



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN  
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS (IIMAS)

ESTIMACIÓN DE FLUJO ÓPTICO EN IMÁGENES DE ULTRASONIDO CARDIACO  
CON REDES NEURONALES

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:

IVAN GAMALIEL CASASOLA RODRIGUEZ

TUTOR PRINCIPAL:

DRA. JIMENA OLVERES MONTIEL

Facultad de Ingeniería

Ciudad Universitaria, CD. MX, Noviembre 2022



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Dedicatoria ...*

A mi madre, quién siempre ha sido principal animadora, apoyo, soporte y fuente de motivación para superar cualquier reto. Gracias por creer siempre en mi.

# Agradecimientos

Se hace un extenso reconocimiento a los proyectos UNAM PAPIIT subsidios TA101121 y IV100420 y también a CONACYT por el apoyo brindado para la investigación de este proyecto durante dos años por medio de la beca con cvu 1083294.



# Notación

# Índice general

<b>Agradecimientos</b>	<b>II</b>
<b>Notación</b>	<b>III</b>
<b>1 Introducción</b>	<b>1</b>
§1.1 Definición del problema . . . . .	1
§1.2 Justificación . . . . .	2
§1.3 Objetivos . . . . .	2
§1.4 Estructura del documento . . . . .	3
<b>2 Estimación de flujo óptico</b>	<b>4</b>
§2.1 Métodos tradicionales . . . . .	4
§2.2 Métodos basados en redes neuronales . . . . .	5
§2.3 Métricas de evaluación y codificación de Flujo Óptico . . . . .	6
§2.4 Arquitectura GMA . . . . .	7
§2.4.1 Arquitectura RAFT . . . . .	7
§2.4.2 Modulo GMA . . . . .	9
§2.5 Métodos de interpolación en flujo óptico . . . . .	10

ÍNDICE GENERAL	IV
§2.5.1 EpicFlow . . . . .	11
§2.5.2 RIC . . . . .	11
§2.6 Estimación de movimiento en imágenes médicas . . . . .	12
<b>3 Base de datos sintética de imágenes de ultrasonido</b>	<b>15</b>
§3.1 Preprocesamiento de los datos . . . . .	17
§3.2 Creación de <i>ground truth</i> . . . . .	18
§3.2.1 <i>Ground truth</i> con EpicFlow . . . . .	19
§3.2.2 <i>Ground truth</i> con RicFlow . . . . .	20
§3.2.3 Post procesamiento . . . . .	20
§3.3 Región de evaluación . . . . .	22
<b>4 Estimación de flujo óptico con GMA</b>	<b>24</b>
§4.1 Características y resultados de entrenamientos . . . . .	25
§4.1.1 Entrenamiento con Edge Aware Interpolation . . . . .	26
§4.1.2 Entrenamiento con Robust Interpolation of Correspondances . . . . .	28
§4.2 Resultados en MPI Sintel . . . . .	30
§4.3 Resultados en base sintética de ultrasonido . . . . .	32
§4.4 Comparación con métodos tradicionales . . . . .	36
§4.5 Evaluación en otros datasets . . . . .	38
§4.6 Estimación en imágenes médicas reales . . . . .	40
<b>5 Conclusiones</b>	<b>44</b>
§5.1 Trabajo a futuro . . . . .	45

# Capítulo 1

## Introducción

### 1.1. Definición del problema

La importancia del movimiento de los órganos del cuerpo se debe a que con este mismo, los médicos son capaces de detectar anomalías que pueden llegar a ser determinantes para el tratamiento de tumores, realizar diagnósticos e incluso preparar adecuadamente un procedimiento quirúrgico. Un correcto funcionamiento de los órganos, tales como el corazón, los pulmones, hígado entre otros, puede ser visualizado mediante su movimiento, es por eso que el estudio de este mismo resulta relevante [1].

A lo largo de los años se han propuesto diversos métodos que permiten realizar la estimación de movimiento y que se han ido perfeccionando para tener una mejor precisión incluso en imágenes médicas, como lo son los ultrasonidos y las tomografías. Sin embargo, algunos de estos métodos (tradicionales) no resultan tan precisos y los que lo son, suelen tardar mucho tiempo en una sola estimación, lo cuál no los hace apropiados para tareas a tiempo real, como lo es una estimación de movimiento cardíaco en una secuencia de ultrasonido. Durante la última década los métodos basados en redes neuronales han adquirido mucha importancia por su buen desempeño realizando tareas de visión computacional, sin embargo, una de las desventajas de querer usar redes neuronales para la tarea de estimación de movimiento es que no se cuenta con los datos y su respectivo *ground truth* o movimiento real en imágenes médicas, principalmente de ultrasonido cardíaco, lo cuál resulta un problema importante para utilizar esta aproximación.

## 1.2. Justificación

La cardiopatía isquémica es considerada la principal causa de muerte a nivel mundial siendo responsable del 16 % del total de muertes. Desde el 2000 el mayor incremento de muertes se ha debido a esta enfermedad, alcanzando más de 2 millones hasta 8.9 millones de muertes en 2019. Aunado a esto, se sabe que las insuficiencias cardíacas congestivas se han incrementado también a nivel mundial debido a fallas en los ventrículos tanto izquierdo como derecho. Cerca de 5 millones de pacientes sufren de este padecimiento y otros 500 mil lo desarrollan cada año en Estados Unidos. Anteriormente se sabía que esto era debido al bombeo insuficiente del ventrículo izquierdo (insuficiencia ventricular sistólica). Ahora, estudios recientes han mostrado una mayor importancia en la insuficiencia ventricular diastólica, donde el proceso sistólico parece ser normal pero el proceso diastólico se ve afectado, siendo la causa del 50 % de las insuficiencias cardíacas congestivas en los pacientes. Por esta razón, resulta necesario un mejor entendimiento del movimiento normal y anormal de la contracción y relajación ventricular para dar mejores tratamientos a estos padecimientos [2, 3].

Para entender el movimiento, se requiere estudiar y analizar directamente las imágenes cardíacas comúnmente utilizadas en medicina, dentro de ellas se encuentran las tomografías computarizadas (CT, por sus siglas en inglés), este tipo de imágenes suelen ser muy limpias pero invasivas ya que involucran radiación hacia el cuerpo; las imágenes de ultrasonido, son un tipo de imágenes menos invasivas puesto que no requieren radiación, sin embargo suelen ser más ruidosas y no permiten obtener distintos cortes de la sección del corazón a diferencia de las imágenes de CT que sí lo permiten.

A la tarea de estimar movimiento en secuencias de imágenes se le conoce como *flujo óptico*, por lo tanto esta tarea resulta necesaria de aplicar en el tipo de imágenes médicas ya antes mencionadas para su análisis.

## 1.3. Objetivos

Como principales objetivos para este trabajo se tienen:

- Crear y entrenar un modelo que permita estimar flujo óptico en imágenes de ultrasonido

de corazón.

- El modelo además de ser altamente preciso, debe ser capaz de realizar estimaciones en el menor tiempo posible, para que pueda ser utilizado en el ámbito médico a tiempo real.
- Realizar una comparación del modelo propuesto con otros métodos tanto en precisión como en tiempo de la estimación.
- Evaluar el modelo propuesto en otro tipo de imágenes.

## 1.4. Estructura del documento

Este trabajo consta de 5 capítulos, los cuales se dividen en: **Capítulo 1**, donde se ha descrito la definición del problema y los objetivos principales de este trabajo; en el **Capítulo 2**, se da una breve introducción al problema de estimación de flujo óptico y en particular en imágenes médicas (ultrasonido cardiaco), posteriormente algunos antecedentes y métodos propuestos para resolver esta tarea y el estado del arte de esta misma; en el **Capítulo 3**, se habla acerca de la base de datos sintética de ultrasonidos de corazón utilizada, con el fin de tener un ground truth para el entrenamiento y evaluación del modelo propuesto; en **Capítulo 4**, se detallan los resultados de este modelo y se hace el análisis correspondiente del rendimiento del mismo en los diferentes conjuntos de datos utilizados, incluidas las imágenes de ultrasonido entre otras; finalmente en el **Capítulo 5**, se tienen las conclusiones de este trabajo junto con el trabajo a futuro para mejorar la tarea de estimación de movimiento en imágenes médicas y el modelo pueda ser de utilidad en el ámbito de la medicina.

# Capítulo 2

## Estimación de flujo óptico

El proceso de estimar flujo óptico consiste principalmente en estimar vectores del movimiento aparente de los píxeles en una imagen a lo largo de una secuencia, por lo que, para resolver este problema muchos métodos tanto tradicionales así como métodos basados en redes neuronales se han desarrollado durante los últimos años.

### 2.1. Métodos tradicionales

Tradicionalmente la estimación de flujo óptico ha sido abordada por métodos de aproximación diferencial y tomando en cuenta la suposición de que la intensidad en los píxeles de una imagen a lo largo de la secuencia, es constante, tal es el caso del algoritmo propuesto por Lucas-Kanade [4], el cual realiza una aproximación en series de Taylor a primer orden (lineal), obteniendo como resultado una ecuación diferencial de manera local que puede ser resuelta usando el método de mínimos cuadrados. Un método más eficiente fue propuesto por G. Farneback en [5], el cual está basado en una expansión polinomial en vez de lineal, como es el caso de Lucas-Kanade. Un método más robusto fue propuesto por T. Senst et al. [6] con su algoritmo RLOF (Robust Local Optical Flow), el cual considera una variación constante en la intensidad de los píxeles, a diferencia de otros métodos. Algunos otros métodos han utilizado una aproximación variacional desde el trabajo realizado por Horn y Schunck [7] y han convertido esta tarea en un problema de minimización de energía, dando como resultado un sistema de ecuaciones diferenciales (Euler-Lagrange), cuya forma tradicional de resolverlas es numérica-

mente. Todos estos métodos son precisos a la hora de estimar flujo óptico, sin embargo pueden ser algo complicados puesto que requieren resolver un problema de optimización de manera manual. Algunos otros de estos métodos también suelen ser muy lentos al momento de realizar una estimación, debido al gran número de iteraciones que estos requieren para converger a un mínimo.

## 2.2. Métodos basados en redes neuronales

En la era del *deep learning*, muchas arquitecturas han surgido durante los últimos años para resolver el problema de estimación de flujo óptico. Las primeras arquitecturas usaron redes neuronales, así como una capa de correlación que ayudara a la red neuronal a aprender mucho mejor las relaciones entre los píxeles de dos cuadros consecutivos, tal es el caso de FlowNet [8]. Una versión mejorada de esta misma arquitectura fue propuesta por E. Ilg et al. [9], en la que implementaron una arquitectura apilada que incluye una deformación en el segundo cuadro, así como también una subred neuronal especializada para desplazamientos pequeños. A pesar de que estas arquitecturas tenían buenos resultados, la cantidad de parámetros a entrenar seguía siendo muy grande, lo cual las hacía arquitecturas que tomaban mucho tiempo para su entrenamiento. Un modelo con menos parámetros fue propuesto en [10] (PWC-Net), la cual era mucho más pequeña en tamaño y el tiempo de entrenamiento era mucho menor a comparación de las arquitecturas previas, esta nueva propuesta usaba un esquema piramidal y de deformación junto con un volumen de costo, logrando mejores resultados que eran comparables con estado del arte. Modelos más recientes han superado por mucho a esta última arquitectura y por consiguiente a los modelos previos, tal es el caso de la arquitectura RAFT [11], Z. Teed y J. Deng usaron un volumen de correlación y una unidad recurrente (GRU) como un mecanismo iterativo para alcanzar una mejor estimación en el flujo óptico, puesto que esto simulaba a los procesos iterativos de optimización de los métodos tradicionales. Otras nuevas arquitecturas han surgido basándose en la arquitectura RAFT, dentro de estas se encuentra la arquitectura GMA (Global Motion Aggregation) [12], la cual agregó un nuevo módulo para tratar el problema de las oclusiones, presentado en muchos casos de una secuencia de imágenes. S Jian et al. fueron los primeros en usar *transformers* en el problema de flujo óptico, esto debido al

reciente éxito que han tenido en tareas de procesamiento de lenguaje natural para modelar dependencias en un rango amplio.

### 2.3. Métricas de evaluación y codificación de Flujo Óptico

Para evaluar un modelo o un método en la tarea de estimación de flujo óptico, comúnmente se usan tanto el error de punto a punto (*EPE*), como el error angular (*AE*) para medir de forma cuantitativa la precisión en la estimación. Dado que esta tarea principalmente consiste en estimar vectores, es posible realizar mediciones tomando la distancia entre ambos, lo cual hace el *EPE*, y a su vez se denota por la ecuación 2.1, mientras que el *AE* se denota por la ecuación 2.2, la cual mide el ángulo entre el vector estimado y el vector real, en lugar de la distancia. Ambas métricas denotan una forma de medir la precisión de un modelo o método de estimación de flujo óptico.

$$EPE = \|V_{est} - V_{gt}\| \quad (2.1)$$

$$AE = \arccos(V_{est} \cdot V_{gt}) \quad (2.2)$$

Además de medir de forma cuantitativa, también es posible realizar una comparación de la estimación de forma cualitativa, y para ello se usa la codificación a color del flujo óptico a partir de la rueda de color mostrada en la Figura 2.1. En la que cada color denota una dirección del movimiento del vector y cada tonalidad, indica la intensidad de ese vector, de esta forma es posible visualizar la precisión en la estimación de flujo óptico, obteniendo un mapa de color de la imagen original y así poder realizar un análisis cualitativo y por región.



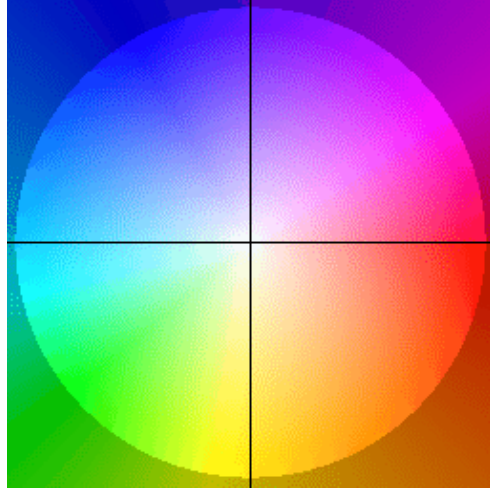


Figura 2.1: Rueda de color para codificación de Flujo Óptico. Rojo, indica desplazamientos hacia la derecha; azules, hacia la izquierda; morados, hacia arriba; amarillos, hacia abajo, etc.

## 2.4. Arquitectura GMA

### 2.4.1. Arquitectura RAFT

Como se ha mencionado anteriormente, la arquitectura GMA está basada en una arquitectura previa denominada RAFT (Recurrent All-Pairs Field Transforms). Esta arquitectura consta de 3 principales componentes, los cuales son:

1. Codificador de características y contexto, los cuales se denotan con  $g_\theta$  (codificador de características), esta es aplicada a ambos cuadros, y  $h_\theta$  (codificador de contexto), esta sólo se aplica al primer cuadro. Ambas tienen la misma estructura y cumplen lo siguiente, siendo  $W$  y  $H$ , el ancho y la altura original del par de imágenes de entrada:

$$g_\theta, h_\theta : \mathbb{R}^{W \times H \times 3} \rightarrow \mathbb{R}^{W/8 \times H/8 \times D}$$

2. Volumen de correlación 4D, el cual se obtiene con el producto punto de aplicar el codificador de características a los dos cuadros, por lo que las dimensiones de este volumen son  $W \times H \times W \times H$ .

Este volumen permite almacenar las similitudes visuales que hay entre dos cuadros con-

secutivos, dicho volumen es calculado por medio de la siguiente ecuación:

$$C_{ijkl} = \sum_h g_\theta(I_1)_{ijh} \cdot g_\theta(I_2)_{kjh} \quad (2.3)$$

Después de obtener este volumen se aplica un esquema piramidal con el fin de tener volúmenes a distintas escalas, esto con el fin de poder realizar estimaciones precisas tanto para desplazamientos largos como cortos, para ello se aplica un *pooling* a lo largo de las últimas dos dimensiones (W, H de la segunda imagen) y esto se hace con la mitad de la escala, un cuarto y un octavo de la dimensión original, tal y como se muestra en la Figura 2.2.

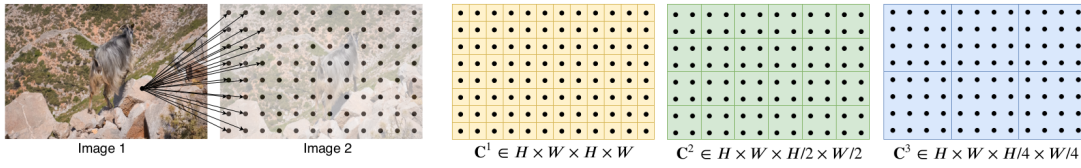


Figura 2.2: Esquema piramidal para el volumen de correlación. (Imagen tomada de [11])

3. Operador de actualización del flujo óptico con una GRU de forma iterativa o recurrente, esta actualización se obtiene tomando como entrada ( $x_t$ ) a la unidad recurrente (GRU), la concatenación de un flujo inicial (cero), el volumen de correlación y las características de contexto, posterior a eso se calculan los términos expresados en las siguientes ecuaciones que dan como resultado la actualización residual que se va ir sumando de manera recurrente.

$$z_t = \sigma(\text{Conv}_{3 \times 3}([h_{t-1}, x_t], W_z)) \quad (2.4)$$

$$r_t = \sigma(\text{Conv}_{3 \times 3}([h_{t-1}, x_t], W_r)) \quad (2.5)$$

$$\hat{h}_t = \sigma(\text{Conv}_{3 \times 3}([r_t \odot h_{t-1}, x_t], W_h)) \quad (2.6)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \quad (2.7)$$

De modo que a este nuevo estado se le aplica una nueva capa convolucional que producirá una nueva actualización del flujo.

Finalmente la función de pérdida está dada por la ecuación:

$$L = \sum_{i=1}^N \gamma^{N-i} \|f_{gt} - f_i\|_1 \quad (2.8)$$

Donde el valor de  $\gamma$  está fijo con un valor de 0.8.

La arquitectura completa de RAFT se muestra en la Figura 2.3, ya con todos estos componentes ensamblados.

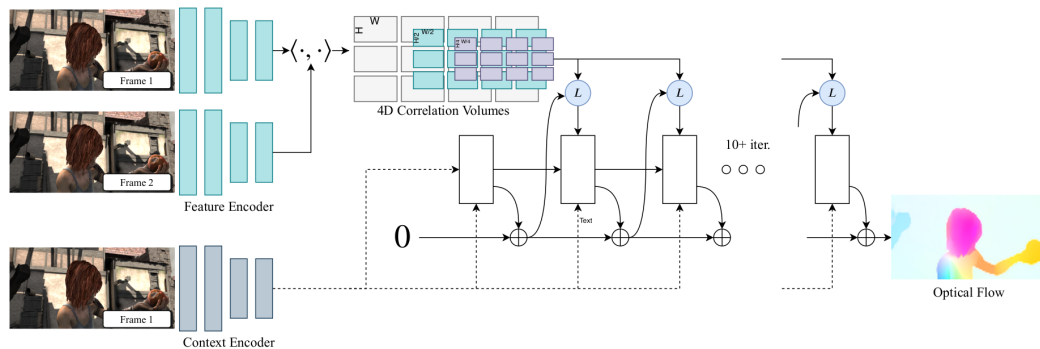


Figura 2.3: Arquitectura RAFT. (Imagen tomada de [11])

### 2.4.2. Modulo GMA

Ahora en la arquitectura GMA a diferencia de RAFT, busca solucionar un problema en particular que es el tema de oclusiones, para lo cual hace uso de *transformers* por su capacidad de crear relaciones en rangos amplios como se ha visto en tareas de procesamiento de lenguaje natural. Esto con el fin de crear relaciones entre píxeles de manera no-local y poder transferir información a los píxeles ocluidos a partir de otros que no lo están. La arquitectura completa de GMA se muestra en la Figura 2.4 mientras que el módulo de GMA se observa en la Figura 2.5.

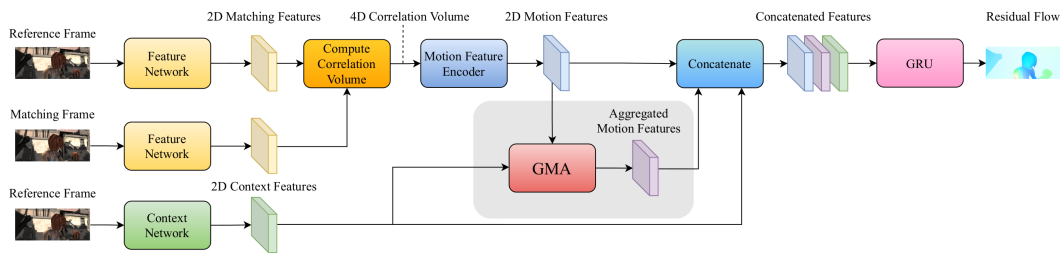
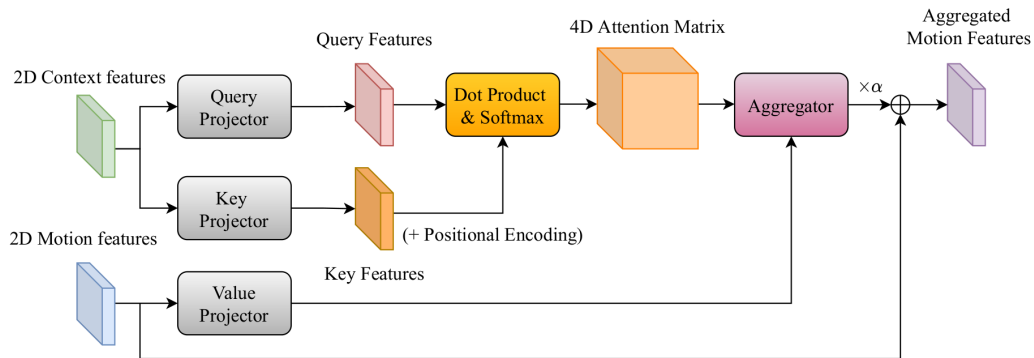


Figura 2.4: Arquitectura GMA. (Imagen tomada de [12])

Figura 2.5: Módulo GMA empleando *transformers*. (Imagen tomada de [12])

## 2.5. Métodos de interpolación en flujo óptico

Algunos métodos tradicionales de flujo óptico realizan una estimación del movimiento en ciertas regiones específicas de una imagen, a esto se le conoce como *flujo óptico disperso* y por medio de métodos de interpolación cubren la estimación del movimiento en todos los puntos de la imagen, lo cual se conoce como *flujo óptico denso*. En varias ocasiones no es del todo posible estimar movimiento en cada punto de una imagen ya que de ser así, los métodos de estimación tardarían mucho más tiempo en realizar el flujo óptico, puesto que los procesos se deben aplicar a todos los puntos de la imagen. Algo más viable es realizar la estimación del flujo óptico disperso y posteriormente con alguna interpolación, obtener el flujo óptico denso. Por esta razón, la precisión en la estimación dependerá de que tan buena es la aproximación de los métodos de interpolación. Algunos de los métodos de interpolación que han mostrado mejores resultados se describen a continuación.

### 2.5.1. EpicFlow

Este método (Edge-Preserving Interpolation of Correspondances) fue propuesto por J. Revaud et al. en [13] y consta principalmente de dos pasos para estimar el flujo óptico denso: 1) Una estimación de correspondencias de punto a punto densa, a partir de una interpolación y preservando los bordes de un conjunto de correspondencias entre puntos de manera dispersa; 2) Un refinamiento variacional de minimización de energía, que se inicializa con la estimación densa. La interpolación, de dispersa a densa, recae en una elección apropiada de una función de distancia. En la Figura 2.6 se muestra el diagrama del proceso completo de obtención de la estimación densa usando EpicFlow.

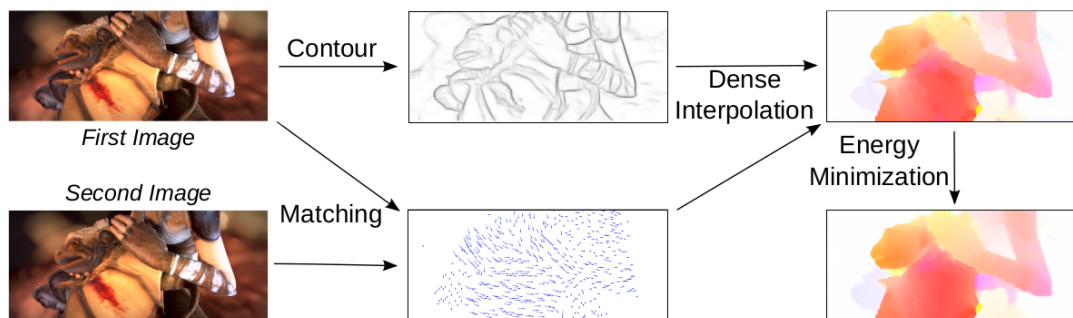


Figura 2.6: Panorama general de EpicFlow. Dadas dos imágenes con un conjunto de correspondencias entre puntos y obteniendo los bordes de la primer imagen, se calcula la interpolación densa y posteriormente con el proceso de minimización de energía se refina el flujo óptico estimado. (Imagen tomada de [13])

### 2.5.2. RIC

A pesar de las ventajas de EpicFlow de preservar los bordes y su eficiencia, seguía teniendo algunas vulnerabilidades para crear correspondencias con imágenes ruidosas. Un método más robusto para atacar este problema se presentó en [14] por Y. Hu et al. (RicFlow), en el cual la primer imagen es segmentada en superpíxeles, donde posteriormente se hace una estimación robusta con los vecinos cercanos en cada superpíxel y utilizando un mecanismo de propagación sobre estos mismos. La descripción de este algoritmo se presenta a continuación:

```

1 Entrada: Imagenes I1,I2, conjunto M de correspondencias entre imagenes
2 Salida: Campo denso de correspondencias F: I1-->I2
3 Segmentacion de I1 en K superpíxeles {Sk}
4 Inicializacion de flujo en superpíxeles con el conjunto M
5 Construccion de grafica G=(V,E) de superpíxeles
6 for cada superpixel en Sk do
7     Se inicializa el modelo con base en el flujo del superpixel
8     Obtener los vecinos de soporte Nn(k) del superpixel
9     Extraccion interior basada en el modelo de propagacion
10    Optimizacion en los interiores con minimos cuadrados ponderados
11 Refinamiento variacional

```

De esta manera RicFlow logra imponerse ante el método EpicFlow, logrando ser más tolerante ante imágenes de entrada con ruido y a su vez usando superpíxeles para segmentar todo la escena bajo la suposición que la mayoría de las escenas de interés consisten en regiones con un patrón de movimiento consistente.

## 2.6. Estimación de movimiento en imágenes médicas

Un método bastante preciso en estimación de flujo óptico en imágenes cardiacas de tomografía computarizada (CT), fue propuesto en [15] por E. Moya-Albor et al. (SHT), el cual se basa en la descomposición polinomial de cada imagen usando la transformada orientada de Hermite como una representación local de características de la imagen. Este modelo está inspirado en las primeras etapas del sistema de visión humana, lo que se conoce como el traslape Gaussiano de los campos receptivos. Usando una ventana Gaussiana se obtienen los coeficientes de Hermite de la descomposición polinomial de modo que se llega una expresión para la energía usando estos coeficientes y las restricciones correspondientes a la intensidad constante, de los coeficientes de Hermite y de suavidad. Posteriormente se resuelve esta expresión de forma numérica resolviendo las ecuaciones de Euler-Lagrange o por un proceso iterativo.

Como método de estimación, es preciso y esto se puede observar en los resultados de [15] mostrados en la Figura 2.7, donde se ha evaluado su modelo usando la secuencia de *Dimetrodon*, una de las secuencias más utilizadas para evaluar métodos y modelos de estimación de flujo óptico, obteniendo un valor bajo para el *AE*. Por otro lado, en la Figura 2.8 se muestra finalmente



los resultados de estimar el flujo óptico en imágenes de CT, logrando buenos resultados en este tipo de imágenes.

A pesar de los buenos resultados que este método puede proporcionar, tiene una deficiencia, la cual no lo hace un método apropiado para usarse en el sector médico, y es que el tiempo de obtención de una estimación es de alrededor de 15 minutos. Es por eso que, el tiempo de estimación de flujo óptico resulta importante para crear aplicaciones útiles para la estimación de movimiento a tiempo real en el ámbito médico y permitir un diagnóstico rápido y preciso con base en los resultados.

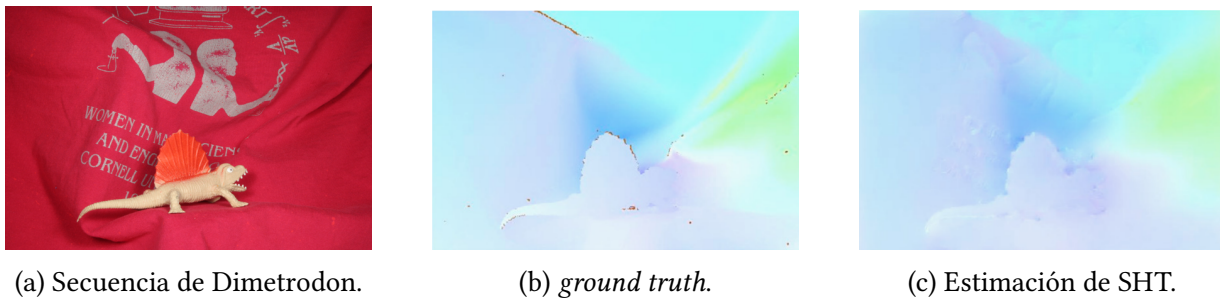


Figura 2.7: Resultado del método SHT en secuencia de *Dimetrodon* con un  $AE= 2.7^\circ$ . (Imágenes tomadas de [15])

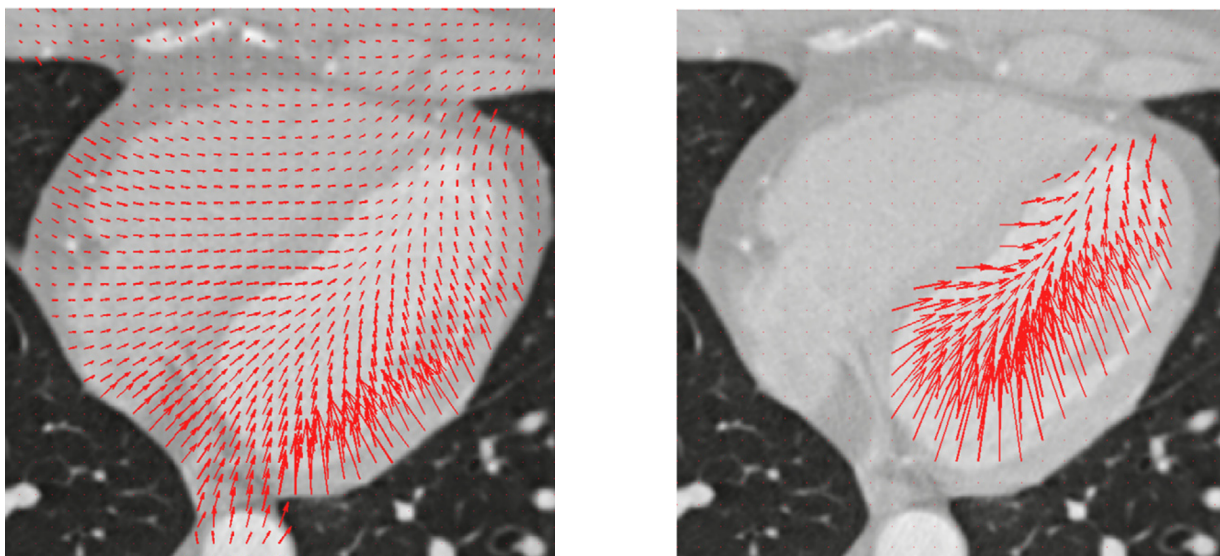


Figura 2.8: Estimación de flujo óptico con SHT en imagen de CT. (Imágenes tomadas de [15])

Otro de los principales problemas de realizar esta tarea, es tratándose de imágenes de ultrasonido, dado que suelen ser imágenes muy ruidosas debido al *speckle*, esto hace que la tarea se

vuelva un poco más difícil de resolver, sin embargo, métodos tradicionales como [15], resuelven bien esta tarea, pero nuevamente el tiempo de estimación de flujo óptico ronda entre los 15 y 20 minutos en un solo ejemplo, lo cual, confirma que es un método no adecuado para resolver la tarea a tiempo real. Ahora, cuando se habla de modelos basados en redes neuronales resulta difícil entrenarlos dado que no se cuenta con conjuntos amplios con *ground truth* para realizar los entrenamientos y las evaluaciones de los modelos. Es por eso que, en este trabajo uno de los principales objetivos es crear un *ground truth* en imágenes de ultrasonido, para entrenar modelos basados en redes neuronales, del cual se discutirá en el siguiente capítulo.



## Capítulo 3

# Base de datos sintética de imágenes de ultrasonido

Una solución alterna al problema de la falta de ground truth en imágenes cardíacas surge a partir del trabajo de M. Alessandrini et al. [16], donde proveen una base de datos de libre acceso con secuencias sintéticas de ultrasonido cardíaco, para algoritmos de rastreo de *speckle*. Para lo cual, el movimiento sintético cardíaco fue obtenido a partir de un modelo de corazón electro-mecánico complejo, mientras que la textura realista específica de los proveedores fue obtenida muestreando una grabación de ultrasonido clínico real. En esta base de datos, han generado 105 secuencias sintéticas que abarcan: patrones de movimiento de casos sanos y casos con isquemia; las orientaciones más comunes del transductor, que dan como resultado las vistas a 2, 3 y 4 cámaras; y calidad de imagen específica del proveedor provenientes de 7 diferentes sistemas. En cada imagen de las secuencias sintéticas se han obtenido un conjunto de puntos de 36x5, es decir, 36 filas donde cada una está conformada por 5 puntos que delimitan el miocardio del ventrículo izquierdo, resultantes de la simulación. En la Figura 3.1 se muestran las características del miocardio en todo el conjunto de simulaciones generadas. Donde cada fila en la figura denota las vistas a 4, 3 y 2 cámaras de arriba hacia abajo respectivamente. Cada recuadro de la rejilla en las representaciones del miocardio corresponde a los segmentos generados por los 5 puntos y 36 filas. Las columnas en la figura, corresponden a los diferentes patrones de movimiento, la primer columna representa los casos sanos y las siguientes columnas son los casos de isquemia. Finalmente cada color corresponde al valor promedio de la deformación longitudinal

del miocardio sobre todas las simulaciones de todos los proveedores.

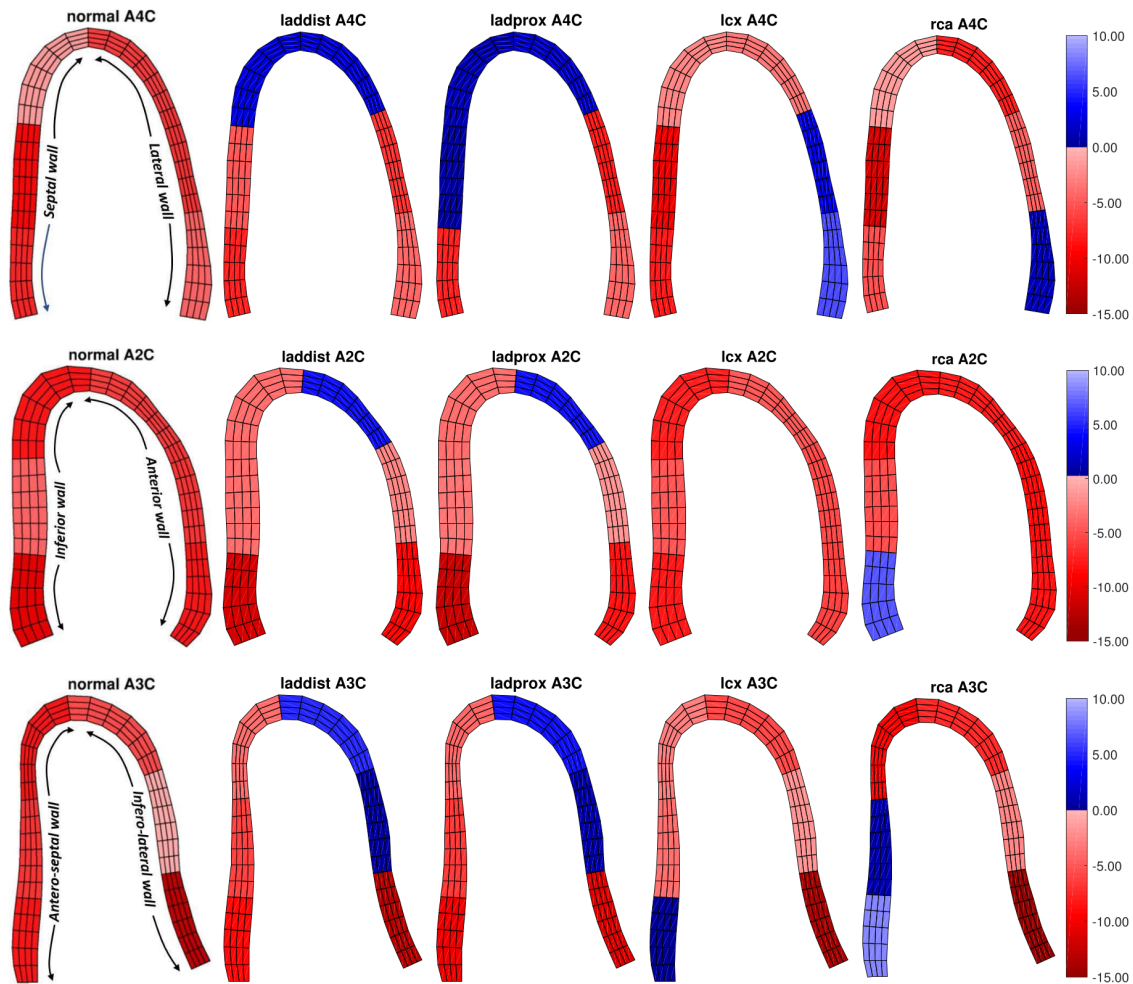


Figura 3.1: Representación del miocardio, en todo el conjunto de datos de todos los proveedores. (Imagen tomada de [16])

Dentro del conjunto de datos se tienen 7 proveedores (Esaote, Ge Vinged Ultrasound, Hitachi Aloka Medical Ltd, Philips Medical Systems, Samsung Medison Co, Siemens y Toshiba Mec US), cada uno contiene las secuencias simuladas de ultrasonido correspondientes a un caso sano (normal) y 4 diferentes casos de isquemia (laddist, ladprox, lcx y rca) para las vistas a 2, 3 y 4 cámaras, que dan como resultado 105 secuencias con cada una compuesta de 40 a 60 cuadros aproximadamente, dando como resultado un total de 6164 imágenes y 6060 pares de ejemplos (compuestos por dos cuadros consecutivos).

Dado que cada imagen tiene delimitada la pared ventricular por un conjunto de puntos de

36x5, es posible obtener el *ground truth* del movimiento de este, a lo largo de cada secuencia. Ahora, para poder entrenar un modelo de estimación de flujo óptico es necesario tener el *ground truth* en toda la imagen y no sólo en el miocardio, por lo que para generar dicho *ground truth* de manera completa, se puede aplicar cualquiera de los métodos de interpolación mencionados en el capítulo 2, considerando que únicamente hay movimiento en la región del abanico del ultrasonido.

### 3.1. Preprocesamiento de los datos

Cada una de las imágenes de todas las secuencias ha sido nombrada con el siguiente formato:

$$image\_ [id\_secuencia]\_ [proveedor]\_ [vista]\_ [caso]\_ [num\_cuadro].png$$

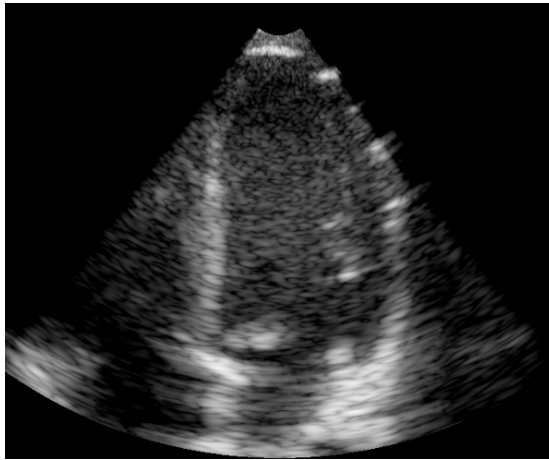
De modo que el *id\_secuencia* corresponde al número de imagen en general y es un valor único (es decir, se tienen 6164 valores); *proveedor* es el nombre de cada uno de los 7 proveedores que se tienen; *vista* corresponde a cualquiera de las vistas a 2, 3 y 4 cámaras; *caso* es el tipo de caso y puede ser ya sea sano o de isquemia; y *num\_cuadro* es el número de imagen dentro de la secuencia a la que pertenece.

Cada una de estas imágenes se ha reescalado a un tamaño de  $596 \times 494$ , la mayoría de los ejemplos tienen un tamaño similar a este, por lo que se busca homogeneizar todo el conjunto de datos. Debido a este reescalamiento, y considerando que cada imagen tiene asociados 180 puntos (36x5) correspondientes a la región de la pared del ventrículo izquierdo, un mapeo de cada uno de estos puntos fue necesario para obtener su nuevo valor en la imagen reescalada.

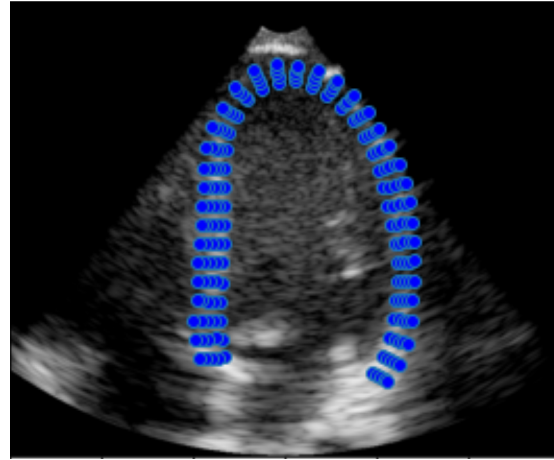
Los valores de cada uno de los puntos se obtienen de los archivos *ground\_truth.mat*, en el cual viene contenida la información de los 180 puntos; y el archivo *info.mat*, en el que se tiene la información del tamaño de la imagen. Estos archivos se encuentran en la página de la base de datos de libre acceso [17]. Algunos de los valores de estos puntos tienen valores *NaN*, por lo que fue necesario usar una interpolación usando los puntos vecinos y así obtener un valor numérico para todos los puntos.

En la Figura 3.2 se observa un ejemplo del resultado del reescalamiento de la imagen (Figura

3.2a) y el mapeo de cada uno de los puntos correspondientes a la región del miocardio (Figura 3.2b).



(a) Simulación de ultrasonido.



(b) Miocardio delimitado por los 180 puntos.

Figura 3.2: Imagen 46 correspondiente al proveedor Esaote, de 4 cámaras, caso normal (sano) y número de imagen 46 en la secuencia (image\_46\_esaote\_a4c\_normal\_46.png).

### 3.2. Creación de *ground truth*

Ahora bien, dado que se conoce cómo se desplaza cada uno de los 180 puntos de una imagen a lo largo de toda la secuencia, es posible obtener el flujo óptico de estos mismos, es decir, se puede obtener el *ground truth* en la región del miocardio. Sin embargo, un modelo de estimación de flujo óptico (denso) realiza la estimación de movimiento en toda la imagen, por lo que se requiere conocer el *ground truth* en toda la imagen para realizar un entrenamiento. Para esto, se usa la información conocida del *ground truth* y con ayuda de los métodos de interpolación vistos en el capítulo 2, se puede completar este mismo en toda la imagen de cada secuencia.

Para completar este *ground truth* una consideración muy importante es que únicamente hay movimiento dentro de la región del abanico del ultrasonido. Es por eso que es necesario ubicar esta región y fuera de ella se considera que no existe ningún movimiento. Lo cual se convierte en un problema de binarización y segmentación de dicha región, la forma de resolverlo ha sido considerando como ceros todos aquellos píxeles que nunca cambiaron de valor cero (negro) a lo largo de toda la secuencia y los que sí cambiaron, se consideran unos, de modo que para cada

secuencia se ha generado una máscara similar a la de la Figura 3.3, cada una generada a partir de las imágenes con conforman una secuencia de las 105 existentes.



Figura 3.3: Máscara generada para la secuencia de Esaote, 4AC, normal.

### 3.2.1. *Ground truth con EpicFlow*

Otro de los nombres con los que se conoce a este método es Edge Aware Interpolation (EAI) o interpolación consciente de los bordes. Para realizar la interpolación en cada una de las imágenes de cada secuencia se ha fijado el parámetro  $K$  con un valor de 128, que corresponde al número de puntos cercanos a un punto en cuestión, para realizar la interpolación sobre este. Los parámetros de suavidad también se han ajustado de manera que no haya cambios tan abruptos de movimiento entre regiones delimitadas por los distintos bordes.

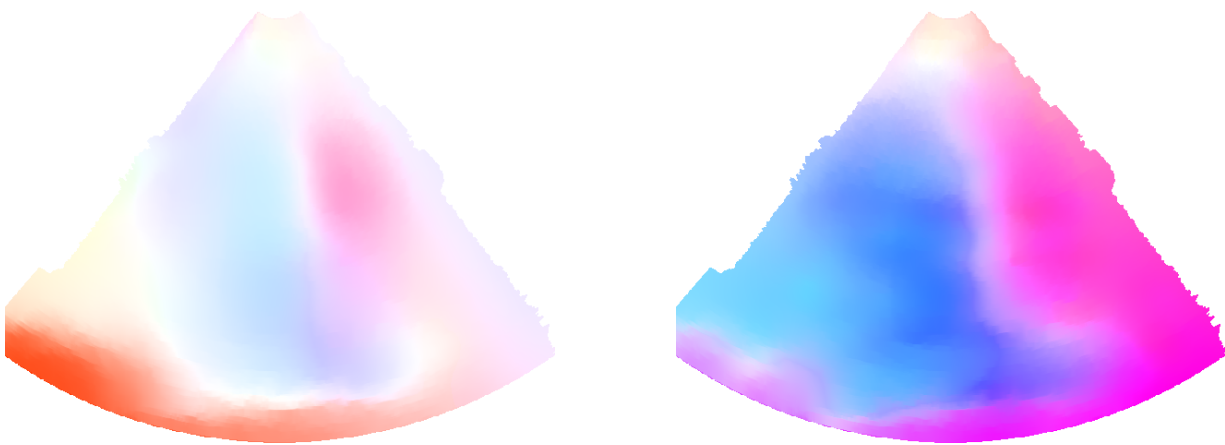
Una vez hecha la interpolación con este método, se hace una multiplicación de esta con la máscara correspondiente (como la de la Figura 3.3) de modo que sólo se conserva la interpolación de los puntos y por consiguiente el movimiento de estos únicamente en la región del abanico del ultrasonido, de tal manera que se obtiene un *ground truth* resultante para este método como el que se observa en la Figura 3.4a.

### 3.2.2. *Ground truth* con RicFlow

Para el caso de la interpolación con el método de RIC se ha usado un valor de  $K$  igual a 32, que corresponde al número de superpíxeles en los que se va a sobre segmentar la imagen. De igual manera, los parámetros de suavidad se han ajustado así como con el método de EAI. A diferencia del método anterior, RIC consta de otro parámetro  $\alpha$ , el cual funciona como un parámetro de peso global para transformar la distancia geodésica en un peso y el cual se ha fijado con un valor de 2.5, es decir, le estamos asignando un peso más grande ya que el valor por default es de 0.7.

Una vez hecha la interpolación, nuevamente se realiza la multiplicación por las máscaras correspondientes, de tal manera que se obtienen un *ground truth* similar a los de la Figura 3.4b.

En la Figura 3.4 se muestran los resultados con ambos métodos, teniendo movimientos más suaves y marcando de manera más significativa los bordes en la interpolación de EAI, mientras que en la interpolación de RIC se muestran movimientos más intensos y homogéneos.



(a) *Ground truth* resultante usando EAI.

(b) *Ground truth* resultante usando RIC.

Figura 3.4: Resultados de las interpolaciones para el número de imagen 46.

### 3.2.3. Post procesamiento

La intención de realizar un post procesamiento al *ground truth* obtenido, surge del hecho de haber fijado los parámetros en ambos métodos para todo el conjunto de datos, ya que para algunos, esta selección de valores no resulta ser la más adecuada, por lo que se tuvieron que

identificar estos datos a partir de la visualización del *ground truth* obtenido, donde en algunos casos visualmente no parecía ser correcta la estimación de flujo óptico. Para estos casos identificados se tuvo que hacer una adecuación en los valores de los parámetros hasta obtener un *ground truth* visualmente consistente con los de la imagen anterior y subsecuente. En la Figura 3.5 se muestra la diferencia en el *ground truth* obtenido de usar un valor de  $K$  igual a 128 y 8 respectivamente, usando el método de EAI. De igual manera en la Figura 3.6 se aprecia la diferencia de cambiar el valor del parámetro  $K$ , de manera que sea consistente con el *ground truth* obtenido de las imágenes anteriores, de esta forma se asegura un poco más la certeza en el resultado de la interpolación.

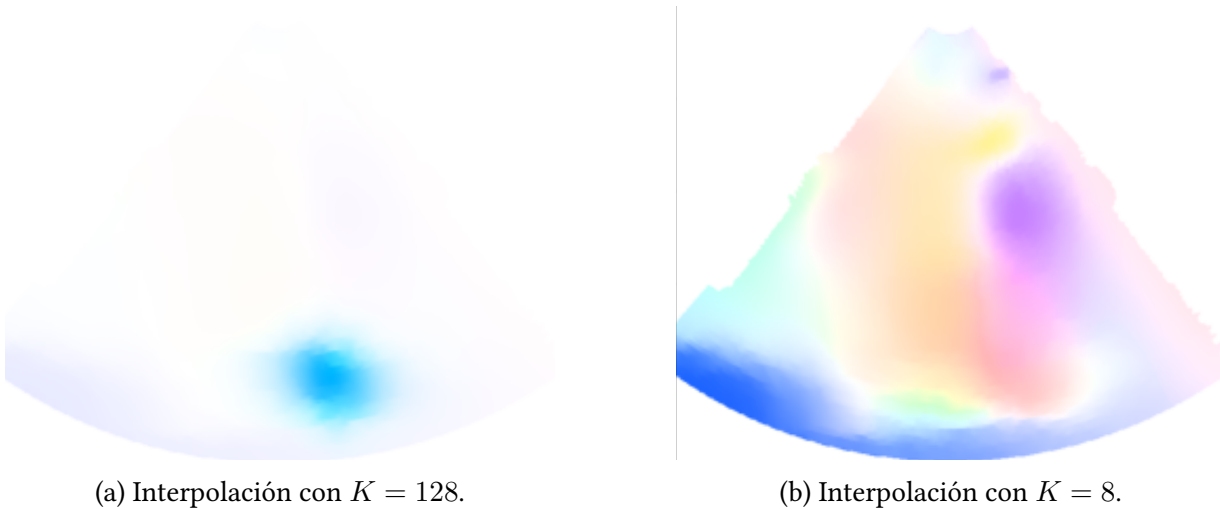


Figura 3.5: Resultados de las interpolaciones para el número de imagen 46.



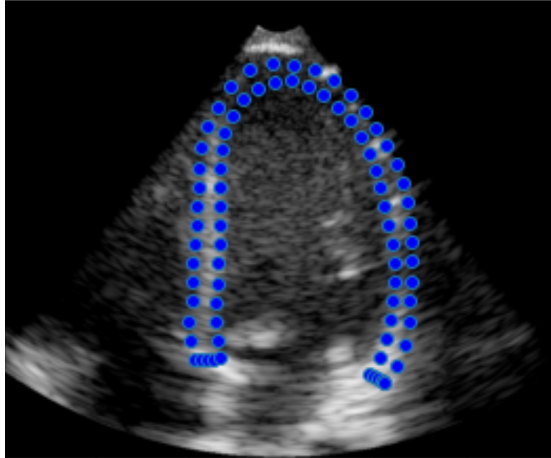
Figura 3.6: Resultados con y sin post procesamiento con el método de EAI.

### 3.3. Región de evaluación

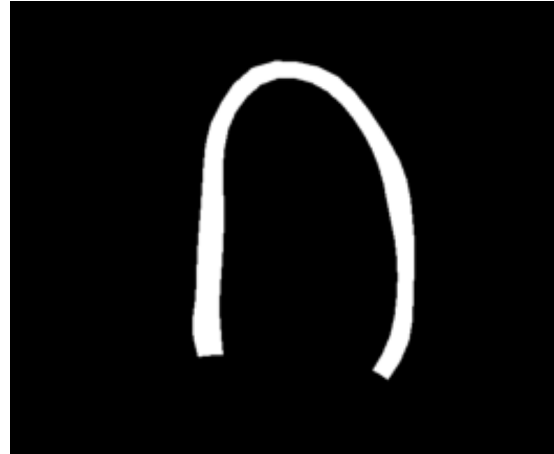
Los resultados obtenidos de las interpolaciones para generar un *ground truth* completo ahora son útiles para realizar un entrenamiento de modelo de flujo óptico, sin embargo, para evaluar estos mismos modelos o cualquier otro método de estimación de flujo óptico, no resulta tan conveniente usarlo de forma completa, ya que aquellas regiones donde se ha hecho interpolación, la certeza del movimiento es menor. Por esta razón, se busca evaluar la estimación de flujo óptico únicamente en aquellas regiones donde la certeza del movimiento es mayor, y esa región corresponde a la de la pared ventricular. Para ello, es necesario crear y localizar la región correspondiente, partiendo de tomar los puntos del borde de todo el miocardio como se muestra en la Figura 3.7a, con estos, posteriormente se realiza una interpolación poligonal de manera que se obtiene la región del polígono que delimita la región de la pared del ventrículo, creando así una máscara para dicha región donde los unos corresponden a la región deseada y los ceros la región que no es miocardio, como se muestra en la Figura 3.7b.

Sin embargo, la pared del ventrículo no es la única región donde hay mayor certeza en el movimiento, si no también en la región externa al abanico del ultrasonido, dado que esta también pertenece a la imagen completa, por lo tanto podemos considerar que no existe ningún movimiento fuera del abanico. Para localizar esta región, no resulta una tarea complicada puesto que ya se ha localizado la región del abanico en la Figura 3.3, por lo que la región comprendida





(a) Borde de la pared del ventrículo izquierdo.



(b) Máscara de pared ventricular.

Figura 3.7: Pared ventricular para la imagen 46.

por el exterior puede obtenerse de tomar la máscara inversa de esta última, obteniendo máscaras resultantes similares a la de la Figura 3.8a.

De esta manera, la región de evaluación total (en blanco) correspondería a la suma de las regiones de las Figuras 3.7b y 3.8a, dando como resultando la máscara como se muestra en la Figura 3.8b, la cual será utilizada para delimitar los puntos en los cuales se va a considerar la estimación de movimiento para la evaluación de los métodos y modelos de flujo óptico.



(a) Región externa al abanico.



(b) Región total resultante para evaluación.

Figura 3.8: Máscaras resultantes para la imagen 46.

# Capítulo 4

## Estimación de flujo óptico con GMA

La implementación de esta arquitectura fue tomada de [18], la cual está hecha en Pytorch, realizando algunas modificaciones sobre esta para crear el *Dataloader* para el conjunto completo de imágenes de ultrasonido y modificando la evaluación de la estimación de flujo óptico en la región de evaluación, que se mencionó en el capítulo 3. De los 6060 ejemplos en total y disponibles, 90 % de estos se tomaron para entrenamiento (5420) y el 10 % para validación (640).

Para mejores resultados en la estimación también se ha tomado el conjunto de datos de MPI Sintel [19], el cual es un conjunto extenso donde proveen el *ground truth* de algunas secuencias de imágenes que conforman pequeñas escenas de una película animada y el cual está dividido a la vez en dos subconjuntos: *Clean*, donde se encuentran las imágenes limpias y sin borrosidades y buena iluminación; y *Final*, en el que las imágenes se encuentran borrosas y con poca iluminación, por lo que en este último conjunto resulta un poco más complicado estimar el flujo óptico con precisión a diferencia del conjunto *Clean*. La ventaja de estos conjuntos de datos, es que se cuenta con el *ground truth* y esto permite a los modelos aprender a estimar mejor el movimiento, incluso aunque no sean datos relacionados con la tarea en cuestión, que es estimar el movimiento en imágenes de ultrasonido. Este conjunto también se dividió en 90 % para entrenamiento ( $\approx 1800$ ) y el 10 % para validación ( $\approx 200$ ). Las distribuciones en los conjuntos de entrenamiento y validación para ambos conjuntos de datos (Sintel e ultrasonidos) se han hecho de forma aleatoria. Ahora, dado que se han usado dos formas distintas para crear el *ground truth* en las imágenes de ultrasonido y para comparar qué método resulta más efectivo, es necesario realizar dos entrenamientos por separado tomando como *ground truth* uno con el

obtenido con RIC y el otro con EAI y así después evaluar los resultados.

## 4.1. Características y resultados de entrenamientos

Ambos entrenamientos fueron realizados en un servidor de la UNAM perteneciente al Laboratorio Avanzado de Procesamiento de Imágenes (LaPI), el cual cuenta con una GPU Titan RTX, que a su vez tiene 4608 cuda cores [20], lo que la hace una tarjeta gráfica con una gran capacidad para paralelizar procesos, en este caso el entrenamiento por batches de una red tan compleja como GMA, la cual cuenta con 5,879,873 de parámetros entrenables, cada uno de los entrenamientos tuvo una duración de 5 días aproximadamente.

Cada entrenamiento se ha realizado con las siguientes características:

- learning rate = 0.0001
- batch size = 8
- image size =  $256 \times 384$
- mixed precision = True
- num steps = 1,000,000
- weight decay = 0.00005
- iterations = 12
- val freq = 10,000

El *learning rate* (tasa de aprendizaje) corresponde al valor del salto en cada iteración del entrenamiento, donde se ha usado como optimizador el método de Adam (Adaptive Moment Estimation); *batch size*, es el número de ejemplos tomados del conjunto de entrenamiento que se procesan antes de actualizar el modelo; *image size*, es el tamaño al cual se van a reescalar todas las imágenes para homogeneizar todos los ejemplos provenientes de diferentes conjuntos de datos; *mixed precision* (precisión mixta) es usada para reducir el tiempo de entrenamiento realizando tantas operaciones como sea posible con la mitad de la precisión de punto flotante,

i.e usa flotantes de 16 bits en lugar de 32 que es el default de Pytorch; *num steps*, corresponde al número de veces que se van a procesar los datos del conjunto de entrenamiento; *weight decay*, es un hiperparámetro utilizado en el optimizador Adam, para calcular la pérdida agregando una penalización con la norma L2 a los pesos; *iterations*, es el número de iteraciones que va a realizar la unidad recurrente (GRU) para actualizar y refinar la predicción de flujo óptico; *val freq*, es el número de pasos o épocas durante el entrenamiento en las cuales se va a generar un checkpoint del modelo entrenado hasta ese momento, esto con el fin de quedarse con el mejor modelo terminado el entrenamiento.

#### 4.1.1. Entrenamiento con Edge Aware Interpolation

En la Figura 4.1 se muestran los resultados del entrenamiento, midiendo la evolución del EPE a lo largo de este mismo en el conjunto de imágenes de ultrasonido, tanto para el conjunto de entrenamiento y validación, los cuales muestran un comportamiento muy similar habiendo una ligera diferencia entre el error obtenido, teniendo a la curva del conjunto de entrenamiento por debajo de la curva del conjunto de validación. En la Figura 4.2 se observa nuevamente la comparación entre los resultados en el conjunto de entrenamiento y validación para los conjuntos de datos de Sintel, en la Figura 4.2b se puede ver que el conjunto de validación alcanza un valor más bajo en el error que el conjunto de entrenamiento, esto puede deberse a que el conjunto de validación en el dataset Final es muy pequeño y los resultados en este son buenos, mientras que la Figura 4.2a el conjunto de entrenamiento alcanza un EPE más bajo que el conjunto de validación. Finalmente en la Figura 4.3 se muestra la evolución del EPE en todo el conjunto de entrenamiento, el cual muestra un comportamiento muy oscilante pero decreciente, lo cual es un comportamiento esperado ya que esto llega a ocurrir cuando se usan batches, pese al comportamiento oscilante el error sigue decreciendo a lo largo del entrenamiento, lo cual indica que el modelo entrenado sí aprendió a realizar la tarea en cuestión, donde el mejor modelo se ha obtenido en el último paso (época), es decir, en la época  $10^6$ .

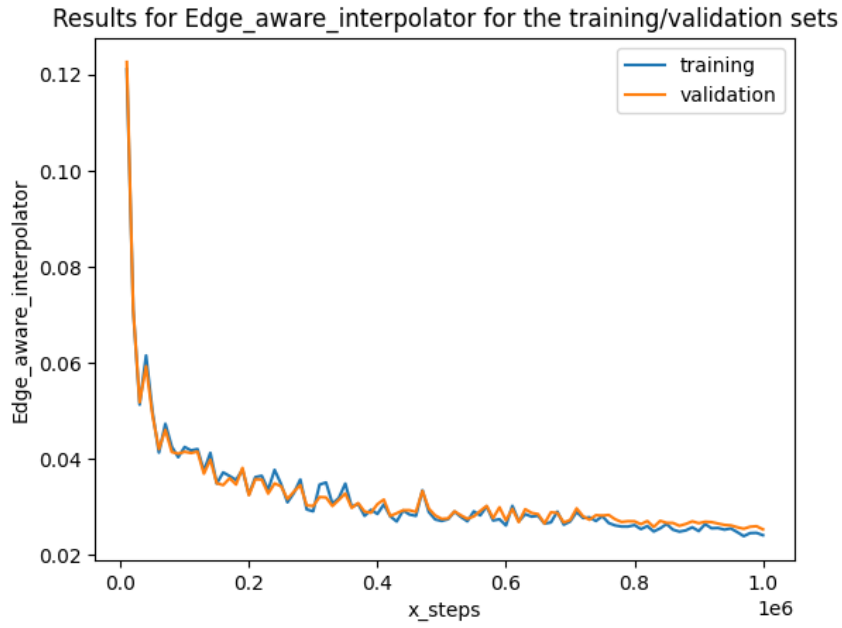
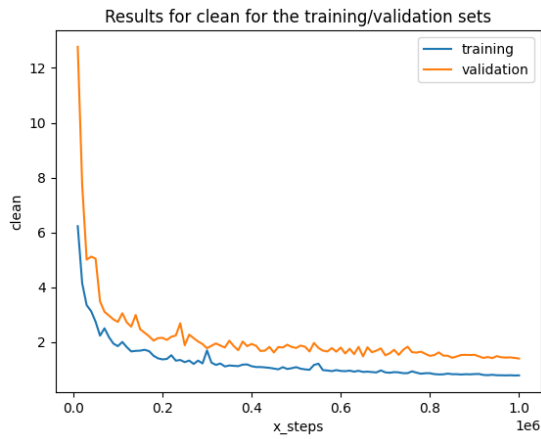
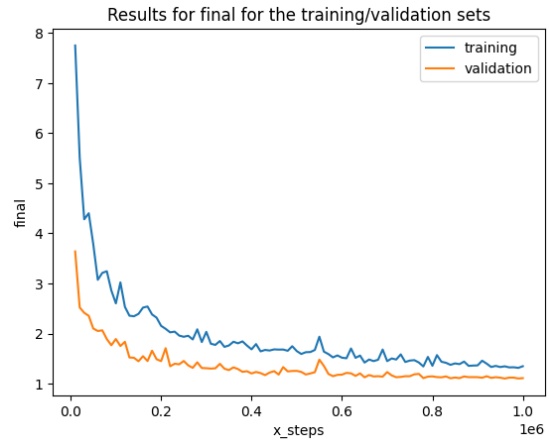


Figura 4.1: EPE en imágenes de ultrasonido.



(a) EPE en el conjunto Clean de Sintel.



(b) EPE en el conjunto Final de Sintel.

Figura 4.2: Resultados de entrenamiento para el conjunto de Sintel.

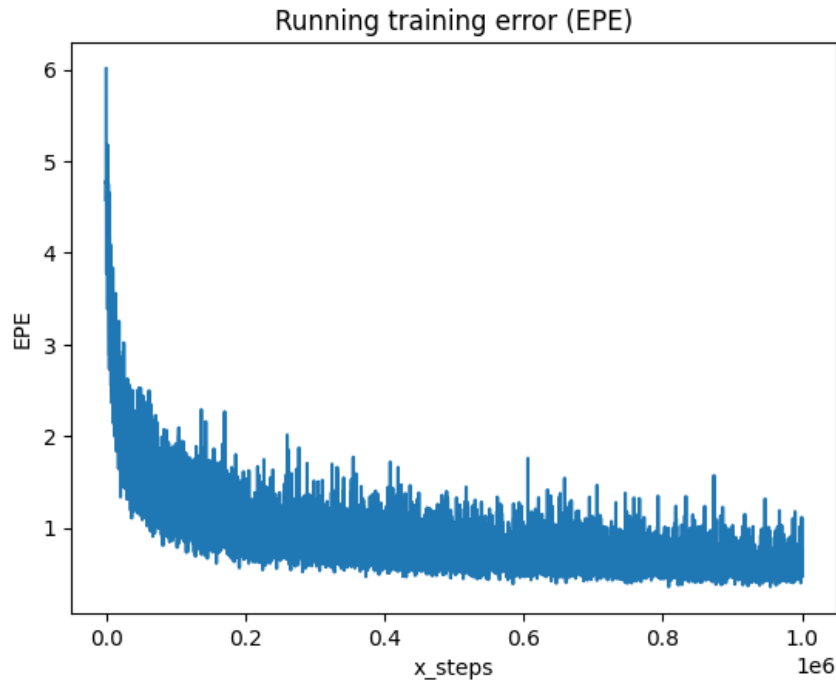


Figura 4.3: EPE en todo el conjunto de entrenamiento.

#### 4.1.2. Entrenamiento con Robust Interpolation of Correspondances

Los resultados para el entrenamiento usando la interpolación de RIC fueron muy similares a los obtenidos con EAI. Los comportamientos son exactamente los mismos para cada uno de los conjuntos de datos, tanto para los conjuntos de entrenamiento como los de validación con solo algunas ligeras diferencias en los valores de los errores obtenidos. En las Figuras 4.4, 4.5a, 4.5b se muestran los resultados para los conjuntos de ultrasonidos, clean y final respectivamente, mientras que en la Figura 4.6 se encuentra la gráfica del error obtenido para todo el conjunto de entrenamiento (ultrasonidos + Sintel), en este caso el mejor modelo se obtuvo en la época 790,000. Los mejores modelos fueron tomados considerando el mejor desempeño (error más pequeño) en el conjunto de validación.

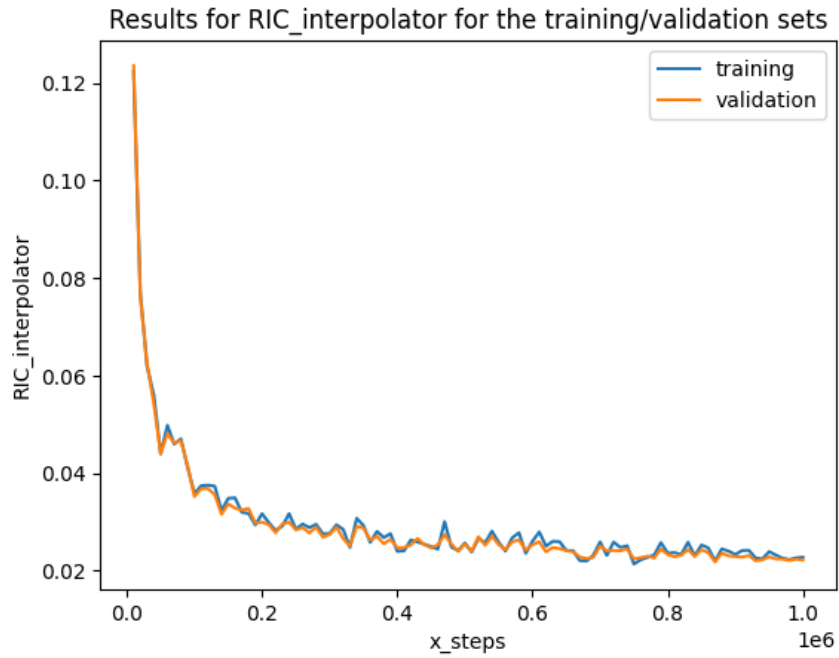
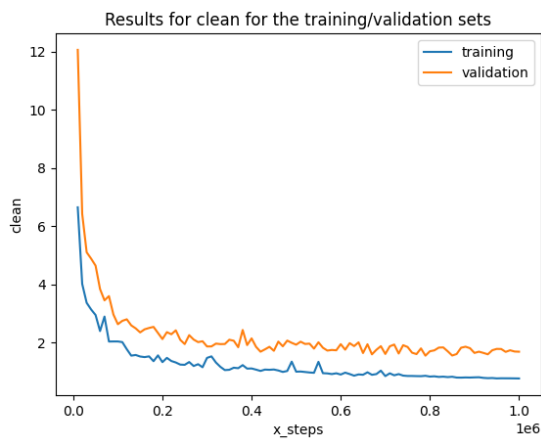
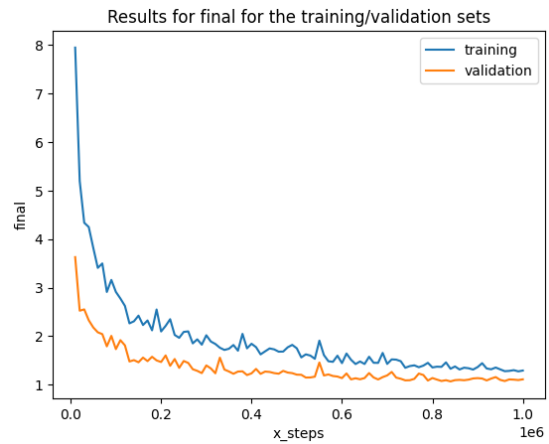


Figura 4.4: EPE en imágenes de ultrasonido.



(a) EPE en el conjunto Clean de Sintel.



(b) EPE en el conjunto Final de Sintel.

Figura 4.5: Resultados de entrenamiento para el conjunto de Sintel.

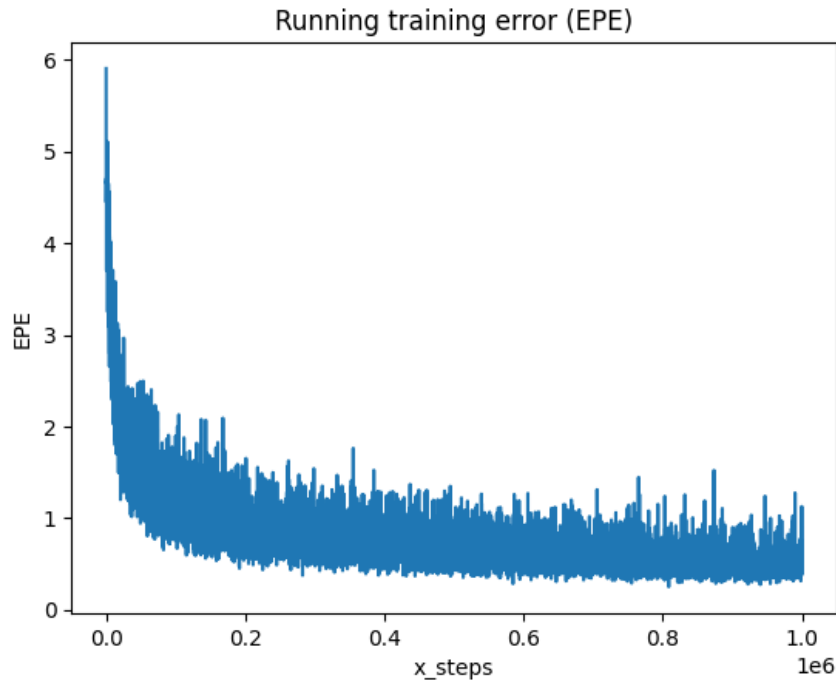


Figura 4.6: EPE en todo el conjunto de entrenamiento.

## 4.2. Resultados en MPI Sintel

Como una primera noción y visualización en la precisión en la estimación de flujo óptico de los modelos entrenados se hace una evaluación en el conjunto de datos de Sintel. En las Figuras 4.7 y 4.8 se muestran algunos resultados de la estimación de flujo óptico codificado a color y con sus respectivo EPE indicados para los modelos usando EAI y RIC, tanto para el conjunto Clean como para Final respectivamente. Considerando el conjunto Clean, de las tres diferentes imágenes que se muestran, sólo una usando EAI tuvo un menor error en la estimación (Figura 4.7a), mientras que en las otras dos con RIC tuvieron ligeramente un error más bajo (Figuras 4.7e y 4.7h). Por otro lado, las 3 imágenes del conjunto Final tuvieron un menor error usando EAI, sin embargo visualmente hay más semejanzas de los resultados obtenidos con RIC con el *ground truth*, que este mismo con los resultados de EAI.





Figura 4.7: Estimación de flujo óptico y EPE en el conjunto Clean usando EAI: (a), (d) y (g); estimación usando RIC: (b), (e) y (h); *ground truth*: (c), (f) e (i).



Figura 4.8: Estimación de flujo óptico y EPE en el conjunto Final usando EAI: (a), (d) y (g); estimación usando RIC: (b), (e) y (h); *ground truth*: (c), (f) e (i).

En la tabla 4.1 se muestra el EPE general obtenido de evaluar la estimación de flujo óptico en los conjuntos Clean y Final completos, a su vez también se hace la comparación de los mismos errores obtenidos en el artículo original de GMA [12]. Si bien los resultados obtenidos con ambos métodos de interpolación son muy parecidos a los del artículo original, siendo menor el error obtenido para el modelo con EAI para los conjuntos de entrenamiento y para el conjunto Clean de validación, mientras que hubo un menor error en el conjunto Final de validación con el modelo de RIC. La razón del porqué no son el mismo valor en los conjuntos de entrenamiento que en el artículo original se puede deber al hecho que en los modelos entrenados para este trabajo se usaron también imágenes de ultrasonido, lo cual puede estar metiendo ruido en la estimación de las imágenes de Sintel, eso y entre otras razones como la selección de hiperparámetros, modelo de la GPU y/o tiempo de entrenamiento.

Modelo	Clean (ent)	Final (ent)	Clean (val)	Final (val)
GT con EAI	0.78	1.34	<b>1.39</b>	1.10
GT con RIC	0.85	1.45	1.54	<b>1.08</b>
Art. original [12]	<b>0.62</b>	<b>1.06</b>	-	-

Tabla 4.1: EPE obtenido en Sintel.

Adicionalmente, de acuerdo con el Ranking de MPI Sintel la arquitectura GMA se encuentra en el noveno puesto dentro de los mejores modelos en la estimación de flujo óptico según [21, 22], al día de hoy.

### 4.3. Resultados en base sintética de ultrasonido

Ahora realizando una evaluación sobre el conjunto de datos de simulaciones sintéticas de ultrasonido, y nuevamente medir y visualizar el desempeño de los modelos entrenados, pero esta vez en la tarea de estimar flujo óptico en este tipo de imágenes, se han obtenido los siguientes resultados mostrados en las Figuras 4.9 y 4.10, las cuales muestran la estimación de flujo óptico codificado a color junto al EPE obtenido para los modelos entrenados con el uso de la interpolación de EAI y RIC, respectivamente.

Considerando los resultados usando EAI, se puede notar que la estimación de flujo óptico es muy precisa tanto visual como cuantitativamente, salvo pequeñas regiones que no coinciden

en color en las Figuras 4.9a, 4.9c y 4.9e, sin embargo, es importante señalar que el EPE en estas imágenes fue medido en toda la imagen y no en la región de evaluación obtenida en el capítulo anterior (Figura 3.8b), la cual sí se usa más adelante. A pesar, de las discrepancias existentes en algunas regiones para estas imágenes, la estimación de flujo óptico por el modelo entrenado con EAI muestra un resultado más homogéneo, recordando que también puede llegar a estar ligeramente equivocado el *ground truth* generado mediante interpolación y que probablemente esos errores se pudieron llegar a corregir mediante el entrenamiento.

Por otro lado, los resultados del modelo entrenado usando el método de RIC, muestran colores más sólidos con cambios de movimiento bruscos de una región a otra. Nuevamente la estimación de flujo óptico se muestra visualmente muy similar al *ground truth* incluso el EPE obtenido es más bajo que los obtenidos con EAI, esto no quiere decir aún que los resultados con RIC sean mejores que los obtenidos con EAI dado que se están usando diferentes *ground truths*, para determinar cuál de estos es mejor modelo es necesario evaluar los modelos en otro tipo de imágenes de prueba, lo cual se discutirá más adelante. Una observación importante sobre los resultados con RIC se muestra en la Figura 4.10e, la cual muestra colores más intensos en la estimación que el *ground truth*, lo cual indica que la estimación es correcta en dirección, pero no lo es tanto en la magnitud del movimiento.

Cabe mencionar además que los resultados mostrados en estas figuras, tanto para EAI y RIC, son exactamente las mismas imágenes del conjunto de ultrasonidos.

Finalmente, en la Tabla 4.2 se muestra el EPE obtenido para ambas implementaciones (EAI y RIC), tanto para el conjunto de entrenamiento y el conjunto de validación. Para la obtención de este EPE si se ha considerado la región de evaluación (Figura 3.8b). Mostrando un menor error en el conjunto de entrenamiento, el modelo entrenado con EAI (5420 ejemplos), mientras que hay un menor error en el conjunto de validación con el modelo entrenado con RIC (640 ejemplos). Nuevamente, estos resultados aún no son determinantes para definir si un modelo es mejor o no que el otro, para ello se llevaron a cabo otras evaluaciones.

El tiempo en realizar la estimación en los 6060 ejemplos es muy similar con ambos modelos, lo cual se puede expresar de otra manera, como 2 estimaciones por segundo, que a su vez es bastante bueno para una tarea de tiempo real.

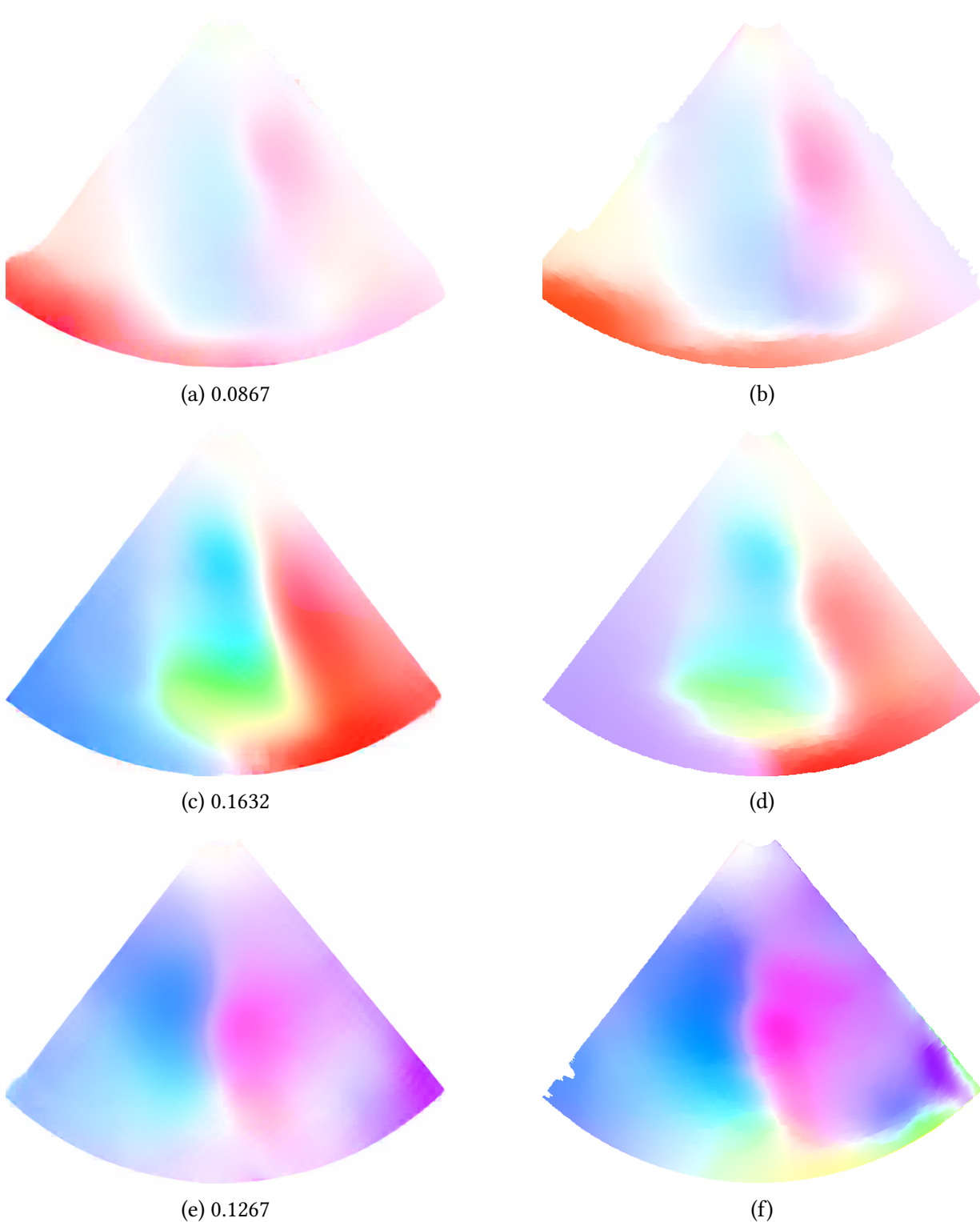


Figura 4.9: Estimación de flujo óptico y EPE en el conjunto de ultrasonidos usando EAI: (a) Imagen 46 de Esaote, caso normal y vista de 4 cámaras; (c) Imagen 40 de Esaote, caso con isquemia y vista a 2 cámaras; (e) Imagen 45 de Philips Medical Systems, caso con isquemia y vista de 3 cámaras; *ground truth*: (b), (d) y (f) respectivamente.

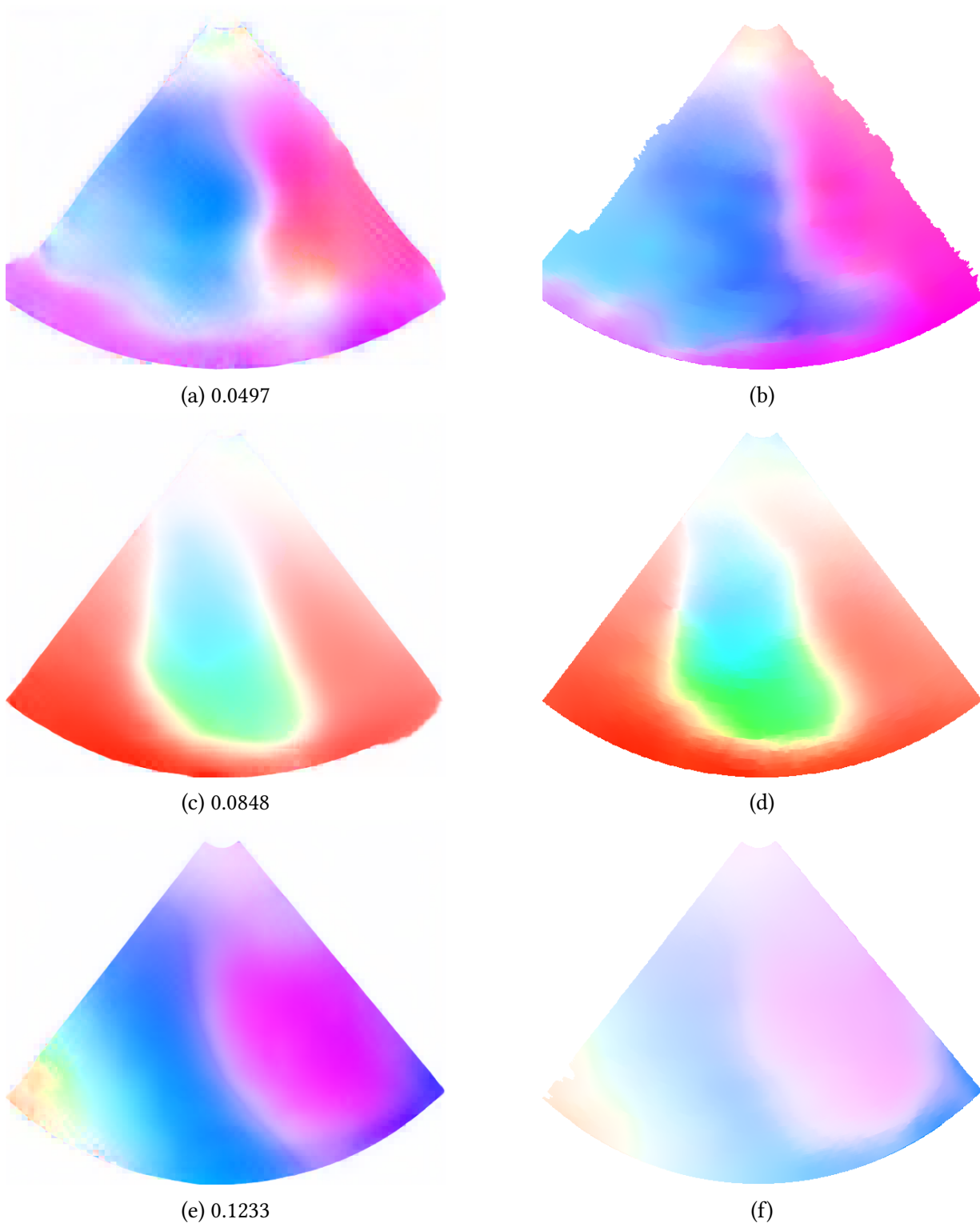


Figura 4.10: Estimación de flujo óptico y EPE en el conjunto de ultrasonidos usando RIC: (a) Imagen 46 de Esaote, caso normal y vista de 4 cámaras; (c) Imagen 40 de Esaote, caso con isquemia y vista a 2 cámaras; (e) Imagen 45 de Philips Medical Systems, caso con isquemia y vista de 3 cámaras; *ground truth*: (b), (d) y (f) respectivamente.

Modelo	Entrenamiento	Validación	Tiempo de estimación (s)
GT con EAI	<b>0.0241</b>	0.0253	2986.58
GT con RIC	0.0257	<b>0.0243</b>	2987.25

Tabla 4.2: EPE de los modelos entrenados con EAI y RIC en entrenamiento y validación de imágenes de ultrasonido.

#### 4.4. Comparación con métodos tradicionales

Se realizaron las implementaciones en Python de los métodos Lucas-Kanade y RLOF, con el fin de poder medir su desempeño en la estimación de flujo óptico en las imágenes de ultrasonido y compararlos con los modelos que se han entrenado. En las Figuras 4.11 y 4.12 se muestra la estimación obtenida codificada a color con cada método, respectivamente. Donde se observa que las estimaciones de ambos métodos no se parecen en lo absoluto con el *ground truth* tanto con EAI como con RIC, incluso las estimaciones muestran movimiento en la región fuera del abanico del ultrasonido, el cual siempre permanece constante y sin movimiento. Esto demuestra lo difícil que es para los métodos tradicionales realizar estimación de flujo óptico en este tipo de imágenes. Por otro lado, en la Tabla 4.3 se presentan los EPE obtenidos de evaluar todo el conjunto de imágenes de ultrasonidos (6060 ejemplos) para ambos métodos de estimación de flujo óptico, así como el tiempo de evaluación en los mismos, con los diferentes valores de *ground truth* obtenidos con EAI y RIC. Donde se ha obtenido un mejor desempeño en la precisión con el método de RLOF que en el obtenido con LK, esto no es de extrañarse puesto que RLOF es un método más robusto.

Haciendo la comparación con la Tabla 4.2, el EPE de los modelos entrenados con la red GMA es aproximadamente 1/64 del obtenido con estos métodos tradicionales, lo cual también indica que con redes neuronales se pueden atacar fácilmente problemas que con métodos tradicionales resultan complejos y complicados de resolver. A pesar de que el tiempo de estimación de flujo óptico resulta mucho más rápido con los métodos tradicionales antes mencionados, no resultan eficientes puesto que se busca maximizar la precisión en la estimación y minimizar el tiempo de estimación a la vez.

Método	EPE (EAI)	EPE (RIC)	Tiempo de estimación (s)
LK	1.5456	1.5456	436.64
RLOF	1.2801	1.2803	1085.70

Tabla 4.3: EPE en todo el conjunto de ultrasonidos para los métodos tradicionales con el *ground truth* generado con EAI y RIC.

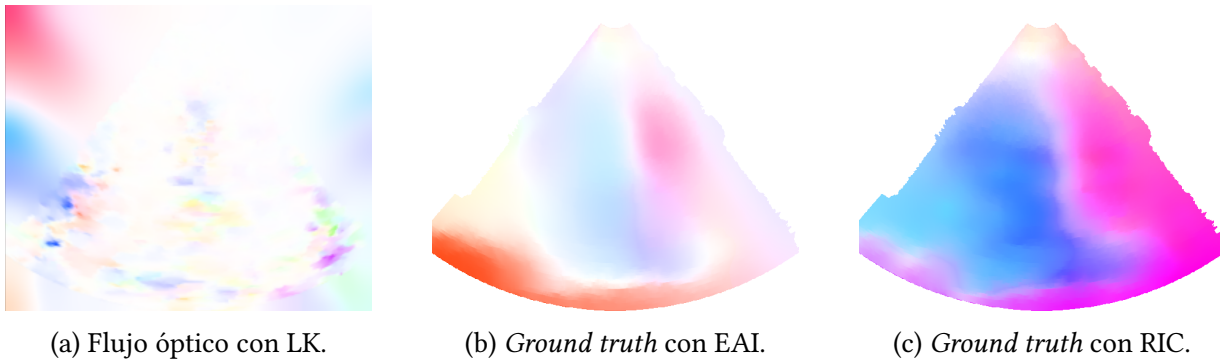


Figura 4.11: Estimación de flujo óptico codificado a color para la imagen 46 de Esaote, caso normal y vista de 4 cámaras, usando el método de Lucas-Kanade.

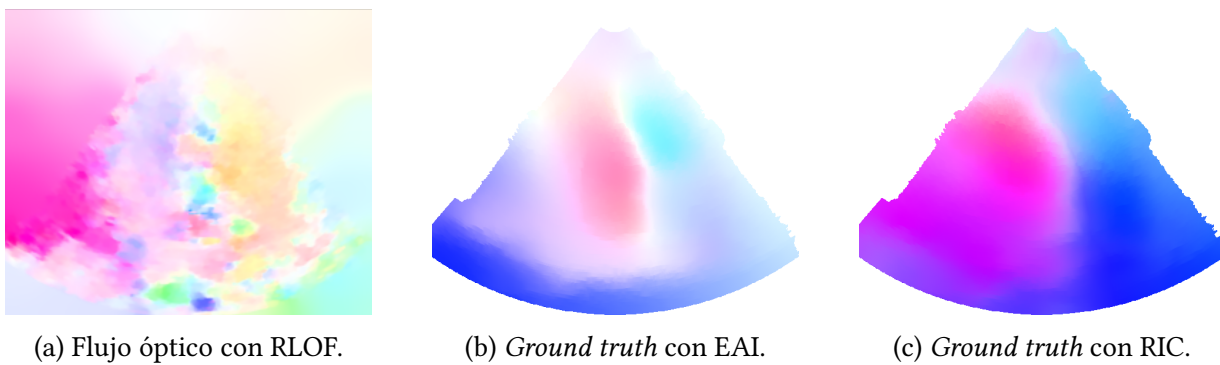


Figura 4.12: Estimación de flujo óptico codificado a color para la imagen 9 de Esaote, caso normal y vista de 4 cámaras, usando el método de RLOF.

## 4.5. Evaluación en otros datasets

Con el fin de evaluar de manera más precisa y confiable los modelos entrenados se ha utilizado un conjunto de datos tomado de [23], el cual cuenta con diversas secuencias de imágenes con su respectivo *ground truth*. Las diferentes secuencias se han pasado como entrada en los modelos entrenados con GMA y en los métodos tradicionales para medir la precisión en la estimación en todas las implementaciones con otro tipo de imágenes (no imágenes de ultrasonido, ni Sintel). En la Figura 4.13 se muestran los resultados cualitativos que resultan de las estimaciones del flujo óptico con cada método y la comparación con el *ground truth*. En esta figura se aprecia que los resultados de los métodos tradicionales nuevamente no parecen ser tan precisos como los modelos entrenados con la arquitectura GMA. De los métodos tradicionales el método de RLOF resulta más efectivo que LK, por otra parte, de los modelos basados en redes neuronales ambos se muestran muy parecidos visualmente y muy cercanos y comparables con el *ground truth*. En la tabla 4.4, se muestran los mismos resultados pero ahora de forma cuantitativa, donde se ha medido el EPE para cada una de las diferentes secuencias del conjunto de prueba. Confirmando el análisis cualitativo, se muestra que efectivamente el peor método en la estimación de flujo óptico fue LK y de forma general los modelos entrenados muestran una mejor precisión que los métodos tradicionales.

Dentro de los modelos entrenados con la arquitectura GMA, es difícil decir qué modelo es mejor que el otro puesto que los EPE obtenidos en el conjunto de prueba son muy similares de acuerdo con la tabla, siendo el modelo entrenado con la interpolación de EAI mejor en 2 de las secuencias y RIC mejor en las otras 3.

Secuencia	LK	RLOF	GMA (EAI)	GMA (RIC)
Groove2	0.29	0.23	0.24	<b>0.22</b>
Groove3	1.04	0.85	<b>0.68</b>	0.71
Urban2	1.35	0.55	<b>0.28</b>	0.29
Urban3	1.15	0.81	0.38	<b>0.37</b>
Venus	0.46	0.41	0.30	<b>0.30</b>

Tabla 4.4: Resultados del EPE en el conjunto de prueba para cada método y modelo entrenado.



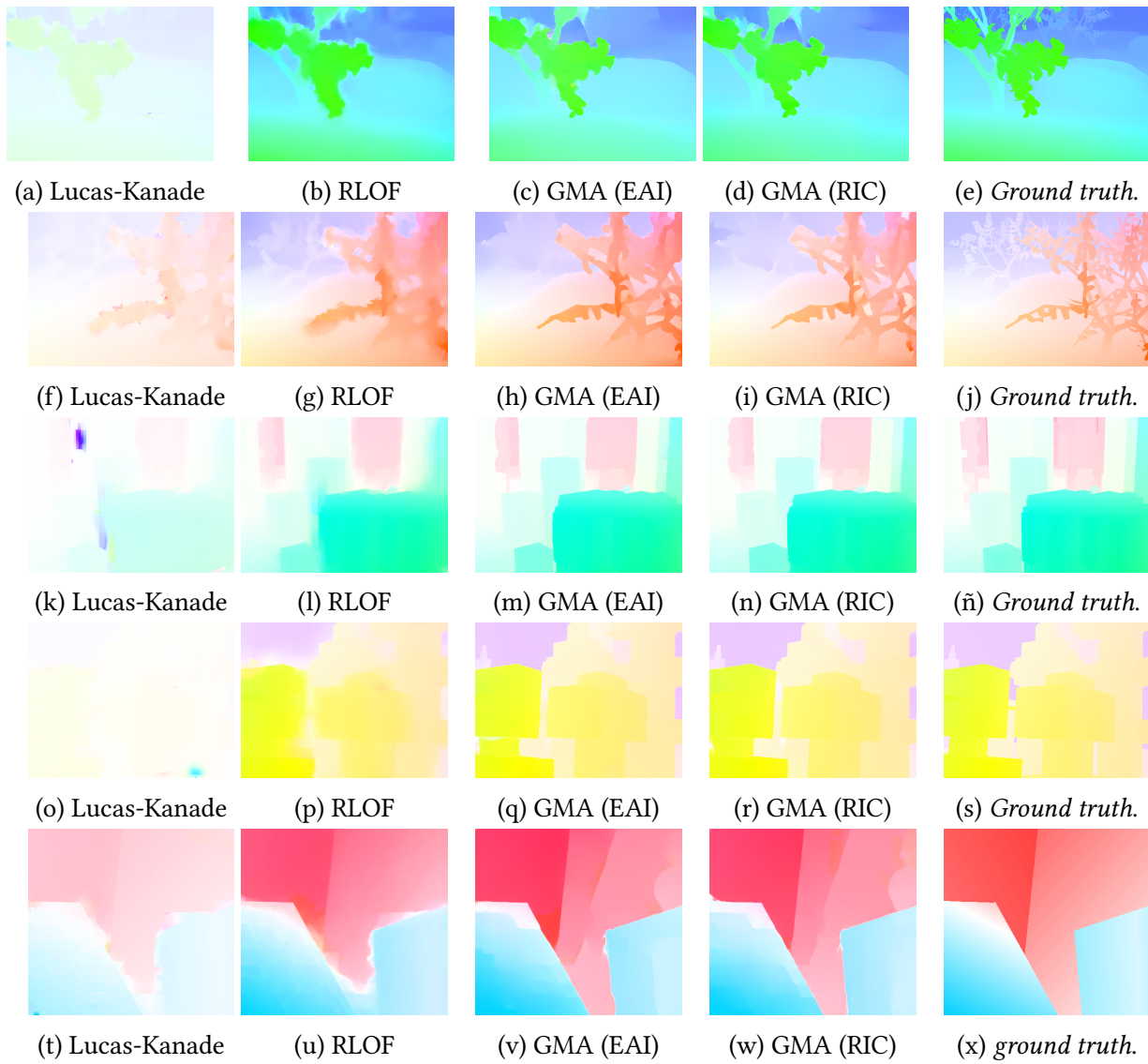


Figura 4.13: Estimación de flujo óptico de cada método y modelo entrenado en el conjunto de prueba. Groove2: (a), (b), (c), (d) y (e); Groove3: (f), (g), (h), (i) y (j); Urban2: (k), (l), (m), (n) y (ñ); Urban3: (o), (p), (q), (r) y (s); Venus: (t), (u), (v), (w) y (x).

## 4.6. Estimación en imágenes médicas reales

Finalmente, ya que se ha comprobado la certeza en la precisión de la estimación de flujo óptico de los modelos entrenados, se pueden realizar estimaciones ahora sí en imágenes médicas reales, donde esta vez únicamente se puede hacer un análisis visual y cualitativo puesto que no se cuenta con el *ground truth* para estas imágenes. Las imágenes fueron tomadas tanto del proyecto CAMUS [24], el cual cuenta con una amplia cantidad de imágenes y secuencias de ultrasonido a 4 y 2 cámaras, dedicadas para segmentación, también se tomaron otras de Echo-Net, base de datos con vídeos de pacientes humanos en ultrasonido cardiaco, proporcionados por la Universidad de Standford en 2020 [25].

Ahora, las estimaciones de movimiento en un par de imágenes del conjunto de CAMUS se presentan en la Figura 4.14, en la cual se presentan dos cuadros consecutivos de una secuencia de ultrasonido y las respectivas estimaciones de flujo óptico con los modelos entrenados. Algo importante de hacer notar en esta figura, es que a pesar de que ahora se están introduciendo imágenes que jamás se vieron durante el entrenamiento ambos modelos estiman perfectamente el movimiento en la región externa al abanico del ultrasonido, la cual corresponde a una zona sin movimiento, por otro lado también se observa que el modelo entrenado con la interpolación de EAI marca notablemente la zona del ventrículo izquierdo mientras que en el modelo con RIC sólo se muestran dos regiones de forma casi homogénea.

Usando también un par de cuadros consecutivos con vista de 4 cámaras del conjunto de Echo-Net, se realizó la estimación de movimiento con ambos modelos, como lo muestra la Figura 4.15, en esta puede observarse nuevamente que es el modelo entrenado con EAI en el que se aprecian los diferentes movimientos en las 4 cavidades del corazón, mientras que en el modelo entrenado con RIC se tiene otra vez una región casi homogénea.

Se han usado también estos modelos para estimación de flujo óptico en otro tipo de imagen pero de mismo propósito, que es evaluar la función cardiaca, dichas imágenes corresponden a las de CT. A pesar de que los modelos entrenados nunca tuvieron una entrada con la misma naturaleza de las imágenes de CT durante su entrenamiento, algo importante de notar se muestra en la Figura 4.16, en la que ahora el modelo entrenado con EAI indica un color morado a toda la región del corazón, lo cual es incorrecto puesto que en los cuadros consecutivos presentados en

esta figura es fácil notar que se habla de una de las primeras etapas de la fase sistólica (contracción del corazón), por otro lado ahora pareciera que la estimación de flujo óptico se muestra mejor en el modelo entrenado con RIC, puesto que esta sí muestra las diferentes direcciones de movimiento correctas para el ventrículo izquierdo.

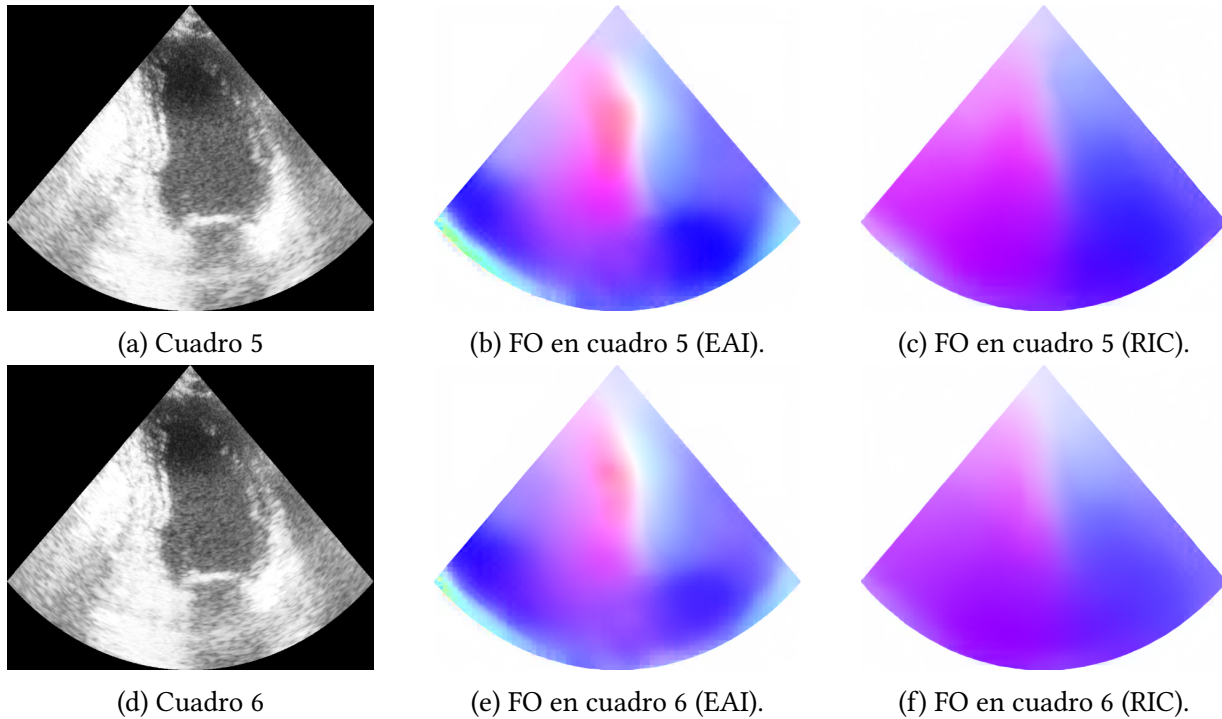


Figura 4.14: Estimación de flujo óptico en imágenes de CAMUS.

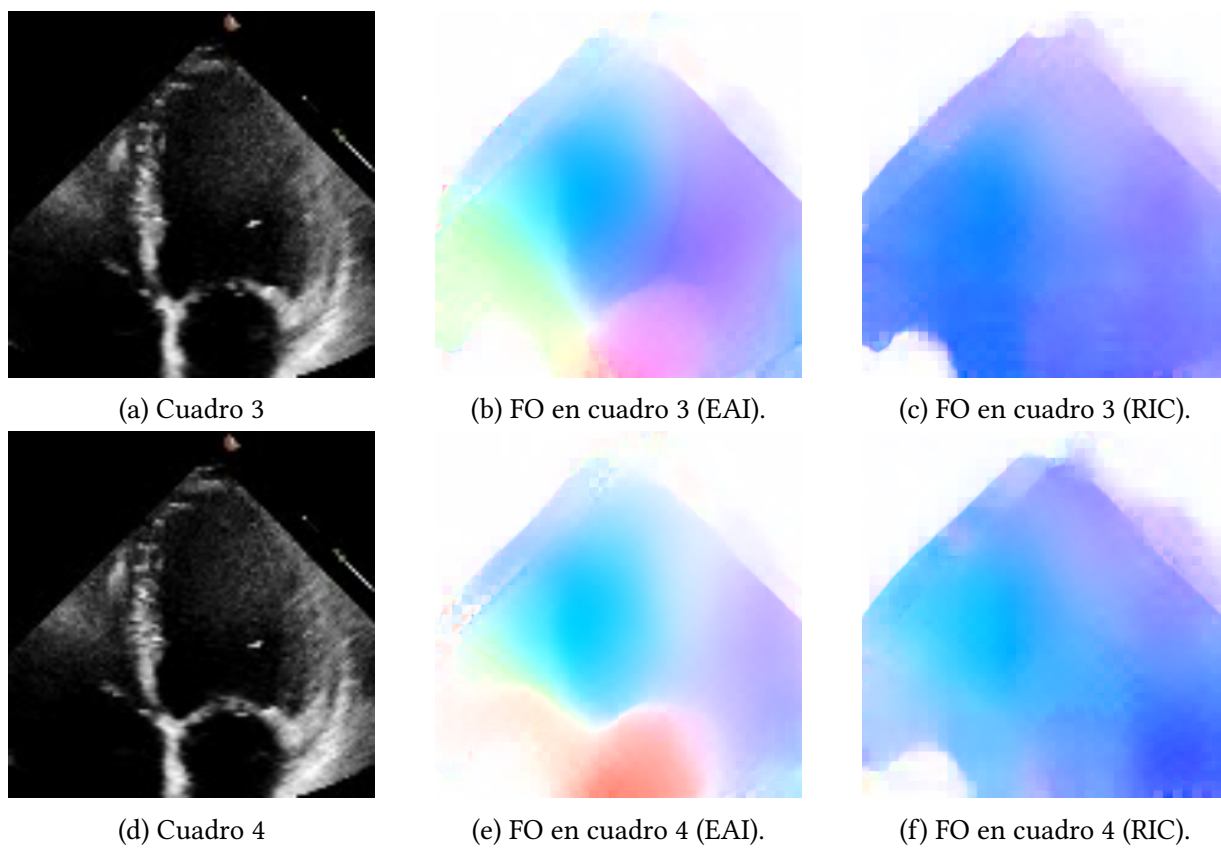


Figura 4.15: Estimación de flujo óptico en imágenes de Echo-Net.

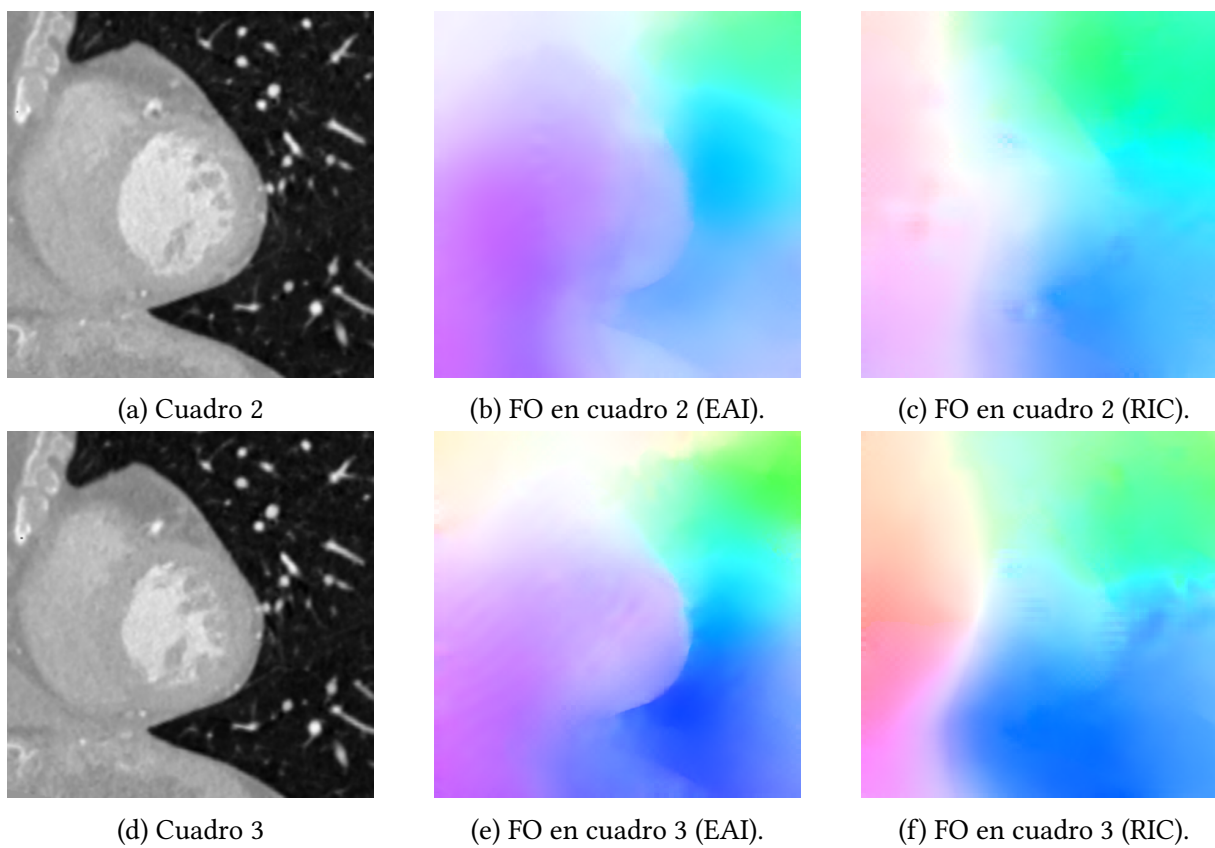


Figura 4.16: Estimación de flujo óptico en imágenes de tomografía computarizada (CT).

# Capítulo 5

## Conclusiones

Durante este trabajo, se ha logrado una nueva forma de resolver el problema de la falta de *ground truth* en imágenes médicas (especialmente ruidosas, como los ultrasonidos), usando distintos métodos de interpolación de flujo óptico, y por lo tanto, usarlo para crear un modelo basado en redes neuronales. A su vez, también se buscaron diferentes conjuntos de datos de otra naturaleza distinta a las imágenes médicas para el entrenamiento y la evaluación de los modelos, con el fin de que este modelo no sólo sea de utilidad en una tarea específica como estimar movimiento en ultrasonido, si no también estimarlo de forma general en cualquier tipo de imagen, lo cual se ha demostrado con los resultados mostrados en la Figura 4.13. También, otro de los puntos importantes realizados en este trabajo fue el de no sólo realizar la implementación y entrenamiento de la arquitectura GMA, si no de igual manera implementar otros métodos (tradicionales) para comparar cómo se comporta el modelo propuesto con estos mismos.

Después de realizar un amplio análisis de los resultados obtenidos en los diferentes conjuntos de datos utilizados, se ha comprobado el buen desempeño que tienen las redes neuronales en tareas de visión computacional, en el caso de este trabajo, con la estimación de flujo óptico, logrando un mejor rendimiento tanto en precisión como en tiempo de estimación que los métodos tradicionales, tal y como se mostró en las Tablas 4.2 y 4.3.

Por otro lado, de los modelos entrenados en este trabajo se puede concluir que ambos tienen muy buen rendimiento en la estimación de flujo óptico de forma general y de forma más específica, se mostró que el modelo entrenado con EAI produce mejores resultados en las imágenes de ultrasonido, puesto que las estimaciones de movimiento se muestran más suaves entre

regiones que con RIC. Sin embargo, la precisión en otro tipo de imágenes (incluidas las imágenes de tomografía computarizada) se muestra ligeramente mejor el rendimiento con el modelo entrenado con RIC, de acuerdo con las Tablas 4.1 y 4.4 y la estimación de movimiento en las imágenes de CT.

Finalmente, la aportación de este trabajo ha sido la de tener un modelo que sea capaz de estimar movimiento en imágenes de ultrasonido y otro tipo de imágenes, de forma confiable y en el menor tiempo posible, así como también la creación de una nueva base de datos de imágenes de ultrasonido con su respectivo *ground truth* de movimiento a lo largo de una secuencia, la cual puede ser de mucha utilidad para realizar otros entrenamientos relacionados con estimación de flujo óptico a quién lo llegase a necesitar.

Sin embargo, los modelos aún no son del todo precisos como uno esperaría, ó bien aún, para que sea de utilidad para un médico en la vida real, puesto que aún hay ciertos puntos que podrían considerarse para mejorar la calidad de este trabajo y resulte de verdadera utilidad en en este ámbito en un futuro.

## 5.1. Trabajo a futuro

Los puntos a considerar para la continuación y mejora de este trabajo son los siguientes:

- Realizar una interpolación del flujo óptico disperso y específica por imagen, de forma que el *ground truth* que se genere con este sea todavía más confiable para realizar un entrenamiento.
- Encontrar una mejor forma de realizar la interpolación del movimiento fuera de la región de la pared ventricular.
- Generar datos para entrenamiento también con imágenes de CT. Esto implica generar *ground truth* para este tipo de imágenes también, ó buscar bases de datos existentes sobre esta posibilidad.
- Utilizar una arquitectura más precisa y con mejor rendimiento que GMA, si bien ya se mencionado que en el Ranking actual de MPI Sintel [21, 22] hay arquitecturas mejores

que la de GMA y esto sigue cambiando constantemente con el paso del tiempo, puesto que nos encontramos en un momento de grandes avances en ciencia y tecnología, en este que se conoce como la era del deep learning.



# Bibliografía

- [1] M. Goitein, “Organ and tumor motion: an overview,” *Semin Radiat Oncol.*, pp. 2–9, 2004.
- [2] WHO, “The top 10 causes of death.” <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>, Junio 2022.
- [3] G. Buckberg, J. Hoffman, A. Mahajan, S. Saleh, and C. Coghlan, “Mechanics revisited: The relationship of cardiac architecture to ventricular function,” *Circulation*, 2008.
- [4] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981.
- [5] G. Farneäck, “Two-frame motion estimation based on polynomial expansion,” in *Image Analysis* (J. Bigun and T. Gustavsson, eds.), (Berlin, Heidelberg), pp. 363–370, Springer Berlin Heidelberg, 2003.
- [6] T. Senst, V. Eiselein, and T. Sikora, “Robust local optical flow for feature tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1377–1387, 2012.
- [7] B. Horn and B. Schunck, “Determining optical flow,” *ARTIFICIAL INTELLIGENCE*, pp. 1,2,3,4, 1981.
- [8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, and V. Golkov, “Flownet: Learning optical flow with convolutional networks,” *IEEE International Conference on Computer Vision*, 2015.

- [9] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of optical flow estimation with deep networks,” *Conference on Computer Vision and Pattern Recognition*, 2018.
- [10] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume,” pp. 8934–8943, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2018.
- [11] Z. Teed and J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow,” in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 402–419, Springer International Publishing, 2020.
- [12] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. I. Hartley, “Learning to estimate hidden motions with global motion aggregation,” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9752–9761, 2021.
- [13] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow,” in *Computer Vision and Pattern Recognition*, 2015.
- [14] Y. Hu, Y. Li, and R. Song, “Robust interpolation of correspondences for large displacement optical flow,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4791–4799, 2017.
- [15] E. Moya-Albor, B. Escalante-Ramírez, and E. Vallejo, “Optical flow estimation in cardiac ct images using the steered hermite transform,” *Signal Processing: Image Communication*, vol. 28, no. 3, pp. 267–291, 2013.
- [16] M. Alessandrini, B. Chakraborty, B. Heyde, O. Bernard, M. de Craene, M. Sermesant, and J. d’Hooge, “Realistic Vendor-Specific Synthetic Ultrasound Data for Quality Assurance of 2-D Speckle Tracking Echocardiography: Simulation Pipeline and Open Access Database,” *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 65, pp. 411–422, Mar. 2018.

- [17] Martino Alessandrini and Bidisha Chakraborty and Oliver Bernard and Mathieu De Craene and MAXime Sermesant and Jan D’hooge, “Open access data base.” <https://gbiomed.kuleuven.be/english/research/50000635/50508167/open-data>, Septiembre 2021.
- [18] Shihao Jiang and Dylan Campbell and Yao Lu and Hongdong Li and Richard Hartley, “GMA implementation in Pytorch.” <https://github.com/zacjiang/GMA>, 2021. Online; accessed 7 December 2021.
- [19] 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)* (A. Fitzgibbon et al. (Eds.), ed.), Part IV, LNCS 7577, pp. 611–625, Springer-Verlag, Oct. 2012.
- [20] NVIDIA, “Nvidia titan rtx.” <https://www.nvidia.com/es-la/deep-learning-ai/products/titan-rtx/>, Julio 2022.
- [21] Butler, D. J. and Wulff, J. and Stanley, G. B. and Black, M. J., “Results and rankings.” [http://sintel.is.tue.mpg.de/quant?metric\\_id=0&selected\\_pass=1](http://sintel.is.tue.mpg.de/quant?metric_id=0&selected_pass=1), Julio 2022.
- [22] Butler, D. J. and Wulff, J. and Stanley, G. B. and Black, M. J., “Results and rankings.” [http://sintel.is.tue.mpg.de/quant?metric\\_id=0&selected\\_pass=0](http://sintel.is.tue.mpg.de/quant?metric_id=0&selected_pass=0), Julio 2022.
- [23] Brad Hiebert-Treuer and Alan Lim, “Optical flow datasets.” <https://vision.middlebury.edu/flow/data/>, Julio 2022.
- [24] S. Leclerc, E. Smistad, J. Pedrosa, A. Ostvik, and et al., “Deep learning for segmentation using an open large-scale dataset in 2d echocardiography,” *IEEE Transactions on Medical Imaging*, 2019.
- [25] D. Ouyang, B. He, A. Ghorbani, N. Yuan, J. Ebinger, C. P. Langlotz, P. A. Heidenreich, R. A. Harrington, D. H. Liang, E. A. Ashley, , and J. Y. Zou., “Video-based ai for beat-to-beat assessment of cardiac function,” *Nature*, 2020.