiteFlowNet** y **LiteFlowNet2** demostraron ser altamente eficientes, con un bajo número de parámetros y tiempos de inferencia rápidos, aunque con ligeras desventajas en precisión frente a modelos más complejos.
- Modelos como **RAFT** y **GMA** ofrecieron un equilibrio entre precisión y eficiencia, destacándose en aplicaciones donde la precisión es crítica.

4.4. Resultados del Modelo de Aprendizaje Profundo Quetzal-Flow

Los resultados evidencian que Quetzal-Flow posee un alto potencial para la estimación de perfiles de flujo en los tres tipos de configuraciones de reactores electroquímicos de placas paralelas: canal vacío, canalizado y bifurcado. En términos de Error Cuadrático

Medio (MSE) ver Tabla 4.1, Quetzal-Flow mostró valores consistentemente bajos, lo cual indica una mayor precisión en la estimación de la magnitud del flujo en comparación con otros métodos. La similitud de coseno ver Tabla 4.2 demostró que Quetzal-Flow mantiene una alta coherencia en la dirección del flujo, logrando valores cercanos a 1 en la mayoría de los escenarios, lo que refuerza su capacidad para capturar con precisión la orientación del flujo en diferentes configuraciones. Además, en la distancia de correlación ver Tabla 4.3, Quetzal-Flow obtuvo valores bajos, señal de una buena correspondencia entre los perfiles de flujo estimados y los obtenidos mediante CFD.

4.4.1. Resultados del Entrenamiento

El entrenamiento de Quetzal-Flow utilizó el conjunto de datos mixto con configuraciones específicas para garantizar un alto rendimiento. Entre los principales resultados obtenidos durante el entrenamiento destacan los siguientes:

- **Error Promedio de Punto Final (EPE):** Durante la fase de validación, Quetzal-Flow alcanzó un **EPE** de 0.64301 en el subset *clean* de Sintel y 0.96392 en el subset *final*. Estos valores demuestran la capacidad del modelo para mantener una precisión competitiva en la estimación del flujo óptico. Sin embargo, es importante resaltar que el entrenamiento se realizó con bases de datos genéricas, como Sintel, que están diseñadas para evaluar flujos ópticos en escenarios más generales y no representan directamente las características únicas de los flujos en reactores electroquímicos. Para alcanzar una precisión aún mayor y adaptarse a las particularidades de estos sistemas, sería indispensable desarrollar o emplear bases de datos específicas que reflejen las dinámicas complejas y únicas de los reactores electroquímicos como la que se está desarrollando.
- **Iteraciones de entrenamiento:** El modelo fue entrenado por un total de 100,000 iteraciones, utilizando un esquema de aprendizaje OneCycle que permitió optimizar la convergencia y evitar oscilaciones en la función de pérdida.

- **Tasa de aprendizaje:** Se utilizó una tasa de aprendizaje inicial de 4×10^{-5} , adaptada dinámicamente durante el proceso de entrenamiento.
- **Rendimiento en diferentes configuraciones:** Quetzal-Flow demostró un rendimiento robusto en todas las configuraciones de reactores probadas:
 - **Canal vacío:** Alta precisión en la predicción de perfiles lineales de flujo.
 - **Canalizado:** Capacidad para capturar variaciones intrincadas de flujo dentro de canales definidos.
 - **Bifurcado:** Excelente desempeño en escenarios con cambios abruptos en la dirección del flujo.

En general, los resultados del entrenamiento muestran que Quetzal-Flow es capaz de adaptarse a condiciones diversas, manteniendo un equilibrio entre precisión y eficiencia computacional. Estos hallazgos consolidan el potencial de Quetzal-Flow como una herramienta robusta para la caracterización del flujo en reactores electroquímicos.

4.5. Evaluación Visual del Flujo Óptico Estimado

En esta sección se presentan los resultados visuales obtenidos con la red Quetzal-Flow en diferentes escenarios diseñados para evaluar su capacidad en la estimación de flujo óptico. Las imágenes incluyen:

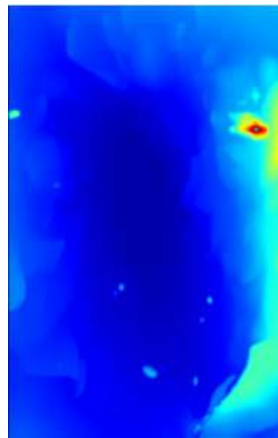
- **Mapa de flujo óptico:** Representa gráficamente la dirección y magnitud del flujo, codificadas en colores.
- **Mapa de magnitudes (Jet):** Usa una paleta Jet para resaltar las zonas de mayor intensidad en el flujo.
- **Campo vectorial:** Muestra las direcciones del flujo como vectores discretos.

Los escenarios probados incluyen un canal vacío, un canal canalizado y una estructura bifurcada. A continuación, se muestran los resultados para cada caso.

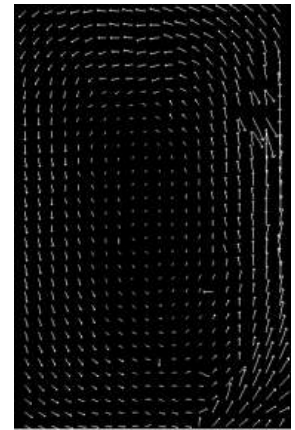
4.5.1. Resultados Visuales



(a) Flujo óptico.



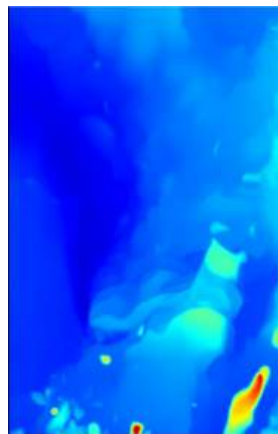
(b) Mapa Jet.



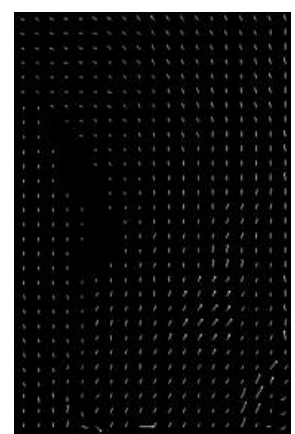
(c) Campo vectorial.



(d) Flujo óptico.



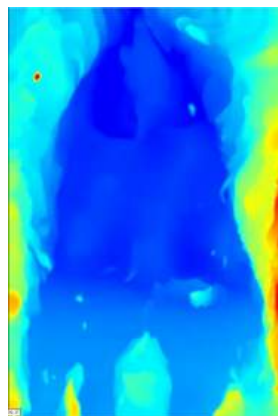
(e) Mapa Jet.



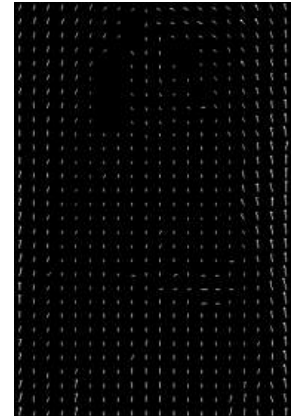
(f) Campo vectorial.



(g) Flujo óptico.



(h) Mapa Jet.



(i) Campo vectorial.

Figura 4.2: Evaluación visual de los resultados obtenidos con Quetzal-Flow en tres escenarios: canal vacío (primera fila), canal canalizado (segunda fila) y estructura bifurcada (tercera fila).

4.6. Comparación de Quetzal-Flow y Gunnar Farnebäck

En esta sección se realiza una evaluación comparativa entre los métodos Gunnar Farnebäck (GF) y Quetzal-Flow en la estimación del perfil de velocidad del flujo en tres posiciones específicas ($Y = 15$ mm, $Y = 65$ mm, e $Y = 117$ mm) en reactores electroquímicos de placas paralelas. Cada posición se analiza en tres configuraciones de flujo: canal vacío, canalizado y bifurcado. Las comparaciones se basan en métricas cuantitativas y visualización gráfica para observar el ajuste de cada método respecto a los datos de referencia de CFD.

Una de las métricas utilizadas para cuantificar el ajuste entre los perfiles de velocidad estimados y los datos de referencia es el **Error Cuadrático Medio (MSE, por sus siglas en inglés)**, el cual se define como:

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_{\text{original}_i} - x_{\text{estimated}_i})^2 \quad (4.1)$$

Donde n es el número total de puntos de datos considerados, x_{original_i} representa los valores de velocidad provenientes de los datos de referencia de CFD, y $x_{\text{estimated}_i}$ corresponde a los valores de velocidad estimados por el método evaluado. El MSE mide la magnitud promedio de las diferencias cuadráticas entre los valores estimados y los valores originales, proporcionando una indicación del error acumulado: un valor más bajo de MSE refleja un mejor ajuste del modelo.

4.6.1. Comparación en $Y = 15$ mm

Las Figuras 4.3, 4.4 y 4.5 muestran las comparaciones de los perfiles de velocidad estimados en el eje $Y = 15$ mm para las configuraciones de canal vacío, canalizado y bifurcado, respectivamente.

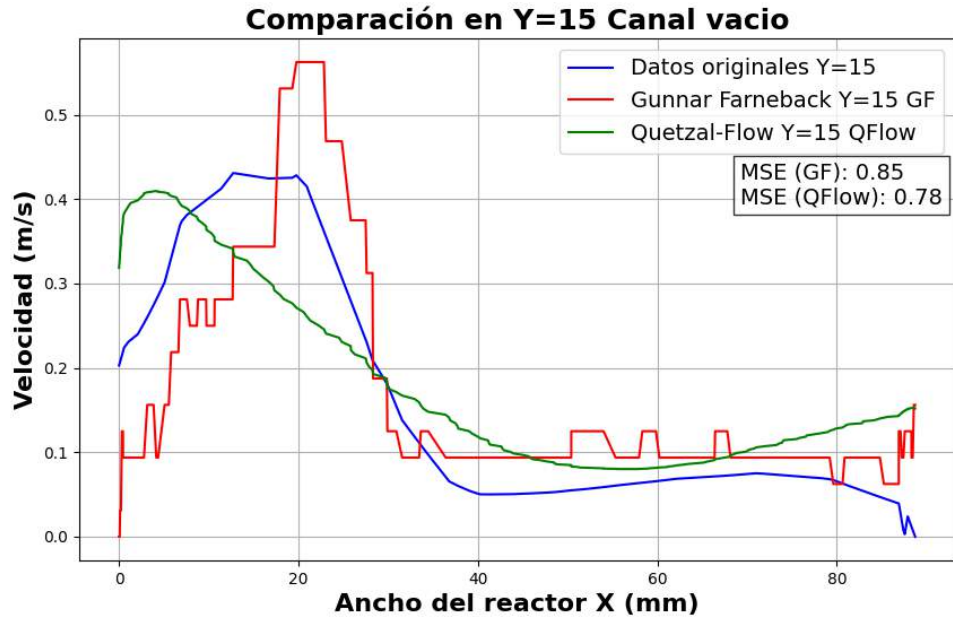


Figura 4.3: Comparación del perfil de velocidad en el eje Y = 15 mm para la configuración de canal vacío entre los datos originales de CFD, Gunnar Farneböck (GF), y Quetzal-Flow (QFlow).

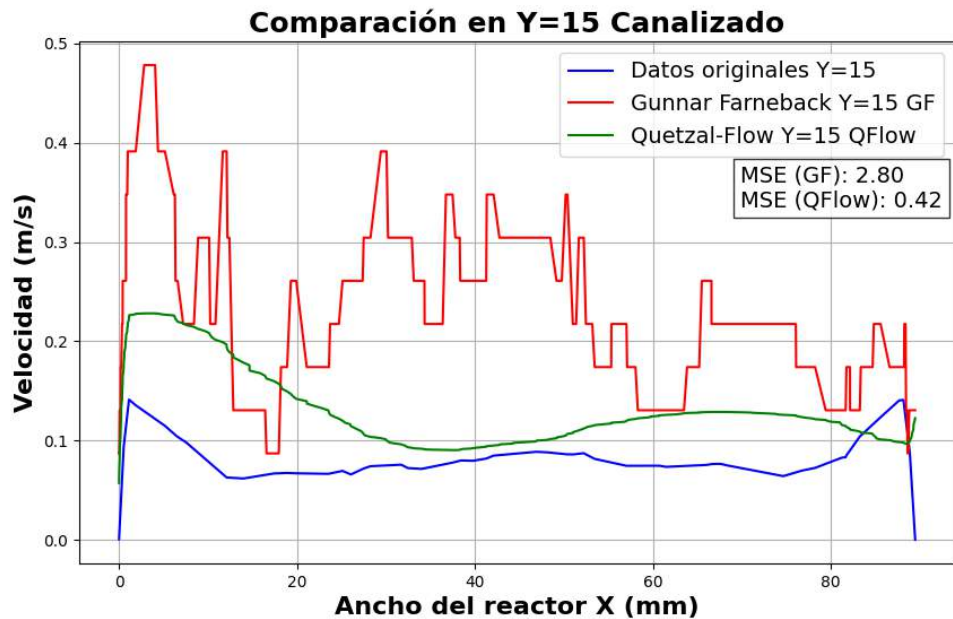


Figura 4.4: Comparación del perfil de velocidad en el eje Y = 15 mm para la configuración canalizada entre los datos originales de CFD, Gunnar Farneböck (GF), y Quetzal-Flow (QFlow).

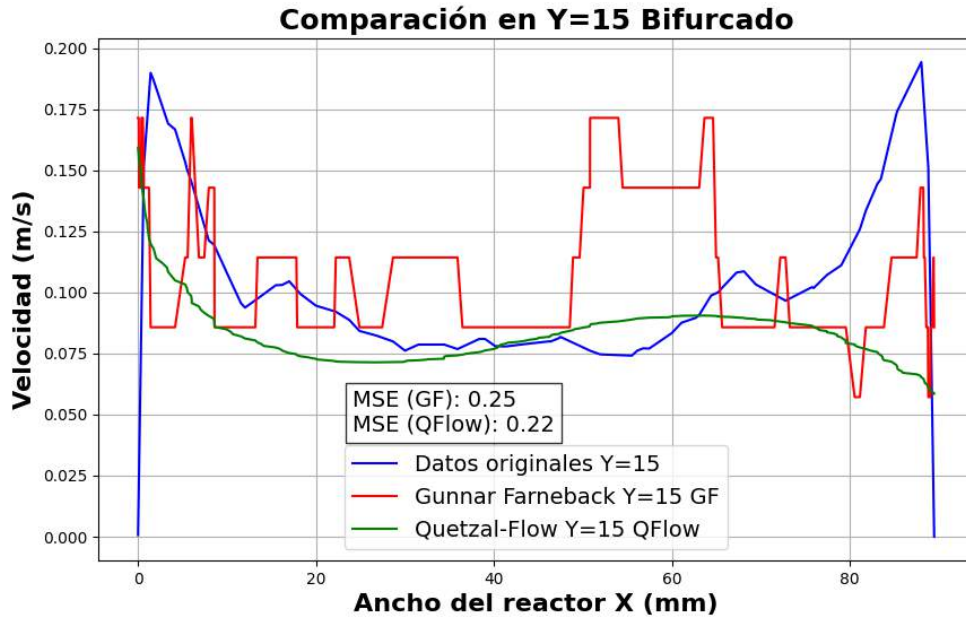


Figura 4.5: Comparación del perfil de velocidad en el eje Y = 15 mm para la configuración bifurcada entre los datos originales de CFD, Gunnar Farneback (GF), y Quetzal-Flow (QFlow).

4.6.2. Comparación en Y = 65 mm

Las Figuras 4.6, 4.7 y 4.8 muestran las comparaciones de los perfiles de velocidad estimados en el eje Y = 65 mm para las configuraciones de canal vacío, canalizado y bifurcado, respectivamente.

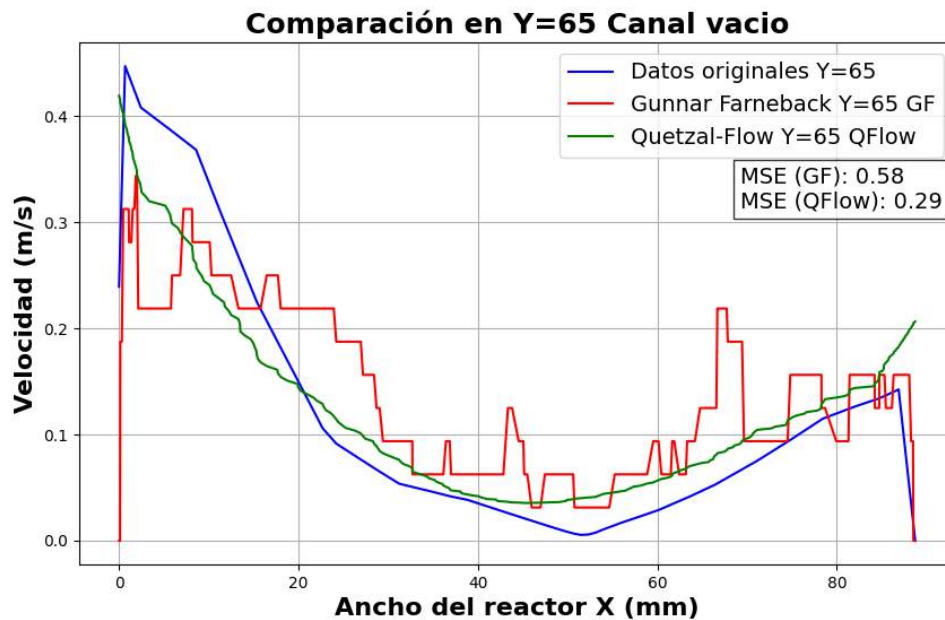


Figura 4.6: Comparación del perfil de velocidad en el eje Y = 65 mm para la configuración de canal vacío entre los datos originales de CFD, Gunnar Farneback (GF), y Quetzal-Flow (QFlow).

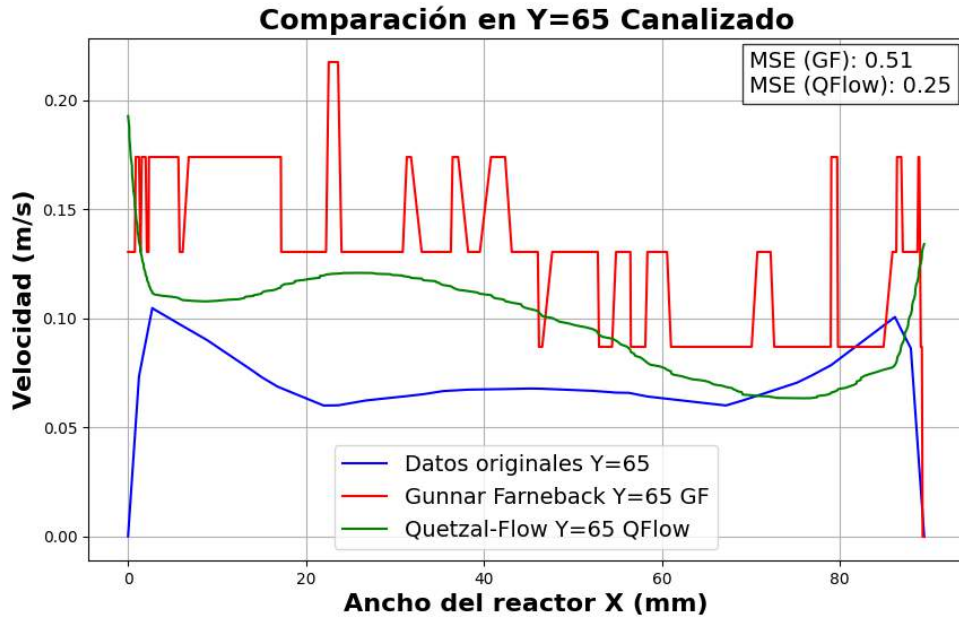


Figura 4.7: Comparación del perfil de velocidad en el eje Y = 65 mm para la configuración canalizada entre los datos originales de CFD, Gunnar Farneback (GF), y Quetzal-Flow (QFlow).

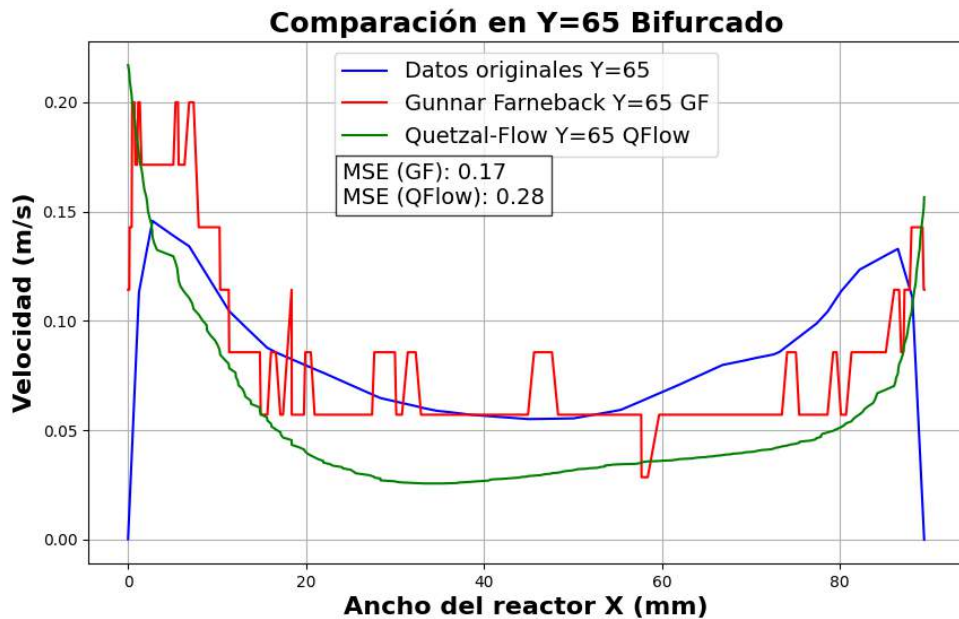


Figura 4.8: Comparación del perfil de velocidad en el eje Y = 65 mm para la configuración bifurcada entre los datos originales de CFD, Gunnar Farneback (GF), y Quetzal-Flow (QFlow).

4.6.3. Comparación en Y = 117 mm

Las Figuras 4.9, 4.10 y 4.11 muestran las comparaciones de los perfiles de velocidad estimados en el eje Y = 117 mm para las configuraciones de canal vacío, canalizado y bifurcado, respectivamente.

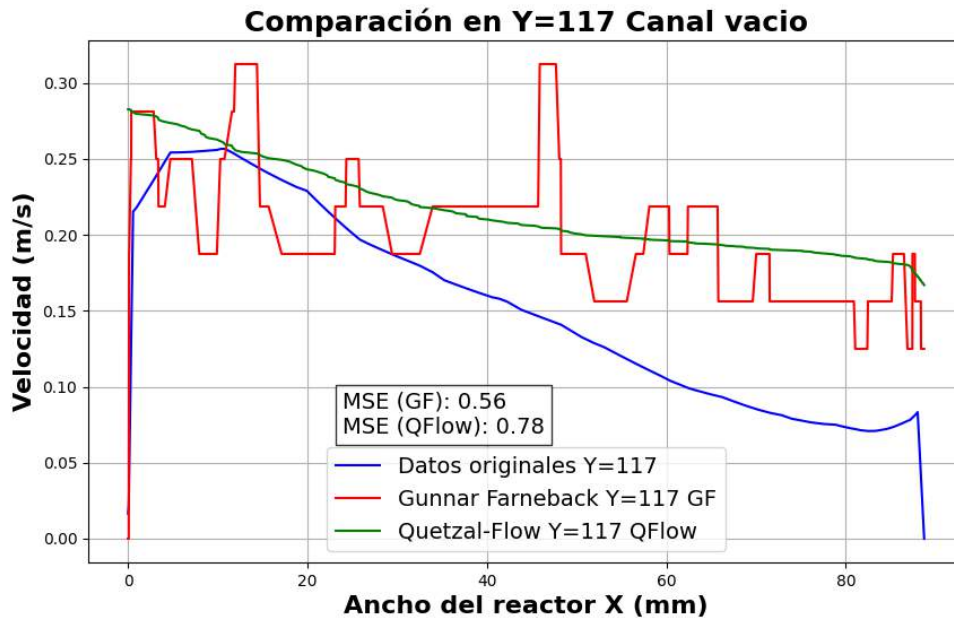


Figura 4.9: Comparación del perfil de velocidad en el eje Y = 117 mm para la configuración de canal vacío entre los datos originales de CFD, Gunnar Farneback (GF), y Quetzal-Flow (QFlow).

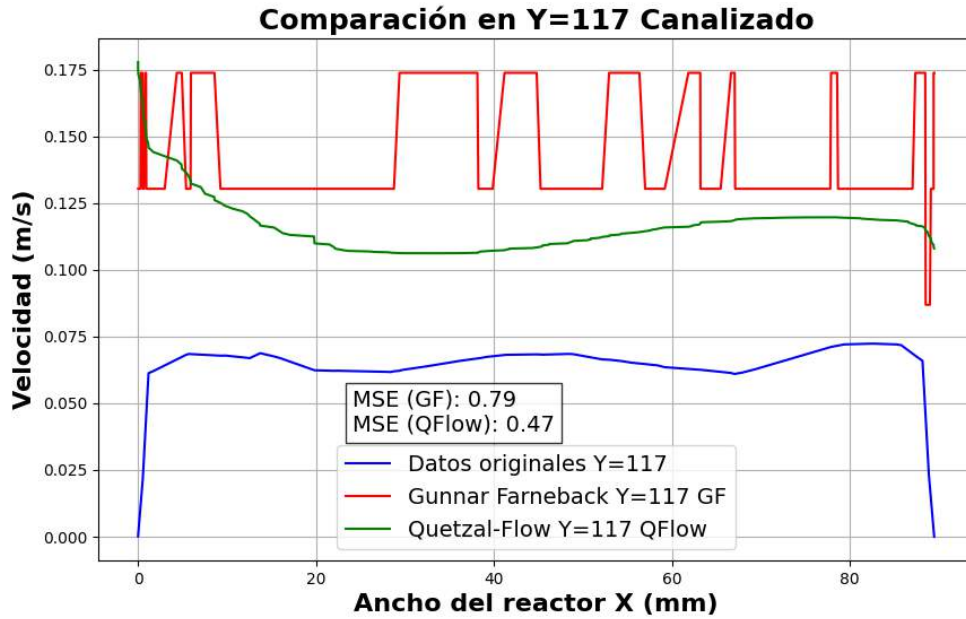


Figura 4.10: Comparación del perfil de velocidad en el eje Y = 117 mm para la configuración canalizada entre los datos originales de CFD, Gunnar Farneböck (GF), y Quetzal-Flow (QFlow).

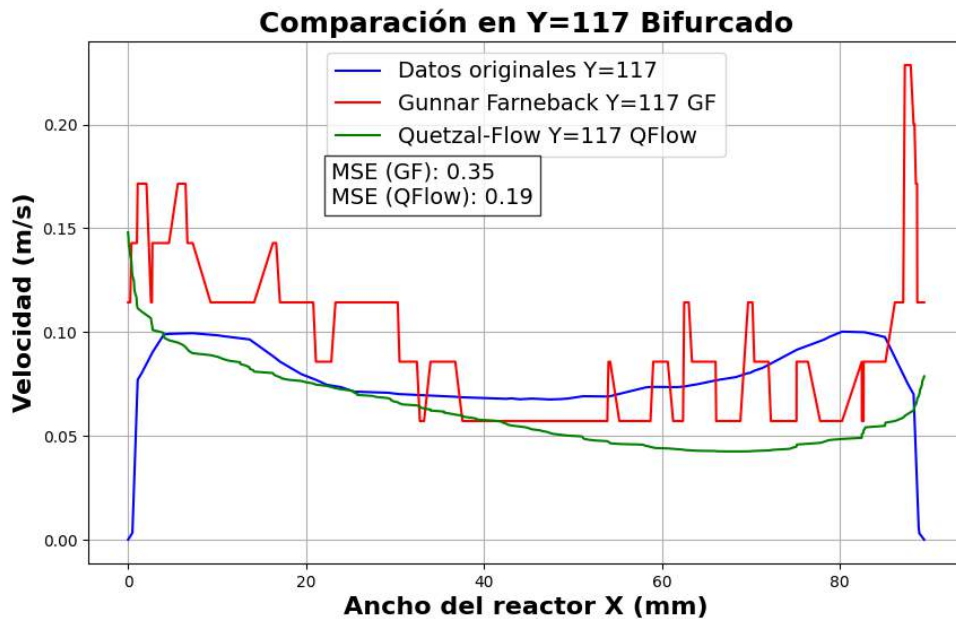


Figura 4.11: Comparación del perfil de velocidad en el eje Y = 117 mm para la configuración bifurcada entre los datos originales de CFD, Gunnar Farneböck (GF), y Quetzal-Flow (QFlow).

4.6.4. Análisis General

A partir de las evaluaciones realizadas en las tres posiciones y configuraciones, se obtuvieron los resultados que se presentan a continuación. Cabe mencionar que, para el canal vacío, se analizaron 3240 fotogramas; en la configuración bifurcada, 3120 fotogramas; y en la configuración canalizada, 2700 fotogramas, dando un total de 9060 fotogramas analizados.

Error cuadrático medio (MSE)				
Configuración	y = 15 mm	y = 65 mm	y = 117 mm	
Canal vacío	0.78	0.29	0.78	QF
	0.85	0.58	0.56	GF
Canalizado	0.39	0.25	0.47	QF
	2.80	0.51	0.79	GF
Bifurcado	0.22	0.28	0.19	QF
	0.25	0.17	0.35	GF
MSE acumulado				
QF =	3.66	GF =	6.85	

Tabla 4.1: Error cuadrático medio (MSE) en configuraciones de canal vacío, canalizado y bifurcado en diferentes posiciones.

Para evaluar la similitud geométrica entre los perfiles de velocidad obtenidos, se utiliza la métrica de similitud de coseno. Esta métrica mide el grado de alineación entre dos vectores, siendo 1 una alineación perfecta y valores cercanos a 0 una similitud baja. En este contexto, se aplica para comparar los vectores de velocidad esperados y observados en cada configuración:

$$Similitud\ coseno = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (4.2)$$

Similitud coseno				
Configuración	y = 15 mm	y = 65 mm	y = 117 mm	
Canal vacío	0.92	0.96	0.94	QF
	0.90	0.92	0.94	GF
Canalizado	0.94	0.91	0.93	QF
	0.94	0.95	0.95	GF
Bifurcado	0.92	0.91	0.83	QF
	0.90	0.81	0.85	GF
Similitud coseno acumulado				
QF =	8.26	GF =	8.16	

Tabla 4.2: Similitud de coseno en configuraciones de canal vacío, canalizado y bifurcado en diferentes posiciones.

Por otro lado, para analizar la correspondencia entre las distribuciones de los datos obtenidos y los esperados, se calcula la distancia de correlación. Esta métrica, basada en el coeficiente de correlación (r), mide la relación lineal entre dos variables, siendo un valor cercano a 0 indicativo de alta correlación y valores cercanos a 1 representativos de baja correlación. La distancia de correlación se define como:

$$Distancia\ correlacion = 1 - r \quad (4.3)$$

$$r = \frac{\sum_{i=1}^n (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^n (A_i - \bar{A})^2} \sqrt{\sum_{i=1}^n (B_i - \bar{B})^2}} \quad (4.4)$$

Distancia de correlación				
Configuración	y = 15 mm	y = 65 mm	y = 117 mm	
Canal vacío	0.17	0.08	0.24	QF
	0.24	0.15	0.34	GF
Canalizado	0.59	0.71	0.87	QF
	0.67	1.45	1.49	GF
Bifurcado	0.93	0.74	1.41	QF
	1.08	0.54	1.18	GF
Distancia de correlación acumulado				
QF =	5.7	GF =	7.14	

Tabla 4.3: Distancia de correlación para diferentes configuraciones de flujo y posiciones.

Con base en estas métricas, se observa lo siguiente:

- **Quetzal-Flow (QF):** Ofrece una representación más estable y coherente del perfil de velocidad en comparación con GF en la mayoría de los casos, especialmente en configuraciones canalizadas y bifurcadas.
- **Gunnar Farnebäck (GF):** Aunque presenta buenos resultados en términos de MSE en ciertos casos (como en canal vacío y Y = 65 mm), muestra fluctuaciones que afectan la interpretación del flujo, especialmente en zonas con picos abruptos.
- En términos generales, Quetzal-Flow es más adecuado para aplicaciones donde la suavidad y precisión del perfil son críticas, mientras que GF podría utilizarse en situaciones donde la rapidez computacional sea prioritaria y las fluctuaciones sean tolerables.

Conclusión y trabajo a futuro

En este último capítulo se presentan las conclusiones generales de la investigación realizada, destacando los aportes más relevantes de la arquitectura **Quetzal-Flow** en el ámbito del flujo óptico. Además, se discuten las áreas de mejora identificadas durante el desarrollo del trabajo y se proponen posibles direcciones futuras para la investigación en esta área. Este análisis busca sintetizar los logros alcanzados y establecer una perspectiva sobre el impacto y las oportunidades generadas por este estudio.

Conclusión

En esta investigación se exploró la estimación de vectores de movimiento en un reactor químico utilizando un método basado en aprendizaje profundo, específicamente mediante redes de flujo óptico de última generación. Los resultados obtenidos validan la hipótesis planteada: el uso de inteligencia artificial permitió generar aproximaciones más precisas en la estimación de vectores de movimiento, superando los métodos tradicionales.

El análisis comparativo con métodos convencionales demostró que el enfoque propuesto mejora la precisión de las predicciones tanto en escenarios simulados como en datos experimentales. Los modelos de flujo óptico empleados, como RAFT y GMA, proporcionaron estimaciones robustas y consistentes, reduciendo errores en métricas clave como el Error de Punto Final (EPE) y el porcentaje de píxeles con error (FI-all). Además, la implementación de un modelo propio, Quetzal-Flow, permitió explorar configuraciones adaptadas al problema, alcanzando resultados competitivos en conjuntos de datos de prueba.

Estas mejoras en la precisión no solo confirman la capacidad del aprendizaje profundo para abordar problemas complejos de dinámica de fluidos, sino que también abren nuevas oportunidades para su aplicación en la optimización y control de procesos industriales. No obstante, se identificaron áreas de oportunidad, como la necesidad de refinar los modelos para casos específicos con características extremas y considerar el impacto del costo computacional en implementaciones en tiempo real.

Trabajo a Futuro

A partir de los resultados obtenidos en esta investigación, se identifican varias áreas de oportunidad para continuar desarrollando y ampliando el alcance del trabajo realizado:

■ Optimización de Modelos de Aprendizaje Profundo

- Utilizar una base de datos más afín a los reactores electroquímicos, considerando sus características específicas como geometrías particulares.
- Desarrollar modelos personalizados que incorporen estas características, mejorando la precisión en la estimación del flujo óptico y su aplicabilidad en este tipo de reactores.

■ Ampliación del Conjunto de Datos

- Ampliar la base de datos utilizada en esta investigación, alcanzando al menos 20,000 pares de imágenes, similar al tamaño de otras bases de datos del estado

del arte. Este incremento permitirá entrenar modelos más robustos y mejorar su capacidad de generalización.

- Garantizar que el conjunto de datos ampliado incluya una mayor diversidad de condiciones operativas, geometrías y patrones de flujo para representar de manera más completa las características de los reactores químicos.

■ **Validación en Escenarios Reales**

- Comparar los resultados obtenidos con técnicas tradicionales como la Velocimetría por Imágenes de Partículas (PIV), para validar la precisión y confiabilidad del modelo en escenarios prácticos.
- Analizar las diferencias en los resultados entre el modelo desarrollado y las mediciones obtenidas por PIV, destacando las ventajas y posibles limitaciones de cada enfoque.

■ **Exploración de Nuevas Tecnologías**

- Incorporar redes neuronales avanzadas, como transformers o modelos basados en atención, que han demostrado un alto rendimiento en tareas similares.
- Investigar la combinación de técnicas de flujo óptico con simulaciones CFD (Computational Fluid Dynamics) para desarrollar modelos híbridos más precisos.

El desarrollo de estas líneas futuras no solo consolidará los avances logrados en esta investigación, sino que también permitirá una mayor integración de tecnologías inteligentes en la industria química, fortaleciendo su competitividad y sostenibilidad en el largo plazo.

Bibliografía

- [1] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3061–3070.
- [2] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [3] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black, “Lessons and insights from creating a synthetic optical flow benchmark,” in *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, ser. Part II, LNCS 7584, A. Fusiello et al. (Eds.), Ed. Springer-Verlag, Oct. 2012, pp. 168–177.
- [4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open source movie for optical flow evaluation,” in *European Conf. on Computer Vision (ECCV)*, ser. Part IV, LNCS 7577, A. Fitzgibbon et al. (Eds.), Ed. Springer-Verlag, Oct. 2012, pp. 611–625.
- [5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *IEEE International Conference on Computer*

- Vision (ICCV)*, 2015. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15>
- [6] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," in *CVPR*, 2019.
- [7] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2017/IMKDB17>
- [8] D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Gusefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner, *et al.*, "The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 19–28.
- [9] S. Cai, S. Zhou, C. Xu, and Q. Gao, "Dense motion estimation of particle images via a convolutional neural network," *Experiments in Fluids*, vol. 60, pp. 1–16, 2019.
- [10] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 402–419.
- [11] C. Lagemann, K. Lagemann, S. Mukherjee, and W. Schröder, "Deep recurrent optical flow learning for particle image velocimetry data," *Nature Machine Intelligence*, vol. 3, no. 7, pp. 641–651, 2021.
- [12] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [13] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings*

of the *IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

- [14] T.-W. Hui, X. Tang, and C. C. Loy, “Liteflownet: A lightweight convolutional neural network for optical flow estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8981–8989.
- [15] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.
- [16] J. Hur and S. Roth, “Iterative residual refinement for joint optical flow and occlusion estimation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5754–5763.
- [17] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, Y. Xu, *et al.*, “Maskflownet: Asymmetric feature matching with learnable occlusion mask,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6278–6287.
- [18] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, “Learning to estimate hidden motions with global motion aggregation,” *arXiv preprint arXiv:2104.02409*, 2021.
- [19] L. Arenas and J. Smith, “Critical analysis of electrode materials for electrochemical applications,” *Journal of Electrochemical Science*, vol. 25, no. 2, pp. 120–135, 2020.
- [20] M. Gleede and E. Johnson, “Large-scale synthesis of organic compounds using flat electrodes,” *Journal of Chemical Engineering*, vol. 15, no. 3, pp. 250–265, 2019.
- [21] M. Lacasa and D. Thompson, “Environmental applications of flat electrodes in water treatment,” *Environmental Engineering Journal*, vol. 40, no. 4, pp. 300–315, 2019.
- [22] L. Arenas and R. Johnson, “Engineering applications of flat electrodes in batteries and fuel cells,” *Journal of Power Sources*, vol. 350, pp. 120–135, 2017.

- [23] F. F. Rivera, B. Miranda-Alcántara, G. Orozco, C. P. de León, and L. F. Arenas, "Pressure drop analysis on the positive half-cell of a cerium redox flow battery using computational fluid dynamics: Mathematical and modelling aspects of porous media," *Frontiers of Chemical Science and Engineering*, vol. 15, no. 2, pp. 399–409, 2021.
- [24] J. Hereijgers, J. Schalck, and T. Breugelmans, "Mass transfer and hydrodynamic characterization of structured 3d electrodes for electrochemistry," *Chemical Engineering Journal*, vol. 384, p. 123283, 2020.
- [25] J. Escudero-González and P. Amparo Lopez-Jimenez, "Redox cell hydrodynamic modelling: Towards real improved geometry based on cfd analysis," *Engineering Applications of Computational Fluid Mechanics*, vol. 8, no. 3, pp. 435–446, 2014.
- [26] R. A. Márquez-Montes, V. H. Collins-Martínez, I. Pérez-Reyes, D. Chávez-Flores, O. A. Graeve, and V. H. Ramos-Sánchez, "Electrochemical engineering assessment of a novel 3d-printed filter-press electrochemical reactor for multipurpose laboratory applications," *ACS Sustainable Chemistry & Engineering*, vol. 8, no. 9, pp. 3896–3905, 2020.
- [27] R. Hreiz, L. Abdelouahed, D. Fuenfschilling, and F. Lopicque, "Electrogenerated bubbles induced convection in narrow vertical cells: Piv measurements and euler–lagrange cfd simulation," *Chemical Engineering Science*, vol. 134, pp. 138–152, 2015.
- [28] L. F. Catañeda, F. F. Rivera, T. Pérez, and J. L. Nava, "Mathematical modeling and simulation of the reaction environment in electrochemical reactors," *Current Opinion in Electrochemistry*, vol. 16, pp. 75–82, 2019.
- [29] D. Honegger, P. Greisen, L. Meier, P. Tanskanen, and M. Pollefeys, "Real-time velocity estimation based on optical flow and disparity matching," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5177–5182.

- [30] G. de Croon, C. De Wagter, and T. Seidl, "Enhancing optical-flow-based control by learning visual appearance cues for flying robots," *Nature Machine Intelligence*, vol. 3, no. 1, pp. 33–41, 2021.
- [31] A. Eltner, H. Sardemann, and J. Grundmann, "Flow velocity and discharge measurement in rivers using terrestrial and unmanned-aerial-vehicle imagery," *Hydrology and Earth System Sciences*, vol. 24, no. 3, pp. 1429–1445, 2020.
- [32] A. Shields, S. V. S. Nagesh, C. Ionita, D. R. Bednarek, and S. Rudin, "Characterization of velocity patterns produced by pulsatile and constant flows using 1000 fps high-speed angiography (HSA)," in *Medical Imaging 2021: Biomedical Applications in Molecular, Structural, and Functional Imaging*, B. S. Gimi and A. Krol, Eds., vol. 11600, International Society for Optics and Photonics. SPIE, 2021, pp. 342 – 359. [Online]. Available: <https://doi.org/10.1117/12.2580888>
- [33] N. Afrashteh, S. Inayat, M. Mohsenvand, and M. H. Mohajerani, "Optical-flow analysis toolbox for characterization of spatiotemporal dynamics in mesoscale optical imaging of brain activity," *NeuroImage*, vol. 153, pp. 58–74, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811917302422>
- [34] J. M. Pardo, E. Moya-Albor, G. Ortega-Ibarra, and J. Brieva, "Freezing front velocity estimation using image processing techniques," *Measurement*, vol. 151, p. 107085, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0263224119309510>
- [35] F. Rivera, P. Hidalgo, F. Castañeda-Záldívar, I. Terol-Villalobos, and G. Orozco, "Phenomenological behavior coupling hydrodynamics and electrode kinetics in a flow electrochemical reactor. numerical analysis and experimental validation," *Chemical Engineering Journal*, vol. 355, pp. 457–469, 2019.
- [36] D. A. Gutiérrez-Jiménez, S. Salazar-Colores, F. F. Rivera, and J. T. López-Maldonado, "A novel method based on gunnar farneback method, mathematical morphology, and artificial vision for flow analysis in electrochemical reactors,"

in *International Congress of Telematics and Computing*. Springer, 2023, pp. 133–156.

- [37] J. T. López-Maldonado, S. Salazar-Colores, S. Piedra, and F. F. Rivera, “Effect of flow distributor configuration on the hydrodynamics in a multipurpose flow electrochemical reactor: Numerical analysis and experimental characterization employing digital image treatment,” *Industrial & Engineering Chemistry Research*, vol. 62, no. 7, pp. 3327–3337, 2023.
- [38] M. H. Zawawi, A. Saleha, A. Salwa, N. H. Hassan, N. M. Zahari, M. Z. Ramli, and Z. C. Muda, “A review: Fundamentals of computational fluid dynamics (cfd),” in *AIP conference proceedings*, vol. 2030, no. 1. AIP Publishing, November 2018.
- [39] D. Sun, S. Roth, and M. J. Black, “Secrets of optical flow estimation and their principles,” in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, June 2010, pp. 2432–2439.
- [40] F. S. Hassan and A. Gutub, “Improving data hiding within colour images using hue component of hsv colour space,” *CAAI Transactions on Intelligence Technology*, vol. 7, no. 1, pp. 56–68, 2022.
- [41] R. El-Khawaldeh, M. Guy, F. Bork, N. Taherimakhsousi, K. N. Jones, J. M. Hawkins, L. Han, R. P. Pritchard, B. A. Cole, S. Monfette, *et al.*, “Keeping an “eye” on the experiment: computer vision for real-time monitoring and control,” *Chemical Science*, vol. 15, no. 4, pp. 1271–1282, 2024.
- [42] P. A. Mendes, M. Mendes, A. P. Coimbra, and M. M. Crisóstomo, “Movement detection and moving object distinction based on optical flow,” in *Proceedings of the Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering, London, UK, 2019*, pp. 3–5.
- [43] Y. Song and Y. Kim, “Optical flow-based moving vehicle detection from single-pass worldview-3 imagery,” in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023, pp. 6466–6469.
- [44] S. Dong, P. Wang, and K. Abbas, “A survey on deep learning and its applications,” *Computer Science Review*, vol. 40, p. 100379, 2021.

- [45] X. Zhang, X. Zhang, and W. Wang, “Convolutional neural network,” in *Intelligent Information Processing with Matlab*. Springer, 2023, pp. 39–71.
- [46] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *International journal of computer vision*, vol. 56, pp. 221–255, 2004.
- [47] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical flow,” *International journal of computer vision*, vol. 92, pp. 1–31, 2011.
- [48] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open source movie for optical flow evaluation,” in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*. Springer, 2012, pp. 611–625.
- [49] M. Contributors, “MMFlow: Openmmlab optical flow toolbox and benchmark,” <https://github.com/open-mmlab/mmlflow>, 2021.