

# **SISTEMA PARA LA OBTENCIÓN DE UN ELASTOGRAMA DE TEJIDOS BIOLÓGICOS**



**Cristian Felipe Gutiérrez Pazos  
Byron Yesid Jaramillo Hurtado**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Electrónica, Instrumentación y Control  
Ingeniería en Automática Industrial  
Popayán, Abril de 2015**

**SISTEMA PARA LA OBTENCIÓN DE UN  
ELASTOGRAMA DE TEJIDOS BIOLÓGICOS**

**Monografía presentada como requisito parcial para optar por el  
título de Ingeniero en Automática Industrial**

**Cristian Felipe Gutiérrez Pazos  
Byron Yesid Jaramillo Hurtado**

**Director: Mg. Elena Muñoz España**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Electrónica, Instrumentación y Control  
Ingeniería en Automática Industrial  
Popayán, Abril de 2015**

Nota de aceptación: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Director: \_\_\_\_\_

Elena Muñoz España

\_\_\_\_\_

Firma del Jurado

\_\_\_\_\_

Firma del Jurado

Popayán, Abril de 2015

## **AGRADECIMIENTOS**

En esta sección, queremos dar agradecimiento primeramente a Dios, por ser el guía y la fuerza necesaria para impulsar este trabajo. Por permitirnos abrir los ojos día a día y servirnos de consuelo en situaciones adversas a este trabajo. Además de permitirnos conocer personas que nos apoyaron de principio a fin en la consecución de este trabajo.

Un agradecimiento especial a nuestros padres, que nos dieron ese apoyo, comprensión y paciencia. Quiero agradecer a manera personal a mi madre Ana Milena Pazos, una mujer luchadora que me impulso a trabajar más duro en este proyecto cuando supe de la enfermedad que le aquejaba, además de enseñarme a diario como ser mejor persona, profesional e hijo, sin su apoyo esto no hubiera sido posible. Agradecer a mi compañero de tesis Byron Yesid Jaramillo por su paciencia y dedicación para sacar este proyecto adelante.

Agradecerles a nuestros familiares, tíos, primas y demás que manifestaron su apoyo y colaboración, además de muestras de cariño y afecto necesarias en los momentos difíciles que pudimos pasar en el desarrollo de este proyecto.

A mis compañeros, amigos y colegas que estuvieron compartiendo con nosotros en los buenos y malos momentos, nos brindaron ayuda y nos llenaron de buenos recuerdos, fuera de los lazos de amistad que se reforzaron aún más.

Un agradecimiento a nuestra directora, la ingeniera Elena Muñoz España, por permitirnos trabajar bajo su dirección, por la paciencia, comprensión y guía en el desarrollo de este trabajo.

Agradecimiento especial al Doctor Carlos Gaviria López, por permitirnos trabajar en su proyecto, servirnos de guía y aportarnos las herramientas necesarias en la ejecución de este.

Agradezco a la Universidad del Cauca, a la Facultad de Ingeniería Electrónica y Telecomunicaciones y especialmente al programa de Ingeniería en Automática Industrial que por medio de sus docentes y administrativos estuvieron siempre atentos y prestos a nuestro servicio.

Y para terminar nos despedimos con una frase inmortalizada por un gran artista musical el cual expresa nuestro agradecimiento: “No solo no hubiéramos sido nada sin ustedes, sino con toda la gente que estuvo a nuestro alrededor desde el comienzo; algunos siguen hasta hoy. ¡Gracias... Totales!”.

## RESUMEN

La elastografía es una técnica desarrollada para medir las propiedades elásticas de un tejido u órgano. En la actualidad existen gran variedad de enfoques de esta técnica, siendo el más atractivo el diagnóstico de tumores. La elastografía tiene más de 20 años de desarrollo pero sigue aún con la necesidad de mejorar en varios aspectos, que van desde el desarrollo tecnológico de las máquinas de adquisición de imágenes ya sea de ultrasonido, resonancia magnética o tomografía computarizada, hasta el desarrollo los algoritmos de elastografía y de estándares y normalizaciones de diagnóstico que permitan altos niveles de precisión, resolución, robustez con el fin de hacer de la elastografía una técnica con la que el médico pueda hacer un diagnóstico de la enfermedad a tratar, de una manera confiable, precisa y repetible bajo todo tipo de entornos y situaciones.

En este estudio se hace un acercamiento a la elastografía, a cada una de sus técnicas y métodos más comunes, siendo la elastografía cuasi estática a mano alzada el enfoque de este proyecto. La herramienta que se ha implementado permitirá a los usuarios tener un acercamiento a los métodos y técnicas utilizados en elastografía, mediante la obtención y visualización de imágenes de elastografía o elastogramas, a partir de imágenes RF e imágenes en modo B de los tejidos que se tomaran de ejemplo.

La herramienta tiene la opción de obtener imágenes de elastografía con tres métodos diferentes: Búsqueda de Bloques Exhaustiva (llamado SSD en este documento), híbrido (HBD) y AM2D, siendo este último un algoritmo desarrollado por Hassan Rivas en [1]. Los resultados obtenidos despliegan información semicuantitativa tal que los usuarios puedan interpretar y llegar al acercamiento a un diagnóstico. Para ello se hace uso de herramientas de cálculo como señal a ruido o la razón entre los promedios de las secciones seleccionadas en la imagen permitiendo determinar la rigidez de los tejidos a evaluar.

La codificación de esta herramienta software se llevó a cabo en el entorno de desarrollo integrado de "Microsoft Visual Studio 2012", también se utilizan otras herramientas informáticas como "Qt-Creator" para el desarrollo de la interfaz, "OpenCV" para el procesamiento digital de las imágenes y "MATLAB" para la implementación de librerías compartidas para el cálculo mediante algoritmos basados en MATLAB.

Finalmente se obtuvo el software Elax1, cuyo nombre guarda relación con la técnica utilizada "Elastografía", que permite evaluar y obtener elastogramas mediante técnicas fuera de línea.

## TABLA DE CONTENIDO

LISTA DE FIGURAS .....	9
LISTA DE TABLAS .....	11
1 INTRODUCCIÓN .....	1
1.1 Motivación.....	1
1.2 Revisión de la elastografía en la literatura .....	2
1.3 Alcances y objetivos .....	7
1.3.1 Objetivo general .....	7
1.3.2 Objetivos Específicos .....	7
1.4 Estructura de la monografía.....	7
2 ELASTOGRAFÍA: CONCEPTOS GENERALES .....	9
2.1 ¿Señales RF o imágenes modo B? .....	21
2.2 Fundamentos de elastografía .....	22
2.3 Mecánica .....	22
2.3.1 Tensión (Stress).....	22
2.3.2 Deformación (Strain) .....	24
2.3.3 Ley de Hooke .....	26
3 ALGORITMOS PARA EL CÁLCULO DE LA DEFORMACIÓN .....	28
3.1 Algoritmos de estimación de desplazamientos .....	28
3.1.1 Búsqueda de bloques exhaustiva.....	28
3.1.2 Estimación de desplazamiento híbrido .....	29
3.1.3 AM2D .....	38
3.2 Cálculo del Gradiente .....	39
3.3 Normalización .....	39
4 IMPLEMENTACIÓN Y DESARROLLO SOFTWARE .....	42
4.1 Herramientas de desarrollo software utilizadas .....	42
4.1.1 QT [54] .....	42
4.1.2 OpenCV [55] .....	44
4.1.3 Matlab Compiler Runtime [56].....	46
4.1.4 Microsoft Visual Studio [57].....	46
4.2 Diseño e implementación.....	47
4.3 Diagrama de clases .....	49

4.4	Método SSD .....	51
4.4.1	Strain.....	51
4.4.2	Gradiente .....	53
4.4.3	NormA1 .....	54
4.4.4	BMA .....	56
4.5	Método Híbrido .....	58
4.5.1	Buscador_piramidal.....	58
4.5.2	Bsc .....	60
4.5.3	Lista.....	61
4.6	Método AM2D.....	63
4.7	Funciones asociadas a la interfaz.....	65
4.7.1	Método ELTGP.....	65
4.7.2	Método SNR_Dialog.....	70
5	RESULTADOS .....	72
5.1	Resultados Phantom.....	72
5.2	Resultados Paciente 3 y 4 .....	84
6	CONCLUSIONES Y TRABAJOS FUTUROS .....	92
6.1	Conclusiones .....	92
6.2	Trabajos futuros.....	93
7	REFERENCIAS BIBLIOGRAFICAS .....	94

## LISTA DE FIGURAS

Figura 1. Tomografía computarizada por resonancia magnética.....	10
Figura 2. Imágenes de ondas, 2A muestra la propagación de ondas de corte a través del parénquima hepático, 2B una imagen de elastografía con valores de rigidez elevados con una media de 15 kPa consistente de fibrosis. ....	10
Figura 3. Ecografía típica de un feto. ....	11
Figura 4. Representación de los componentes de una máquina de ultrasonido....	12
Figura 5. Máquina de ultrasonido convencional.....	13
Figura 6. Máquina de ultrasonido portable.....	13
Figura 7. Tipos de sonógrafos. ....	14
Figura 8. Enfoques generales de las elastografía.....	16
Figura 9. Elastografía casi estática típica. ....	17
Figura 10. Criterios de evaluación en tiempo real para lesiones mamarias de acuerdo con las puntuaciones Itoh et al.....	19
Figura 11. Elastografía mamaria de paciente 1. ....	20
Figura 12. Elastografía mamaria de paciente 2. ....	20
Figura 13. Componentes Tensor de Deformación. ....	23
Figura 14. Ilustración de las relaciones para definir las componentes del tensor de deformación .....	25
Figura 15. Definición de Kernel y región de búsqueda. ....	29
Figura 16. Representación esquemática del seguimiento del desplazamiento del algoritmo multinivel. Modificada de: [11]. ....	30
Figura 17. Representación esquemática de una correspondencia de bloques general.....	32
Figura 18. Ilustración del algoritmo de seguimiento guiado por la calidad .....	33
Figura 19. Ilustración del Método Híbrido de Estimación de Desplazamiento. ....	34
Figura 20. Ilustración de algoritmo multi-resolución modificado de: [11].....	35
Figura 21. Ilustración del mínimo número de propagaciones que hacen que una semilla sea buena. ....	38
Figura 22. Planos de normalización. Izquierda: Plano según (2.2). Derecha: Plano según (2.4). Blanco para deformaciones grandes, negro para deformaciones pequeñas. ....	40
Figura 23. Interfaz de desarrollo Qt .....	43
Figura 24. Interfaz de Qt Designer.....	44
Figura 25. Interfaz Visual Studio 2012. ....	47
Figura 26. Diagrama de casos de uso Elax1 .....	48
Figura 27. Diagrama de relaciones de casos de uso de Elax1 .....	49
Figura 28. Diagrama de clases de Elax1 .....	50
Figura 29. Diagrama de clases del método SSD .....	51
Figura 30. Diagrama de clases del método Híbrido .....	58
Figura 31. Diagrama de clases método AM2D .....	63
Figura 32. Interfaz Elax1 .....	66
Figura 33. Botón agregar imagen de referencia .....	67

Figura 34. Pestaña para la selección del algoritmo .....	67
Figura 35. Ventana auxiliar para configuración de parámetros.....	68
Figura 36. Selección del plano de normalización.....	69
Figura 37. Imagen de deformación ploteada .....	69
Figura 38. Ventana calculo señal a ruido de imagen de desplazamiento .....	70
Figura 39. Resultados obtenidos Phantom de prueba sin normalización .....	73
Figura 40. Resultados del algoritmo SSD sin plano de normalización.....	74
Figura 41. Resultados del algoritmo SSD con el plano de normalización 1.....	74
Figura 42. Resultados del algoritmo SSD con el plano de normalización 2.....	75
Figura 43. Resultados del algoritmo HBD sin plano de normalización.....	75
Figura 44. Resultados del algoritmo HBD con el plano de normalización 1.....	76
Figura 45. Resultados del algoritmo HBD con el plano de normalización 2.....	76
Figura 46. Resultados del algoritmo AM2D sin plano de normalización. ....	77
Figura 47. Resultados del algoritmo AM2D con el plano de normalización 1. ....	77
Figura 48. Resultados del algoritmo AM2D con el plano de normalización 2. ....	78
Figura 49. Relación De Deformación usado en el método de elastografía SSD....	80
Figura 50. Relación De Deformación usado en el método de elastografía HBD. ..	80
Figura 51. Relación De Deformación usado en el método de elastografía AM2D. ....	81
Figura 52. SNR de imagen de deformación del método SSD.....	82
Figura 53. SNR de imagen de deformación del método HBD.....	82
Figura 54. SNR de imagen de deformación del método AM2D. ....	83
Figura 55. Resultados obtenidos Phantom de prueba sin normalización .....	84
Figura 56. Resultados de Elastografía del paciente 3 utilizando el algoritmo SSD. .....	85
Figura 57. Resultados de Elastografía del paciente 3 utilizando el algoritmo HBD. .....	86
Figura 58. Resultados de Elastografía del paciente 3 utilizando el algoritmo AM2D. .....	86
Figura 59. Resultados de Elastografía del paciente 4 utilizando el algoritmo SSD. .....	87
Figura 60. Resultados de Elastografía del paciente 4 utilizando el algoritmo HBD. .....	88
Figura 61. Resultados de Elastografía del paciente 4 utilizando el algoritmo AM2D. .....	88

## LISTA DE TABLAS

Tabla 1. Parámetros de configuración para los algoritmos SSD, HBD y AM2D ....89

# 1 INTRODUCCIÓN

## 1.1 Motivación

Mejorar la detección del cáncer, es uno de los mayores objetivos de la elastografía y es el motivo que lleva a este proyecto a hacer un acercamiento significativo a la elastografía, específicamente a la elastografía mano alzada y al desarrollo de una herramienta software llamada Elax1 que permita visualizar un elastograma con la posibilidad de escoger entre diferentes algoritmos y parámetros.

Para entender más a fondo las razones de la motivación de este proyecto, a continuación se explica de forma general la enfermedad del cáncer y los pasos a seguir para su diagnóstico.

El cáncer es una enfermedad que provoca la multiplicación de un grupo de células de forma descontrolada, estas células se propagan de forma rápida en tejidos u órganos, formando tumores malignos. Generalmente los tumores malignos se caracterizan por tener una consistencia rígida, existen casos que no siguen este patrón, como en la leucemia.

El cáncer generalmente es diagnosticado por medio de una biopsia. Para que este examen sea practicado, primero se necesita ubicar el tumor con la ayuda de una ecografía, tomografía computarizada o resonancia magnética. El médico se vale de estas herramientas para detectar formaciones irregulares en el tejido u órgano, que pueden ser tumores, ya sean benignos o malignos. Una vez detectado la ubicación del órgano o tejido a examinar, se procede a la cirugía para extraer una muestra del tejido, que dependiendo de muchos factores, puede ser sencilla o compleja, posteriormente la muestra pasa a ser analizada en un microscopio.

Una práctica muy útil y sencilla para apoyar al diagnóstico de un posible tumor, es la palpación. La palpación es una práctica médica estándar, que usa el tacto para examinar y hacer un diagnóstico de un tejido u órgano. Pero no todos los órganos o tejidos se encuentran en una zona que se pueda palpar, además la evaluación resulta ser muy subjetiva y por ende sujeta a un alto porcentaje de error en la estimación. En general el diagnóstico por palpación es frecuentemente utilizado para hacer un diagnóstico rápido del cáncer de mama.

Herramientas o técnicas para hacer un diagnóstico preliminar de enfermedades como el cáncer, son muy necesarias para evitar posibles falsas alarmas, y así, disminuir al máximo, el uso innecesario del examen de biopsia, lo que reduce

cualquier tipo de afectación emocional y física que pueda recibir el paciente, y más aún, si se encuentra en un estado delicado de salud [2].

El diagnóstico del cáncer en general, es una de las primeras y principales razones por las que se han propuesto gran variedad de técnicas de ultrasonido para el análisis de la rigidez o elasticidad del tejido biológico, entre ellas se encuentra la elastografía.

A medida que se ha venido desarrollando estas técnicas, han aparecido nuevos enfoques y campos de aplicación entorno a la medicina.

Esta herramienta ha demostrado tener grandes funcionalidades y buenos resultados en los diferentes estudios hechos hasta ahora, por lo que su acogida ha aumentado en los últimos años, y ahora ya se encuentran en el mercado algunas máquinas de ultrasonido con módulos de elastografía implementados, estos módulos de elastografía difieren de una máquina a otra.

Por las razones descritas en este segmento, se decidió abordar el estudio de algunos métodos de elastografía y desarrollar la herramienta software Elax1.

## 1.2 Revisión de la elastografía en la literatura

La elastografía es una técnica que estudia la mecánica de los tejidos blandos basándose en las propiedades elásticas. Esta técnica inició su desarrollo hace más de 20 años por Ophir et al [3], donde este explica que se debe aplicar una compresión al tejido, para obtener las imágenes antes y después de la compresión, luego de obtenidos los datos de compresión, un algoritmo de **búsqueda de bloques**<sup>1</sup> encuentra los desplazamientos de las imágenes a través de una **correspondencia de bloques**<sup>2</sup>, usando generalmente un coeficiente de correlación. Por último se obtiene un campo de **strain**<sup>3</sup>, derivando espacialmente el desplazamiento.

En el año 1998 Yeung et al [4], proponen trabajar la elastografía con nuevos métodos basados en la correspondencia de bloques, que consideren las características particulares de las imágenes de ultrasonido: por ejemplo, la deformación del tejido biológico, la decorrelación de las imágenes ultrasónicas,

---

<sup>1</sup> Los bloques son pedazos de una imagen pre-deformada, y lo que busca el algoritmo, es encontrar a través de una estrategia ese bloque en la imagen post-deformada con una correspondencia de bloques.

<sup>2</sup> Es una medida que me indica que bloque en la imagen de post-deformación corresponde al bloque en la imagen de pre-deformación.

<sup>3</sup> Es una medida de deformación del tejido excluyendo rotación y traslación.

etc. Además presenta los beneficios de desarrollar un algoritmo de búsqueda de bloques en un esquema multinivel, que mejora los costos computacionales, la precisión y resolución del campo de strain.

En el año 2000 Pesavento et al [5], describe un nuevo concepto de procesamiento en tiempo real, para la obtención de imágenes en deformación utilizando técnicas por ultrasonido, basándose en conceptos expuestos por Ophir et al [3]. Este estudio resalta las principales desventajas en el cálculo de la imagen de deformación: En estudios anteriores se concibió como una técnica fuera de línea, que además necesita de grandes compresiones en el tejido en comparación con otras técnicas. Cabe resaltar que en ese momento se hablaba del desarrollo de métodos y máquinas para la obtención de elastogramas, mas no existía un software propietario dedicado a este tema.

El software y hardware para la elastografía ha evolucionado, y a lo largo de los años se han encontrado diferentes métodos para la obtención de imágenes en deformación por elastografía. En [6], [7] hacen un análisis de las máquinas comerciales que han implementado módulos con diferentes métodos, para la visualización de elastogramas, dentro de las cuales se resaltan: transient elastography (TE; Fibroscan®, Echosens, Paris, France), acoustic radiation force impulse imaging (ARFI; Siemens Healthcare, Erlangen, Germany), shear wave elastography (SWE; SuperSonic Imagine S.A., Aix-en-Provence, France), Free-hand elastography ultrasound (Hitachi®), free-hand elastography (Toshiba®), free-hand elastography (Philips®), Free-hand elastography (Siemens®).

La elastografía a mano alzada es una técnica de bajo costo, cuya incorporación en una sonda comercial es sencilla y además es de fácil uso. El elastograma muestra generalmente un mapa de colores o valores de strain relativos obtenidos en comparación con el tejido circundante.

Las sondas han mejorado su sensibilidad, por lo que basta con generar una presión suave en el tejido para la adquisición de una imagen. La resolución en dirección axial de un ecógrafo genérico en la adquisición de los datos **RF**<sup>4</sup> es alta en comparación con la dirección lateral y de elevación, es por esto que generalmente el campo de strain para elastografía es mostrado en dirección axial. Las desventajas que presenta esta técnica es el hecho de que solo puede ser usado para órganos superficiales como la mama, tiroides, próstata, etc. Otra desventaja es que pueden presentarse grandes variaciones en los valores de strain, que pueden alterar las imágenes de deformación mostradas, debido a la dependencia en la habilidad del operador, al momento en el que este aplique presión a la sonda [6].

---

<sup>4</sup> Radio frecuencia.

En la literatura se han encontrado diferentes métodos desarrollados para la obtención del campo de strain [8], [9], [3], [10], [11], [12], [13], [14], [5], [15], [16], [17], [18], [19], [4], [1]. En [9] Lindop estudia el desempeño de métodos de correspondencia de bloques usando algoritmos de Correlación Cruzada Compleja para obtener estimaciones de desplazamiento: Maximum Absolute Value Search (MAVS), Phase zero search (PZS), Efficient phase zero search (EPZS), EPZS con compresión logarítmica (LEPZS), Exhaustive normalised maximum absolute value search (NMAVS).

En general EPZS, LEPZS mostraron buenos resultados con respecto a la calidad de la imagen, siendo el último un poco más superior. Con respecto a la velocidad EPZS tiene velocidad marginalmente más alta. EPZS “se basa en un método gradiente que hace reestimaciones de la posición de fase cero de forma iterativa” [9].

LEPZS es una variación de EPZS que suprime la amplitud mientras que la fase no se altera, esto es para controlar las estimaciones erróneas del strain, que se deben, al suponer que cada estimación de desplazamiento es una medición del desplazamiento en el centro de la ventana. Suposición que no necesariamente es real.

El método híbrido de estimación de desplazamiento es una técnica que combina la virtud del seguimiento de calidad guiada [19], correlación multinivel [17] y búsqueda de fase cero [15], para lograr estimaciones de desplazamiento axial de forma precisa, robusta y con velocidad adecuada para su implementación como herramienta para apoyar el diagnóstico de tumores malignos de tiroides [11]. De acuerdo a las simulaciones y pruebas en vivo comparadas con otras alternativas importantes para la elastografía, se encontró mejoras significativas en cuanto a la reducción de error de estimación de desplazamiento en un rango de 5 a 50%, reducción de errores de **peak-hopping**<sup>5</sup> y reducción de errores debido a tejidos deslizantes. Este método tiene como particularidad la búsqueda del desplazamiento lateral con una resolución baja, que se utiliza para mejorar la estimación del desplazamiento axial. Este método es robusto a señales de decorrelación, y obtiene una buena reconstrucción de las imágenes para **tejidos discontinuos**<sup>6</sup>.

El método de Estimaciones de Strain Adaptativo permite obtener imágenes menos ruidosas eliminando también errores debidos a algunos artefactos de imágenes de ultrasonido, pero con un alto costo computacional. El método consiste en invertir la transformación sufrida en la imagen de post-deformación con un estiramiento de

---

<sup>5</sup> Artefactos causados cuando se hacen estimaciones de desplazamiento gruesas.

<sup>6</sup> Tejidos adyacentes deslizantes, debido a cambios drásticos de rigidez entre sí.

los datos, que revierta los cambios que sufrió la imagen de pre-deformación después de la compresión, para luego obtener el campo de *strain* con una mejor relación señal ruido [20].

En [12] se plantea una mejora de las técnicas de correspondencia de bloques tradicional. Generalmente en la correspondencia de bloques se considera que el valor de la estimación del desplazamiento de una ventana, corresponde al centro de la ventana, esto causa un artefacto “**modulación de amplitud**”<sup>7</sup>. Corrección de amplitud modulada (AMC) es una nueva técnica que encuentra la ubicación óptima para cada estimación de desplazamiento de una ventana.

En [15] se desarrolla una nueva técnica de estimación de deformación basada en fase. Que conjuga los atributos de AMC y un nuevo algoritmo para la estimación del desplazamiento, Separación de Fase Ponderada (WPS). Los resultados determinan que WPS es una estructura sencilla, con un gran rendimiento, además de que se pueden configurar sus atributos para que se adapte a una aplicación específica de la elastografía.

En [17] se estudia un método de estimación de *strain* multinivel para tejidos discontinuos. En varios estudios se ha considerado el tejido de forma uniforme, pero no siempre es el caso, lo que puede afectar la estimación de la tensión. Se pueden hallar discontinuidades por ejemplo en el seno por deslizamiento de paredes adyacentes o entre las paredes de la arteria carótida. Aquí se proponen niveles de baja resolución que busquen estimaciones de desplazamiento en la **imágenes de modo B**<sup>8</sup>, de esta forma se calculan las tendencias globales del tejido, y un segundo nivel de alta resolución que use los datos RF y las estimaciones anteriores como guía, para encontrar las estimaciones de desplazamiento óptimas.

En [16] se desarrolló un nuevo método de auto-correlación en tiempo real, el cual suplirá las necesidades de mayor procesamiento y precisión a la hora de obtener imágenes de elastografía. El método de auto-correlación combinada CAM se basa en dos pasos: el primer paso es hacer una estimación ordinaria de la **correlación máxima de la envolvente**<sup>9</sup>. Como segundo paso, se hace una estimación más fina con los datos obtenidos en el primer paso utilizando los datos de fase de la no envolvente.

---

<sup>7</sup> Es un artefacto que sucede porque la ubicación real de la estimación dentro de una ventana se concentra en el lugar donde la señal de amplitud es más alta. Es decir la estimación del desplazamiento de la ventana no corresponde necesariamente al centro de la ventana, como se ha supuesto siempre.

<sup>8</sup> Imagen de típica ecografía que muestra la amplitud.

<sup>9</sup> Imagen en modo-B.

En [19] se estudia un método de seguimiento de calidad guiada, que busca obtener resultados precisos de estimaciones de *strain* al usar una estructura multinivel. Las estrategias de seguimiento de desplazamiento actuales hacen una **propagación de semillas**<sup>10</sup> en direcciones específicas, por lo que errores debido a la mala calidad de algunas semillas se pueden propagar. Esto puede terminar en una mala estimación. Se propone una propagación de las semillas guiadas por la calidad de los datos para solucionar estos problemas.

En [1] es presentado una técnica para obtener imágenes de deformación basados en la minimización de una función de costos utilizando programación dinámica, además de implementar técnicas de suavizado en la función de costo para reducir ruidos de correlación.

La elastografía se está utilizando para aplicaciones generales como el apoyo en la detección y caracterización de tumores cancerígenos de mama, próstata, tiroides, hígado, riñones, cerebro, páncreas, esófago, estómago y suprarrenales. También está siendo utilizada para ayudar al diagnóstico de enfermedades genitourinarias: miomas, adenomatosis, incompetencia cervical, hiperplasia, y enfermedades cardiovasculares: infarto, ateromas, aneurismas [21], [17]. Específicamente la aplicación de la elastografía a mano alzada en la caracterización de lesiones mamarias está siendo muy estudiada [22], con muy buenos resultados, siendo uno de los principales objetivos de la elastografía mamaria [23], [24]. Ya ha sido validada en la mayoría de las lesiones de mama que se han sometido a biopsia [25].

Carlos Gaviria en [26] desarrolló un primer trabajo en la Universidad del Cauca referente a la elastografía con técnicas de mano alzada. Se hizo una revisión general y un primer acercamiento a una técnica de elastografía utilizando correspondencia de bloques y búsqueda de ventanas de forma exhaustiva, mediante pruebas en **Phantoms**<sup>11</sup>, que permitieron obtener datos e imágenes aproximadas para el cálculo del *strain*.

Dentro del desarrollo de aplicaciones, para el cálculo del *strain*, Joel Lindop et al [18], resalta en su trabajo una herramienta software para la visualización y construcción de imágenes en 2D y 3D, a partir de datos en RF. En esta herramienta se resalta el uso de una pestaña donde se configura los parámetros, para el cálculo del *strain*, utilizando diferentes modelos de algoritmos propuestos en dicho trabajo. A diferencia del desarrollo planteado en [18], en este trabajo, se desarrollara una herramienta aplicada únicamente a la elastografía, que permita

---

<sup>10</sup> Datos de desplazamiento de un nivel grueso, usados como punto de partida para estimar desplazamientos vecinos en un nivel más fino.

<sup>11</sup> Phantom se refiere a un cuerpo artificial, que presenta características similares a los de un tejido humano.

visualizar imágenes de deformación, que sirvan como fase exploratoria, para que a futuro, pueda servir en estudios para el apoyo de diagnóstico en tiempo real, en la detección de anomalías en los tejidos **in vivo**<sup>12</sup>.

### **1.3 Alcances y objetivos**

La elastografía a mano alzada ha venido tomando mucha fuerza a lo largo de los años y aún hasta ahora se siguen desarrollando nuevas técnicas y mejoras. Este método está incursionando en muchos campos de la medicina donde ya se están estudiando y desarrollando prácticas clínicas estándar para la obtención y análisis de los resultados. Con el fin de tener un acercamiento más amplio y cercano a las diferentes bondades que ofrece la elastografía se plantean los siguientes objetivos generales y específicos a seguir durante la ejecución de este proyecto.

#### **1.3.1 Objetivo general**

Diseñar e implementar un sistema para determinar el grado de dureza de tejidos biológicos, aplicando técnicas de elastografía por ultrasonido.

#### **1.3.2 Objetivos Específicos**

- Calcular una imagen de deformación que permita establecer el grado de dureza de tejidos a partir de imágenes de ultrasonido.
- Seleccionar diferentes parámetros y algoritmos de procesamiento de imágenes en una interfaz gráfica, para desplegar información sobre la dureza de los tejidos.
- Verificar el desempeño del sistema, a partir de un conjunto de imágenes de prueba conocidas.

### **1.4 Estructura de la monografía**

El trabajo se divide en seis capítulos: (1) Introducción; (2) Elastografía: Conceptos generales; (3) Algoritmos para el cálculo de la deformación; (4) Implementación y simulación; (5) Resultados; y (6) Conclusiones.

El capítulo 1, presenta la importancia de la elastografía en el campo médico, en específico en el diagnóstico del cáncer y la necesidad de tener un acercamiento a

---

<sup>12</sup> In vivo se refiere al escaneo del tejido de un organismo vivo.

las diferentes técnicas que existen, también es presentada una revisión general de los diferentes tipos y técnicas de la elastografía, aplicaciones, el alcance y objetivos del proyecto, la estructura de la tesis y las contribuciones.

El capítulo 2, presenta los principios generales de la elastografía, la mecánica de los tejidos y el ultrasonido.

En el capítulo 3, se presentan y definen todas las técnicas y algoritmos a tratar por este proyecto. Este capítulo se divide en algoritmos para el cálculo del desplazamiento y algoritmos para el cálculo del gradiente. Los algoritmos para el cálculo del desplazamiento que serán estudiados son tres, de entre ellos, solo el algoritmo de Búsqueda de bloques exhaustiva y el algoritmo de Estimación de desplazamiento híbrido serán implementados en este proyecto, mientras que el algoritmo AM2D ya ha sido implementado en [19], y solo hará parte de este documento para que por medio de la comparación de resultados, se pueda evaluar el desempeño de los demás algoritmos y para que haga parte del software Elax1 que será explicado e implementado en el capítulo 4.

En el capítulo 4, se hace una descripción de la implementación del software Elax1 y de los algoritmos que lo componen, así como los diferentes tipos de software y librerías que se usaron para la ejecución de este proyecto.

En el capítulo 5, son presentados, comparados y analizados los resultados de los algoritmos para el cálculo de la deformación.

Finalmente, el capítulo 6, culmina el proyecto con las conclusiones y los trabajos que se pueden desarrollar a futuro.

La información adicional de este trabajo se puede encontrar en los siguientes anexos:

- Anexo A: Manual de instalación.
- Anexo B: Creación de recursos e interfaces.
- Anexo C: Creación de librería compartidas en Matlab.
- Anexo D: Manual de usuario Elax1.

## 2 ELASTOGRAFÍA: CONCEPTOS GENERALES

La elastografía ha sido estudiada por varios años, y en diferentes partes del mundo, lo que ha permitido el desarrollo de una gran cantidad de enfoques y técnicas y el desarrollo del hardware necesario para que estos enfoques se puedan implementar.

La mayoría de los enfoques y técnicas de elastografía necesitan de sistemas para la obtención de imágenes. El ultrasonido fue el primer sistema de adquisición de imágenes utilizado para la obtención de imágenes de elastografía. En los últimos años la elastografía ha venido incursionando en métodos basados en Imágenes de Tomografía de Coherencia Óptica y Resonancia Magnética (**MRI**) también llamado Tomografía por Resonancia Magnética (**TRM**). **MRI** es una técnica que usa olas de campo magnético y de radio para crear imágenes detalladas de los órganos y tejidos en el cuerpo [27], esta técnica aplicada en la elastografía es llamada Elastografía por Resonancia Magnética (**MRE**), este es un método que ha cobrado mucha fuerza por su alta resolución tanto en el eje axial, como lateral y de alzado, lo que recae en una mayor precisión a la hora de obtener las imágenes de elastografía. Generalmente las imágenes de elastografía muestran la deformación en dirección axial, en este eje, las imágenes de ultrasonido se equiparan en resolución con las de **MRI**, a diferencia del eje lateral y de alzado donde la resolución es pobre. La precisión de la deformación axial usando imágenes de ultrasonido se ve afectada, debido a que en la mayoría de técnicas, el desplazamiento en dirección lateral del tejido es utilizado de forma indirecta para el cálculo de la deformación axial.

El **MRE** resulta también muy atractivo por su capacidad de obtener grandes volúmenes de imágenes de elastografía en 3D con alta resolución en los tres ejes. Cabe notar que imágenes de elastografía en 3D también son posibles utilizando imágenes de ultrasonido, pero con una precisión y resolución menor que en **MRE**.

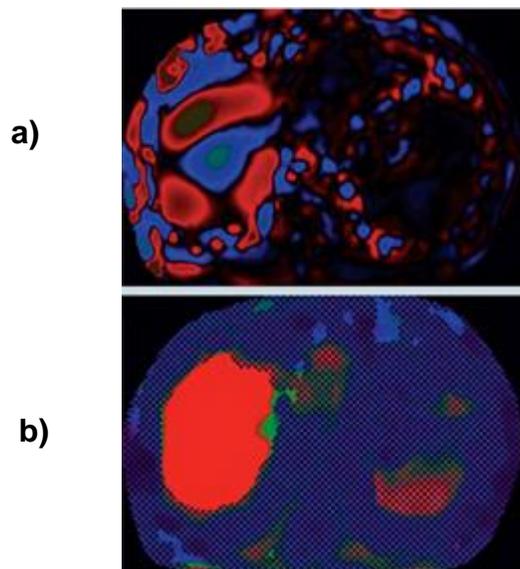
**MRE** es principalmente utilizado para detectar el endurecimiento del hígado causado por enfermedades hepáticas crónicas, en [27] figura que en Estados Unidos, los estudios han demostrado gran precisión para detectar la fibrosis hepática.

Una de las principales desventajas del **MRE** es su alto costo. En la Figura 1 y 2 se observa un equipo de **MRI** e imágenes de elastografía utilizando **MRE**, respectivamente.



**Figura 1. Tomografía computarizada por resonancia magnética.**

**Fuente: [28]**



**Figura 2. Imágenes de ondas, a) muestra la propagación de ondas de corte a través del parénquima hepático, b) una imagen de elastografía con valores de rigidez elevados con una media de 15 kPa consistente de fibrosis.**

**Fuente: [29]**

El ultrasonido es el sistema de obtención de imágenes de tejido y órganos que se estudia en este proyecto. A continuación es explicado el funcionamiento de una máquina de ultrasonido.

Una máquina de ultrasonido permite visualizar una imagen, o una secuencia de imágenes, que muestren tejidos y órganos en el interior del cuerpo humano; para esto un transductor envía ondas de sonido a alta frecuencia (1 a 5 MHz), imperceptibles al oído humano. Estas ondas generan ecos al rebotar con los límites de cada tejido (cuando cambia la consistencia o dureza de un tejido de una región a otra), alguna de las ondas se reflejan antes de llegar a un límite. Las ondas son generadas y recibidas mediante el principio del efecto piezoeléctrico; una sonda posee uno o más cristales piezoeléctricos, que al recibir señales eléctricas, vibran y producen ondas sonoras. Los cristales también son capaces de recibir y leer las ondas sonoras, porque al ser golpeados por estas, emiten corrientes eléctricas.

El transductor recibe los ecos, que guarda en forma de valores de distancia e intensidad, estos valores son dependientes de las propiedades del tejido.

La máquina de ultrasonido calcula la distancia de los límites de cada tejido con el conocimiento de la velocidad de la onda en el tejido, y del tiempo que la onda tarda en volver. En la máquina de ultrasonido se puede visualizar la forma de los tejidos, tamaños y consistencias, tal como se muestra en la Figura 3.



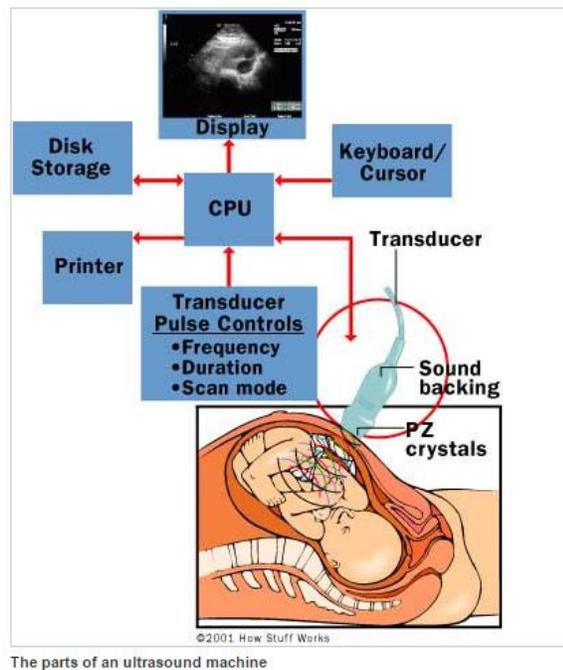
**Figura 3. Ecografía típica de un feto.**

**Fuente: [30]**

Una máquina de ultrasonido se compone de la siguiente manera:

- Sonda transductora: Envía y recibe las ondas de sonido.
- Unidad de procesamiento central (CPU): hace los cálculos y suministra de energía a la sonda.
- Control de pulsos del transductor: Cambia la amplitud, frecuencia y duración de los pulsos emitidos por el transductor.
- Pantalla: Muestra imágenes de los datos de ultrasonido procesados por la CPU.
- Teclado / cursor - entradas de datos y toma mediciones de la visualización.
- dispositivo de almacenamiento en disco - almacena las imágenes adquiridas.
- Impresora - imprime la imagen de los datos visualizados.

En la Figura 4 es mostrada una ilustración de los componentes de una máquina de ultrasonido.



**Figura 4. Representación de los componentes de una máquina de ultrasonido.**

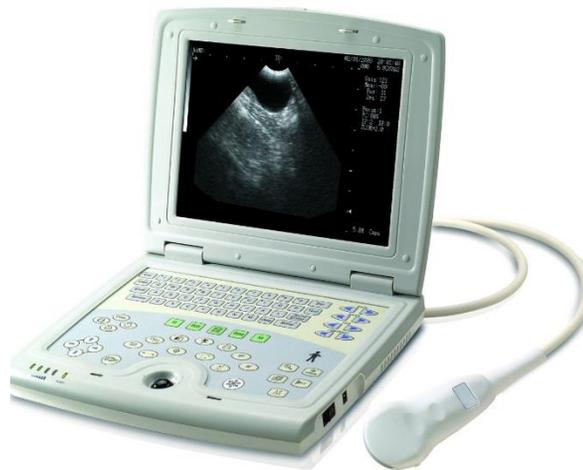
**Fuente: [30]**

En la Figura 5 y 6 son mostradas ilustraciones de máquinas de ultrasonidos convencionales y portables, respectivamente.



**Figura 5. Máquina de ultrasonido convencional.**

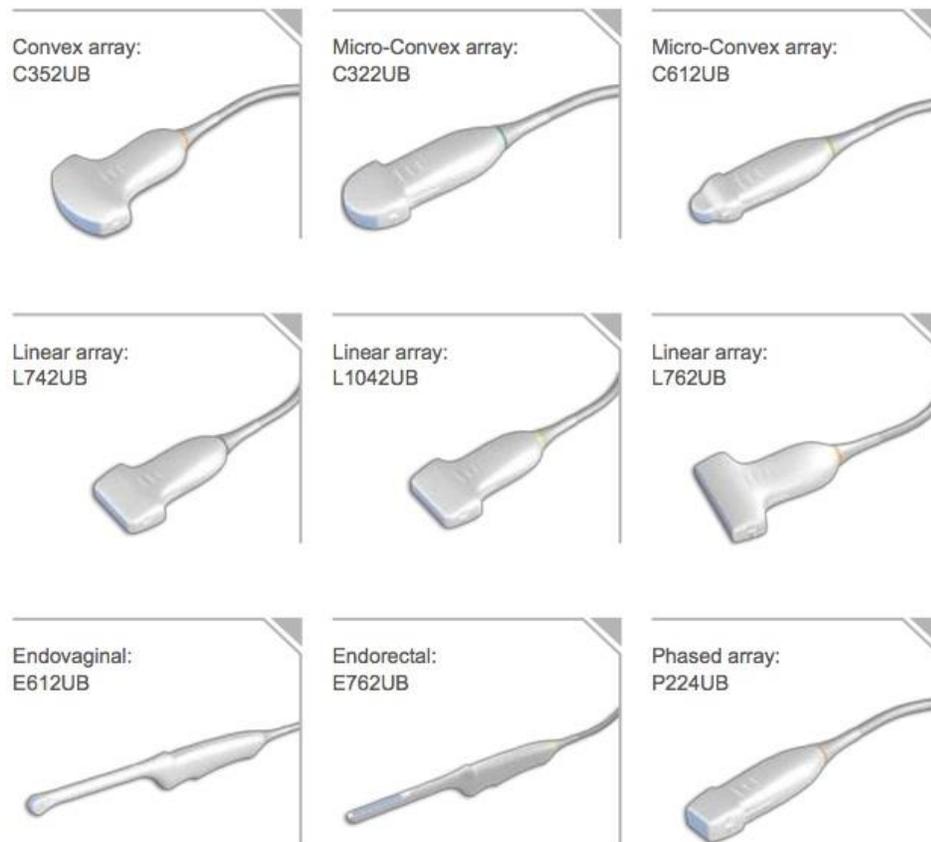
**Fuente: [31]**



**Figura 6. Máquina de ultrasonido portable.**

**Fuente: [32]**

Existen gran variedad de sondas con diferentes formas y para diferentes aplicaciones, véase en la Figura 7. La forma de la sonda determina el campo de visión, que a su vez va determinado por la aplicación o el caso clínico en el que se vaya a trabajar. Hay sondas que han sido diseñadas específicamente para que sean insertadas por aberturas del cuerpo humano, como la vagina, el recto y el esófago, esto con el fin de conseguir mejor calidad de imágenes de ultrasonido de órganos que están en el interior del cuerpo, por ejemplo: útero, próstata, estomago.



**Figura 7. Tipos de sonógrafos.**

**Fuente: [33]**

En la actualidad existen ultrasonidos Doppler (para medir el flujo de sangre) y ultrasonidos que pueden conseguir imágenes en 3D. Este último mejora la visualización de un órgano; para nuestro caso de estudio, es especialmente útil, porque permite determinar de mejor manera, la condición de un tumor; pero no

está al alcance de este proyecto, por lo tanto no se hará un estudio más a fondo en esta dirección.

Las imágenes de ultrasonido tienen muchos usos en la medicina, de entre ellos, la competencia de este proyecto, el diagnóstico. El médico se basa en las imágenes de ultrasonido para hacer el diagnóstico de diferentes condiciones que pueden afectar a un órgano o tejido blando, más específicamente para este trabajo, el cáncer. El médico con la ayuda de una máquina de ultrasonido, puede identificar un tumor, cuando encuentra un bulto o una formación del tejido que sea fuera de lo normal.

Otro uso muy importante de las imágenes de ultrasonido y que vale la pena resaltar, es la posibilidad de guiar al médico en operaciones mínimamente invasivas, como la biopsia, donde se necesita conocer la ubicación exacta del tumor.

Las ventajas del ultrasonido son muchas, algunas de ellas son:

- Es indoloro, no requiere ningún tipo de inyección o incisión.
- No hay efectos dañinos, ni secundarios, tampoco hay exposición a radiación, al contrario de las técnicas de radiografía y tomografía computarizada.
- Existen tejidos blandos en los que la visualización resulta mejor con técnicas de ultrasonido.
- Es altamente portable, accesible y menos costoso que otros métodos.
- Arroja imágenes en tiempo real, lo que da la posibilidad de guiar al médico en medio de una operación.

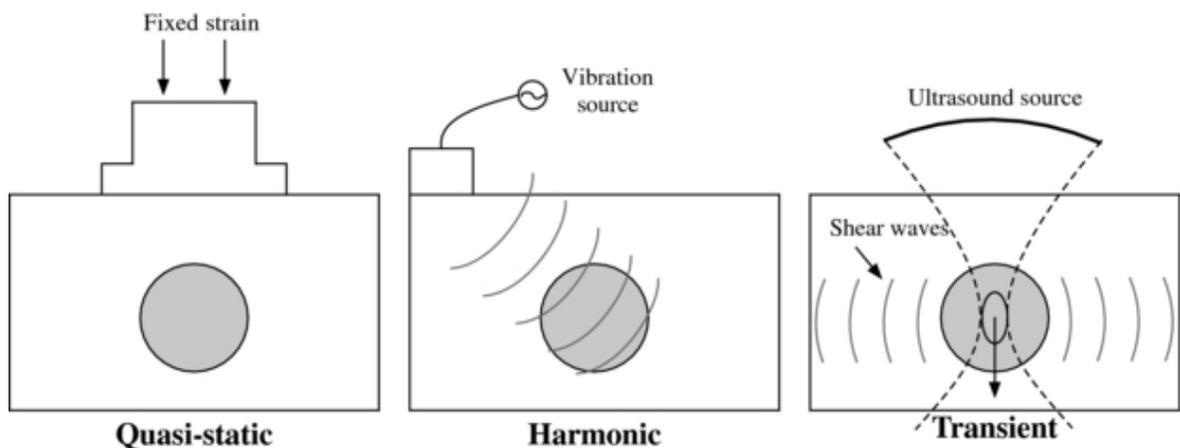
Las desventajas del ultrasonido son las siguientes:

- Existen partes del cuerpo donde resulta difícil obtener una imagen de ultrasonido ya sea por sus características (huesos densos, intestinos) o porque están muy al interior del cuerpo.
- En el diagnóstico de cáncer, no se logra distinguir un tumor maligno de uno benigno, por lo que es muy común obtener falsos positivos.
- El tumor resulta muchas veces difícil de visualizar, además de que puede mostrar una geometría poco exacta y confusa.

Con lo anterior se puede concluir que el ultrasonido por sí solo, no es totalmente confiable a la hora de apoyar al diagnóstico de un tumor maligno. Esta es una de las razones por las que se recomienda hacer diagnósticos de cáncer con imágenes de ultrasonido apoyadas por imágenes de elastografía. Las imágenes de elastografía resuelven algunas desventajas de las imágenes de ultrasonido que hacen que el diagnóstico sea más preciso y confiable.

En términos generales, la elastografía consiste en aplicar una tensión conocida a un órgano o tejido blando, utilizando una fuente casi estática, armónica o transitoria; la deformación causada por la tensión aplicada, se mide, usando técnicas basadas en ultrasonido, resonancia magnética o resonancia óptica; y finalmente las propiedades mecánicas de los tejidos se infieren a partir de los tipos de resultados que arroje la técnica de elastografía usada [34].

Existen tres enfoques generales de la elastografía, que parten de la forma en que se introduce la deformación en el órgano o tejido blando, véase en la Figura 8.



**Figura 8. Enfoques generales de las elastografía.**

**Fuente: [35]**

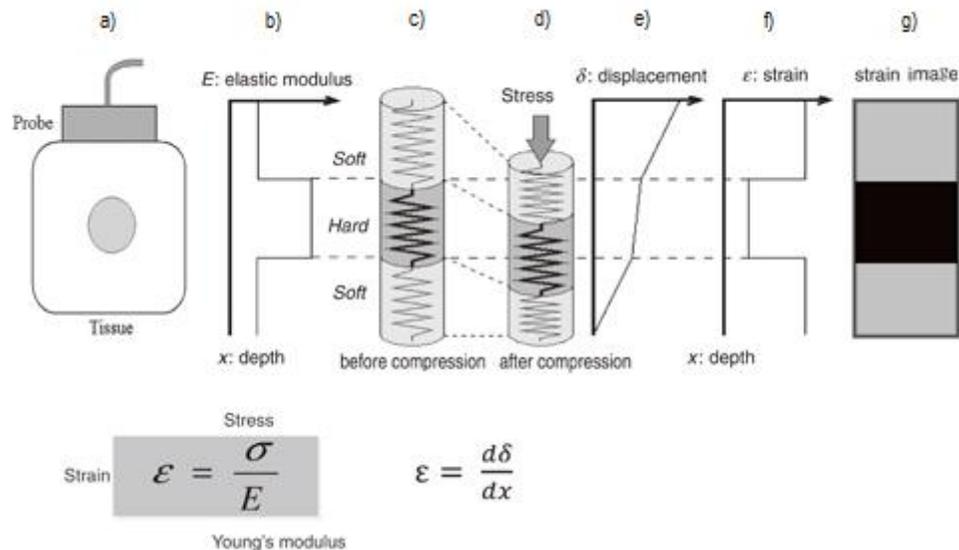
La elastografía Casi Estática fue el primer enfoque desarrollado para obtener imágenes de elasticidad de tejidos blandos.

Este procedimiento consiste en [18]:

- Obtener una imagen de pre-deformación sin aplicar ningún tipo de tensión al tejido.

- Comprimir el tejido de forma casi estática.
- Obtener la imagen de post-deformación.
- Calcular la deformación del tejido comparando las imágenes de pre y post compresión, usando técnicas de procesamiento de señales.

En la Figura 9 se ilustra la elastografía casi estática típica.



**Figura 9. Elastografía casi estática típica.**

**Modificado de: [36]**

En la figura 9, a) una sonda es utilizada para obtener imágenes de pre compresión y post compresión de un tejido con un tumor en su interior. b) El módulo de Young (propiedad elástica del tejido) es mayor en el tumor que en el tejido adyacente. c) el tejido para este ejemplo, se aproxima a modelo 1D con un conjunto de resortes donde sus constantes de resorte representan su módulo de Young, el resorte en negro es más rígido que los resortes en gris. d) una fuerza se aplica en el tejido, generalmente debido a una presión ejercida en la sonda, esta fuerza genera un estrés en cada elemento del tejido (resorte) que a su vez genera una deformación, e) el desplazamiento en el tejido es calculado, f) la deformación ( $\epsilon$ ) se obtiene por la diferenciación espacial del desplazamiento; si el estrés en todo el tejido es uniforme la deformación será inversamente proporcional al módulo de Young. En g) las intensidades de los píxeles revela la variación del módulo de Young para cada resorte.

Existen diferentes formas de generar una deformación en el tejido a un ritmo casi estático [9]; la más común se llama a mano alzada y consiste en que el médico presione ligeramente la sonda de ultrasonido contra el tejido hasta que este sea comprimido en aproximadamente 1%. Por esta razón, depende de la pericia del médico conseguir buenas imágenes, pero el hecho de que las imágenes de ultrasonido y de elastografía se obtengan en tiempo real, le dan la posibilidad al médico de corregir sus movimientos hasta obtener imágenes más certeras.

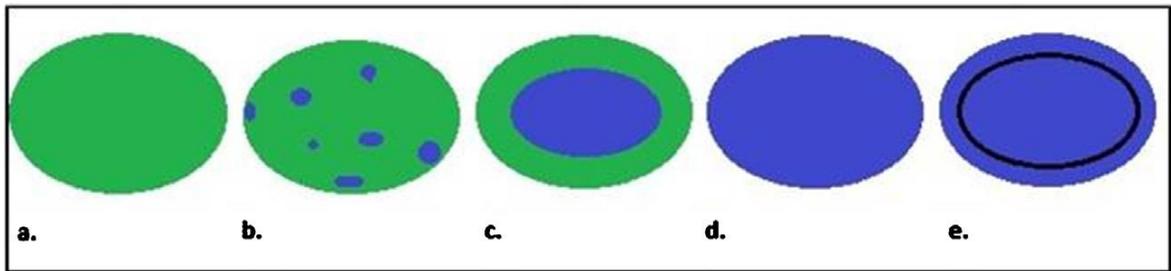
También se puede comprimir el tejido de forma más definida y exacta utilizando un actuador en la sonda que controle la presión ejercida en el tejido, esto hace que un movimiento sea repetible y que los datos obtenidos sean más confiables. Este es un método que puede ser útil para determinadas aplicaciones, por ejemplo, en la estandarización de un proceso a la hora de realizar un estudio médico de elastografía; pero en una práctica clínica habitual resulta engorroso [18].

Existen casos en que se obtienen imágenes de elastografía usando otros medios para generar la deformación cuasi estática en el tejido, por ejemplo, cuando la compresión es generada por la misma respiración del paciente o en el caso del diagnóstico de enfermedades cardiovasculares, cuando la compresión de una arteria se logra gracias al cambio de la presión arterial durante el ciclo cardiaco [37].

Los valores de deformación arrojados por la Elastografía Casi Estática son medidas relativas de elasticidad, dado que dependen, tanto del módulo de elasticidad del tejido, como de la magnitud de compresión o tensión aplicada al mismo [34]; es por esto que la Elastografía Casi Estática muestra resultados cualitativos, ya sea en una imagen en escala de grises o de colores; también es posible arrojar datos semicuantitativos, dados por la necesidad de tener datos objetivos que permitan mayor seguridad y fiabilidad a la hora de identificar una lesión.

Los resultados cualitativos que arrojan las máquinas comerciales son mostrados a través de un mapa de colores superpuesto en las imágenes de modo B. cada pixel de la imagen de elasticidad representa la cantidad de movimiento que se produjo en esa ubicación del tejido [7].

En la evaluación cualitativa de la elastografía mamaria se ha propuesto algunos sistemas de puntuaciones [25, 38, 39] para describir el comportamiento elástico de una lesión, siendo la clasificación de Ueno-Itoh [25] la más utilizada en la elastografía mamaria. La clasificación de Ueno-Itoh es mostrada en la Figura 10.



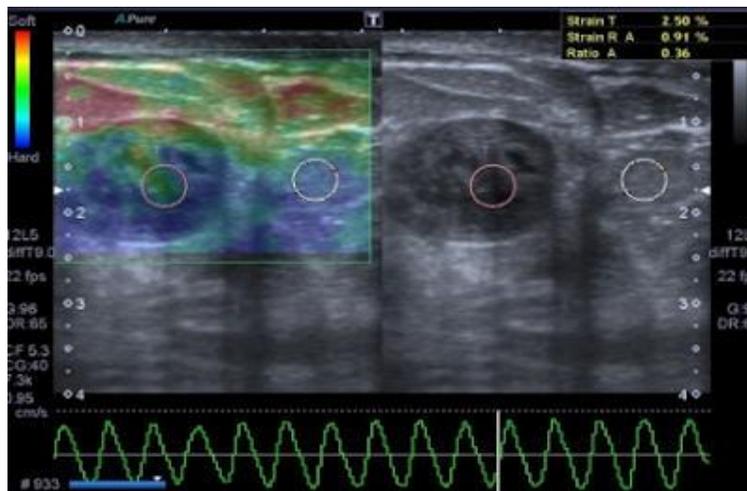
**Figura 10. Criterios de evaluación en tiempo real para lesiones mamarias de acuerdo con las puntuaciones Itoh et al.**

En la figura 10, (a) la puntuación de 1 indica una lesión hipoecoica entera uniformemente sombreada en verde. (b) La puntuación de 2 es asignada a lesión hipoecoica con un patrón de mosaico de verde y azul. (c) La puntuación de 3 indica elasticidad con deformación en la periferia de la lesión hipoecoica (verde), y con escasas de deformación en el centro de la lesión (azul). (d) La puntuación de 4 indica deformación en la periferia (azul) y sin deformación en toda la lesión hipoecoica. (e) La puntuación de 5 indica ausencia de deformación en toda la lesión hipoecoica y en sus alrededores (ambos representados en azul).

En [25] fue hecho un examen de 111 nódulos (59 benignos, 52 malignos, <30 mm de tamaño), asumiendo un punto de corte entre las puntuaciones de 3 y 4, la elastografía casi estática tenía 86,5% de sensibilidad (tasa de resultados positivos cuando la enfermedad está presente), 89,8% de especificidad (tasa de resultados negativos cuando hay ausencia de la enfermedad) y 88.3 de precisión (tasa de resultados verdaderos). Usando un BI-RADS de valor de corte entre 4 y 5 la elastografía casi estática alcanzó 71,2% de sensibilidad, 96,6% de especificidad y 84.7 de precisión.

Resultados semicuantitativos son alcanzados por medio de una “relación de deformación” y una “relación longitudinal”.

La relación de deformación consiste en calcular la deformación media de una región de interés dentro del tumor y la deformación media de una región de interés de un tejido sano, adyacente y horizontal al tumor, véase la Figura 11 y Figura 12.



**Figura 11. Elastografía mamaria de paciente 1.**

En la figura 11 es mostrada una imagen de elastografía mamaria de una mujer de 47 años de edad con un pequeño bulto en el pecho. El ultrasonido de referencia muestra un nódulo hipo isoecoico con márgenes regulares y bien definidos, la Elastografía demuestra un patrón 2 de acuerdo con la clasificación Ueno-Itoh con prevalencia de elasticidad. La Relación de deformación, medida mediante la colocación de dos regiones distintas de interés (colocado en el tejido normal de mama y dentro del nódulo), fue de 0,36. La patología bruta confirmó un fibroadenoma benigno.



**Figura 12. Elastografía mamaria de paciente 2.**

**Fuente: [40]**

En la figura 12 es mostrada una imagen de elastografía mamaria de una mujer de 45 años de edad con una masa palpable en el cuadrante inferior exterior a 2 cm de distancia del pezón. No había eritema, equimosis, ulceración de la piel, o formación de hoyuelos y no se detectó linfadenopatía axilar. El ultrasonido de referencia muestra un bulto isoecoico con bordes regulares, con algunas áreas quísticas dentro. En la elastografía, el nódulo presenta un mapa de color de puntuación de 4 según la clasificación Ueno-Itoh con la ausencia de elasticidad, representado como una zona completamente azul con un resplandor periférico diferente. Con la colocación de dos regiones distintas de interés (en el tejido normal de mama y dentro del nódulo), se obtiene una relación de deformación obtenida de 8,96. Anatomía patológica macroscópica confirmó un cáncer quístico.

La relación longitudinal se obtiene de dividir la longitud horizontal máxima del tumor observado en las imágenes de elastografía casi estática, por la longitud horizontal máxima del tumor observado en las imágenes de modo B. Este elemento de evaluación se debe, a que en la mayoría de las ocasiones, los tumores benignos se ven de un tamaño menor o igual en las imágenes de elastografía casi estática en relación con el tamaño del tumor observado en las imágenes de modo B; mientras que un tumor maligno se ve de un tamaño mayor en las imágenes de elastografía casi estática, comparado con el tamaño del tumor observado en las imágenes de modo B.

Algunos estudios hechos, evalúan el desempeño del diagnóstico de la elastografía, al emplear la relación de deformación o relación longitudinal para caracterizar lesiones mamarias focales. En [25] se describen los resultados encontrados por Sadigh et al, y se hace un análisis de los estudios realizados en [41, 42, 43, 44, 45], con base al diagnóstico por relación de deformación y en [39, 46, 47], con base al diagnóstico por relación longitudinal. Los estudios basados en el diagnóstico por relación de deformación, evaluaron un total de 2087 masas mamarias, los resultados dieron un 88% de sensibilidad y 83% de especificidad. Los tres estudios basados en el diagnóstico por relación longitudinal, evaluaron un total de 450 masas mamarias con resultados que muestran un 98% de sensibilidad y un 72% de especificidad. La relación de deformación y relación longitudinal dieron un aumento en la especificidad de los resultados del diagnóstico de las lesiones mamarias.

## **2.1 ¿Señales RF o imágenes modo B?**

Los escáneres comerciales utilizan módulos propietarios que transforman las señales de radiofrecuencia (RF) crudas desde los transductores, a imágenes procesadas de alto nivel que pueden interpretar los operarios. En esta revisión se encontró que la mayoría de los aportes acerca de elastografía desde imágenes de

ultrasonido, utilizan las señales de RF en lugar de las imágenes modo B de alto nivel, si bien estas señales RF se deben convertir a imágenes modo B antes de utilizar los algoritmos de procesamiento. En [48] se muestra como obtener señales RF desde escáner comerciales, y se justifica la razón por la cual es conveniente tener disponibles las señales RF como se explica a continuación. La imagen desplegada en una pantalla de escáner ultrasónico, conocida como modo B (o envolvente) y usada para el diagnóstico médico, contiene menos información que la señal RF original recibida desde el transductor. La señal desplegada sufre un proceso de muchos pasos de procesamiento como son: compensación de la ganancia en el tiempo, filtrado, rectificación y compensación logarítmica. Para un transductor sectorial, la imagen resultante se interpola desde los datos crudos para convertir las coordenadas polares de la señal adquirida, en las coordenadas cartesianas de la pantalla. Aún más, cada escáner realiza esas operaciones a su propio modo. Debido a la especificidad del proceso de cada escáner, las imágenes desplegadas en la pantalla son difíciles de comparar de una máquina a otra. En la práctica, la imagen modo B es una representación de la envolvente de la amplitud de la señal de eco, la cual, una vez procesada, ha perdido su proporcionalidad a la respuesta de presión del medio. La pérdida de información debida a ese procesamiento es crítica para tomar una ventaja total de la señal recibida en términos de procesamiento de señal y diagnóstico automático.

## 2.2 Fundamentos de elastografía

Se presentan a continuación los fundamentos de la mecánica y conceptos de ultrasonidos necesarios para comprender el procedimiento de elastografía. Estos conceptos han sido tomados de [49].

## 2.3 Mecánica

### 2.3.1 Tensión (Stress)

Supongamos una incisión imaginaria sobre un pequeño plano  $\Delta S$  en un cuerpo deformable continuo que es sometido a fuerzas externas. A no ser que las tensiones sobre el elemento de área sean solo de compresión directa, habrá algún movimiento relativo entre las caras opuestas del área de la incisión. En general, esas caras se apartarán ligeramente obedeciendo a lo que se denominan tensiones normales o axiales (*axial stress*). También se deslizarán como resultado de lo que se denominan tensiones cortantes (*shear stress*). Antes de la incisión, cada cara había ejercido una fuerza  $\vec{\Delta F}$  sobre la otra cara. La razón  $\frac{\Delta F}{\Delta S}$  recibe el nombre de tensión promedio actuando sobre  $\Delta S$ . Si esta es uniforme, actuaría con

la misma intensidad sobre los elementos de la superficie. Los componentes del vector de tensión tienen unidades de presión (Pascal). Si no es uniforme, se define el vector de tensión como:

$$\vec{t} = \lim_{\Delta S \rightarrow 0} \frac{\Delta \vec{F}}{\Delta S} \quad (1)$$

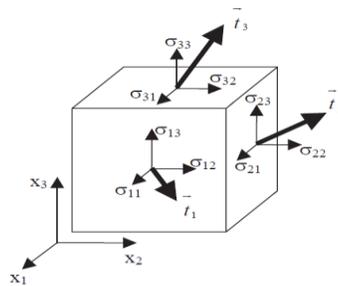
La orientación del elemento de superficie  $\Delta S$  puede definirse por el vector unitario  $\vec{n}$  normal a la superficie y orientado hacia la otra cara. Las tensiones en el punto P sobre la superficie  $\Delta S$  pueden resolverse en una deformación normal  $\sigma_n$  dada por el escalar,

$$\sigma_n = \vec{t} \cdot \vec{n} \quad (2)$$

Y una deformación cortante  $\sigma_t$  dada por el escalar,

$$\sigma_t = \sqrt{\vec{t} \cdot \vec{t} - \vec{t} \cdot \vec{n}} \quad (3)$$

Un valor negativo para  $\sigma_n$  indicaría una deformación de compresión a lo largo de la normal impartida desde la cara opuesta, mientras que un valor positivo indicaría una deformación de estiramiento. En un sistema coordenado  $O_{x_1, x_2, x_3}$  en el punto P existe una componente de deformación con subíndices impares, como este  $\sigma_{12}$ , que denotan deformaciones cortantes, mientras que los subíndices pares denotan deformaciones axiales, laterales y alzadas.



**Figura 13. Componentes Tensor de Deformación.**

**Fuente: [49]**

fuerza-área de un componente de fuerza a lo largo de la dirección positiva  $x_2$ , y actuando sobre un elemento de área plana con normal saliendo de ella en la dirección positiva  $x_1$ . En este caso, existen 9 componentes como se ilustra en la figura 13 y que se resumen por:

$$|\Sigma| = \begin{bmatrix} \sigma_{11} & \sigma_{21} & \sigma_{31} \\ \sigma_{12} & \sigma_{22} & \sigma_{32} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \quad (4)$$

### 2.3.2 Deformación (Strain)

La deformación se define como la variación relativa de la distancia entre partículas de un medio continuo bajo una tensión mecánica. Considere dos puntos P and Q en un conjunto de ejes orto normales. Sean  $dx_1, dx_2$  y  $dx_3$  las tres componentes del vector  $\overrightarrow{PQ}$ . La distancia entre esos dos puntos es:

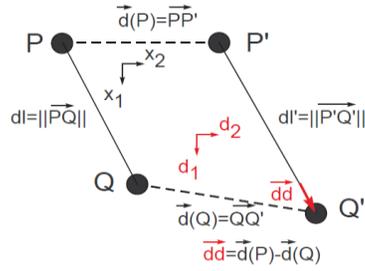
$$d_l = \sqrt{\sum_{i=1}^3 dx_i^2} \quad (5)$$

Cuando el sólido es deformado, esos dos puntos se mueven a P' y Q' respectivamente. Siendo  $\vec{d}(P) = \overrightarrow{PP'}$  el desplazamiento del punto P; y  $\vec{d}(Q) = \overrightarrow{QQ'}$  el desplazamiento del punto Q, la variación en el desplazamiento entre los puntos P y Q es  $\vec{d}\vec{d}(P, Q) = d(Q) - d(P)$ . Luego, la distancia entre los puntos P' y Q' se puede escribir como:

$$dl' = \sqrt{\sum_{i=1}^3 (dx_i + dd_i)^2} \quad (6)$$

Que puede escribirse como,

$$dl'^2 = dl^2 + 2 \sum_{i=1}^3 (dd_i dx_i) + \sum_{i=1}^3 d_i^2 \quad (7)$$



**Figura 14. Ilustración de las relaciones para definir las componentes del tensor de deformación**

Si los componentes del vector de desplazamiento  $dd(P, Q)$  se escriben como,

$$dd_i = \sum_{j=1}^3 \frac{\partial d_i}{\partial x_j} dx_j \quad (8)$$

Entonces (7) se puede escribir,

$$dl'^2 = dl^2 + \sum_{i=1}^3 \sum_{j=1}^3 \varepsilon_{ij} dx_i dx_j \quad (9)$$

Donde las componentes de tensión  $\varepsilon_{ij}$  se definen como,

$$\varepsilon_{ij} = \frac{1}{2} \left( \frac{\partial d_i}{\partial x_j} + \frac{\partial d_j}{\partial x_i} + \sum_{k=1}^e \frac{\partial d_k^2}{\partial x_i \partial x_j} \right) \quad (10)$$

La figura 14 describe las relaciones descritas antes para un sistema 2D.

El tensor de tensión se compone de nueve parámetros en un sistema de 3 coordenadas:

$$\varepsilon = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{21} & \varepsilon_{31} \\ \varepsilon_{12} & \varepsilon_{22} & \varepsilon_{32} \\ \varepsilon_{13} & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix} \quad (11)$$

Donde esta matriz es simétrica con lo que solo se necesitan conocer seis componentes. Si los desplazamientos son pequeños, las componentes de segundo orden son despreciables y entonces se puede simplificar:

$$\varepsilon_{i,j} = \frac{1}{2} \left( \frac{\partial d_i}{\partial x_j} + \frac{\partial d_j}{\partial x_i} \right) \quad (12)$$

Los componentes del tensor de deformación con índices iguales representan la deformación en la dirección normal, mientras que las otras corresponden a la deformación cortante.

### 2.3.3 Ley de Hooke

No existe un modelo matemático simple que pueda describir el comportamiento mecánico de la amplia variedad de materiales existentes. Para que el modelo sea útil se deben hacer muchas simplificaciones. Bajo la asunción de que se aplica una compresión pequeña cuasi estática, los tejidos biológicos se pueden modelar como un sólido elástico lineal. Este modelo asume que el material es: sólido de una sola fase (no hay fluidos o porosidades en el interior), homogéneo (invariantes a posición en escala macroscópica), elástico (los materiales regresan a su forma y dimensiones previas a la deformación), y lineal (cada componente de tensión depende linealmente de los componentes de deformación). Bajo estas condiciones puede establecerse una relación lineal entre la tensión y la deformación dada por la ley de Hooke y que puede escribirse como:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix} \quad (13)$$

Los coeficientes  $\lambda$  y  $\mu$  se conocen como las constantes de Lamé.  $\lambda$  es el coeficiente de elasticidad de compresión y  $\mu$  es al coeficiente de cizallamiento (o de corte).

En ingeniería se usan comúnmente dos constantes relacionadas que representan propiedades de elasticidad: el módulo de Young  $E$  y la razón de Poisson  $\nu$ . El módulo de Young representa la razón entre la tensión aplicada y la deformación a

lo largo del eje de la tensión aplicada bajo una carga de estiramiento de una barra. La razón de Poisson es la razón entre la tensión lateral (perpendicular al eje de la tensión aplicada) y la deformación axial. Si la tensión se aplica a lo largo de  $x_1$ , se puede escribir:

$$E = \frac{\sigma_{11}}{\varepsilon_{11}} \quad (14)$$

y,

$$\nu = -\frac{\varepsilon_{22}}{\varepsilon_{11}} = -\frac{\varepsilon_{33}}{\varepsilon_{11}} \quad (15)$$

El módulo de Young se expresa en Pascals. La razón de Poisson es adimensional. Los materiales rígidos exhiben grandes módulos de Young, mientras que los materiales blandos poseen un módulo de Young pequeño. Con algunas excepciones en materiales compuestos, la razón de Poisson está entre 0 y 0.5. Materiales con razón de Poisson 0.5 son incompresibles, cualquier tensión axial induce deformaciones laterales para mantener el volumen constante. Un material con razón de Poisson 0 es totalmente comprensible; ante una compresión o expansión axial, no se deforma lateralmente pero cambia su volumen.

### 3 ALGORITMOS PARA EL CÁLCULO DE LA DEFORMACIÓN

En este capítulo se describen los algoritmos que se usaron para la obtención y el análisis de imágenes de elastografía.

#### 3.1 Algoritmos de estimación de desplazamientos

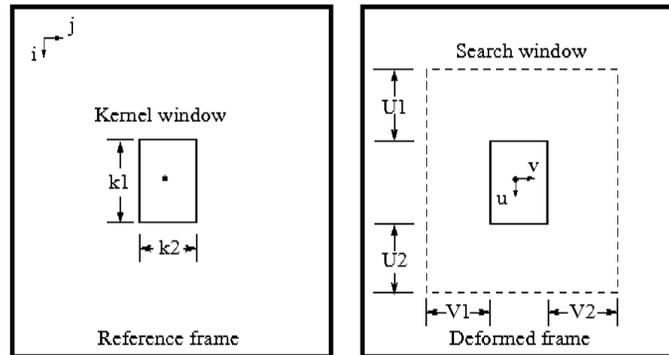
En este proyecto se trabaja con tres algoritmos para la obtención de imágenes de deformación escogidos de la literatura, a continuación es mostrada una definición de cada uno.

##### 3.1.1 Búsqueda de bloques exhaustiva

El algoritmo básico de búsqueda de bloques por suma del cuadrado de las diferencias (SSD) entre marcos de RF antes y después de la compresión, se describe en [50] para un kernel rectangular sobre una región de búsqueda como sigue:

$$SSD(u, v) = \sum_{m=-\frac{k_1-1}{2}}^{\frac{k_1-1}{2}} \sum_{n=-\frac{k_2-1}{2}}^{\frac{k_2-1}{2}} [r_1(i + m, j + n) - r_2(i + m + u, j + n + v)]^2 \quad (16)$$

Donde  $r_1$  y  $r_2$  son los campos de ecos RF de pre y post compresión respectivamente.  $I, J$  son los índices de muestreo axial y lateral para la ubicación donde se está estimando el desplazamiento;  $u$  y  $v$  definen las posiciones de búsqueda en la región de búsqueda; y  $K_1$  y  $K_2$  son el alto y el ancho del kernel respectivamente. Un aspecto muy importante es la obtención de valores iniciales para los tamaños de la región de búsqueda y kernel, con algún criterio razonable. En [50] se describe: El tamaño del kernel se elige empíricamente como 11 muestras en la dirección axial y 5 A-líneas ( $K_1 = 11$ ,  $K_2 = 5$ ) para un transductor de 7,5 MHz, o aproximadamente la mitad del área de la función de dispersión del punto del pulso de eco.



**Figura 15. Definición de Kernel y región de búsqueda.**

Una ilustración del kernel y región de búsqueda se muestra en la figura 15. Para cada ubicación  $(i, j)$  en que se estima el desplazamiento, se calcula la función SSD para cada posición muestra del marco RF que está dentro de la región de búsqueda (rango de centros de kernel) definido por  $-U1 \leq u \leq U2$  y  $-V1 \leq v \leq V2$ , donde  $U1$ ,  $U2$ ,  $V1$  y  $V2$  representan distancias de búsqueda hacia arriba, abajo, izquierda y derecha, respectivamente, como se muestra en la figura 2.3. El alto y ancho de la región de búsqueda son  $U = U1+U2+1$  y  $V = V1+V2+1$  respectivamente. La distribución de los desplazamientos usualmente no necesita ser tan fina como las muestras RF. Se usan  $k$  y  $l$  como índices de estimados de desplazamiento axial y lateral. La ubicación  $(umin, vmin) = (di(k, l), dj(k, l))$  donde se encuentra el mínimo SSD, se considera como la posición desplazada del kernel. De aquí,  $di(k, l)$  y  $dj(k, l)$  son los estimados del desplazamiento axial y lateral, respectivamente.

El costo computacional del algoritmo de búsqueda del bloque produce un estimado del desplazamiento que está determinado principalmente por los tamaños del kernel y la región de búsqueda. De hecho, para estimar un vector del desplazamiento, la operación SSD de la ecuación descrita anteriormente necesita hacerse  $K1K2UV$  veces. Los valores calculados SSD se comparan  $UV$  veces para encontrar el mínimo. Ya que el tamaño del kernel usualmente esta predefinido y fijo, la tarea de reducir el costo computacional del algoritmo de búsqueda de bloque es encontrar una forma de minimizar  $U$  y  $V$ .

### 3.1.2 Estimación de desplazamiento híbrido

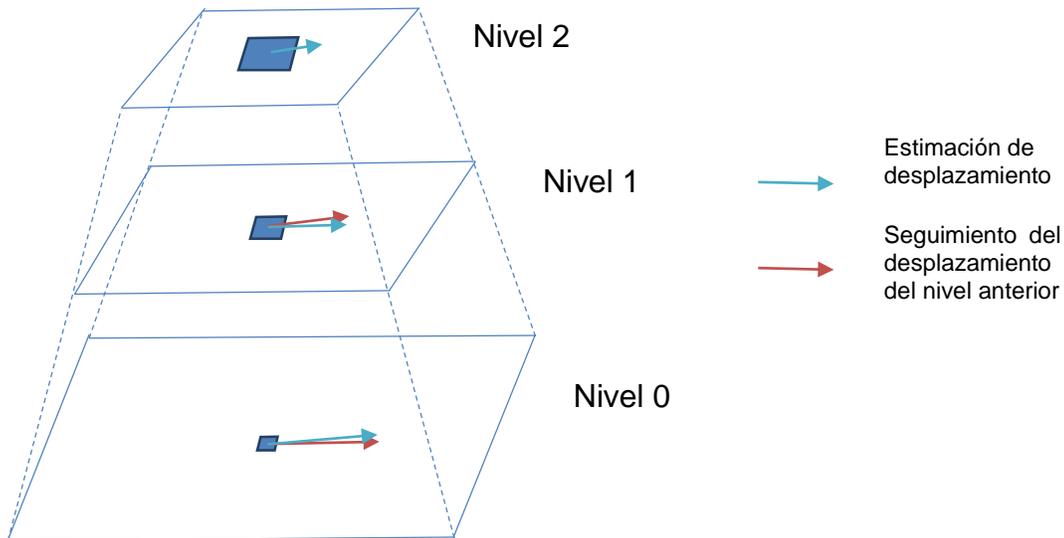
El método híbrido está constituido por varios algoritmos que cumplen funciones determinadas en pro de la obtención de estimaciones de desplazamiento robustas y precisas.

### 3.1.2.1 Multinivel

A diferencia de métodos que trabajan bajo la suposición de que el tejido será continuo [50], este enfoque tiene como propósito ser capaz de reconstruir imágenes de deformación en medios discontinuos, como en el caso de la arteria carótida o de las paredes adyacentes deslizantes en el seno.

Las estrategia de multinivel tal como se muestra en la Figura 16 se compone de una pirámide con varias etapas o niveles de procesamiento, donde las imágenes de referencia y de deformación son tratadas en una resolución gruesa en el nivel más alto de la pirámide, hasta llegar a una resolución fina en el nivel más bajo de la pirámide. En cada etapa se estiman los desplazamientos de ventanas distribuidas en la imagen de pre-deformación utilizando una estrategia de búsqueda, si se quiere, diferente, de una etapa a la otra, como en el método híbrido [11].

La estrategia de multinivel permite obtener semillas de resolución gruesa que representen estimaciones de diferentes regiones de la imagen, luego, sea hace un seguimiento de dichas estimaciones para encontrar los desplazamientos en la resolución más fina, de esta forma se puede conseguir resultados más precisos y robustos ante señales poco correlacionadas y regiones quebradas o discontinuas, en la Figura 16 se ilustra el algoritmo multinivel.



**Figura 16. Representación esquemática del seguimiento del desplazamiento del algoritmo multinivel. Modificada de: [11].**

### 3.1.2.2 Correlación cruzada

Para que un algoritmo de correspondencia de bloques como el que se muestra en la Figura 17, encuentre una ventana de la imagen de pre-deformación en una imagen de post-deformación, es necesaria una métrica de similitud que determine de entre un grupo de ventanas candidatas cual se asemeja más a la ventana buscada.

Se han encontrado gran cantidad de algoritmos que miden la similitud entre dos ventanas, entre las más destacadas de la elastografía se encuentran las métricas de correlación cruzada, suma de las diferencias de los cuadrados, fase de correlación y separación de fases.

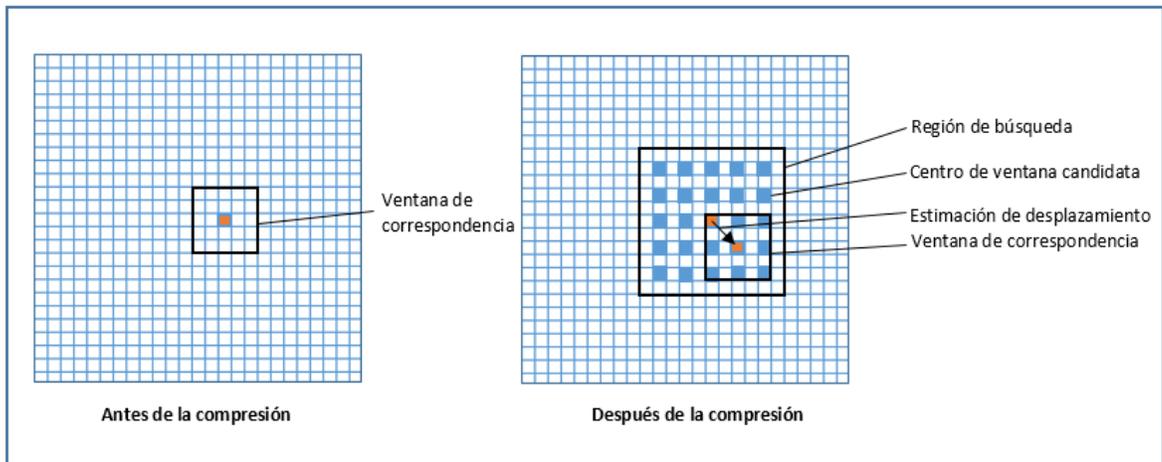
El método híbrido usa de forma combinada un algoritmo de separación de fases y de correlación cruzada en los diferentes niveles del algoritmo multinivel. En los niveles más altos, en donde las ventanas son más grandes, es implementada la métrica de correlación cruzada de [11] tal como se muestra a continuación:

$$CC(i, j) = \frac{\sum_{x=1}^M \sum_{y=1}^N [I_0(x, y) - \bar{I}_0][I_1(x + i + i_0, y + j + j_0) - \bar{I}_1]}{\sqrt{\sum_{x=1}^M \sum_{y=1}^N [I_0(x, y) - \bar{I}_0]^2 \times \sum_{x=1}^M \sum_{y=1}^N [I_1(x + i + i_0, y + j + j_0) - \bar{I}_1]^2}} \quad (17)$$

Donde  $I_0$  y  $I_1$  representan las imágenes de pre-compresión y post-compresión,  $i$  y  $j$  denota el desplazamiento relativo entre las imágenes  $I_0$  y  $I_1$ ,  $i_0$  y  $j_0$  son los cambios de desplazamiento relativo de las ventanas vecinas o de las ventanas que se encuentran en un nivel superior,  $M$  y  $N$  son los tamaños de las ventanas, donde  $\bar{I}_0$  y  $\bar{I}_1$  denotan los valores medios de las imágenes  $I_0$  y  $I_1$ .

En el nivel más bajo del algoritmo multinivel en donde las ventanas son muy pequeñas el método híbrido implementa una métrica de separación de fase llamada separación de fases ponderada (WPS).

En este proyecto se decidió tomar la métrica de correlación cruzada para todos los niveles del algoritmo.



**Figura 17. Representación esquemática de una correspondencia de bloques general.**

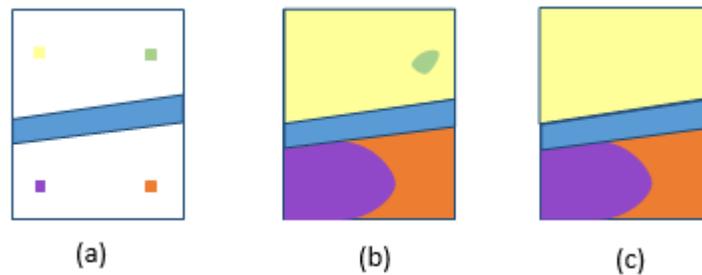
### 3.1.2.3 Seguimiento de la calidad

Las estrategias de seguimiento utilizan estimaciones de desplazamiento vecinas o de un nivel superior para inicializar la búsqueda de una ventana, es decir, que el punto desde donde se inicia la búsqueda de la ventana, corresponde al desplazamiento estimado de una ventana vecina o una ventana que se encuentra en la misma posición, pero en un nivel superior. Las estrategias de seguimiento son varias [19, 13]. El método híbrido implementa el algoritmo de seguimiento guiado por la calidad [19].

El algoritmo de seguimiento guiado por la calidad a diferencia de otros métodos que restringen el seguimiento en una dirección específica, ya sea de izquierda a derecha, de arriba abajo, entre otros, realiza un seguimiento basado en la calidad de sus vecinos, es decir que una ventana inicia su búsqueda partiendo del desplazamiento hallado por el vecino con la mejor exactitud en su estimación.

La exactitud está representada por el coeficiente de similitud, el método híbrido utiliza el algoritmo de Correlación Cruzada Normalizada para encontrar los coeficientes, por lo que los valores cercanos a 1 corresponden a los coeficientes de similitud con mayor exactitud o calidad.

En la Figura 18 es mostrada en una ilustración las características del algoritmo de seguimiento guiado por la calidad modificada de [19].



**Figura 18. Ilustración del algoritmo de seguimiento guiado por la calidad**

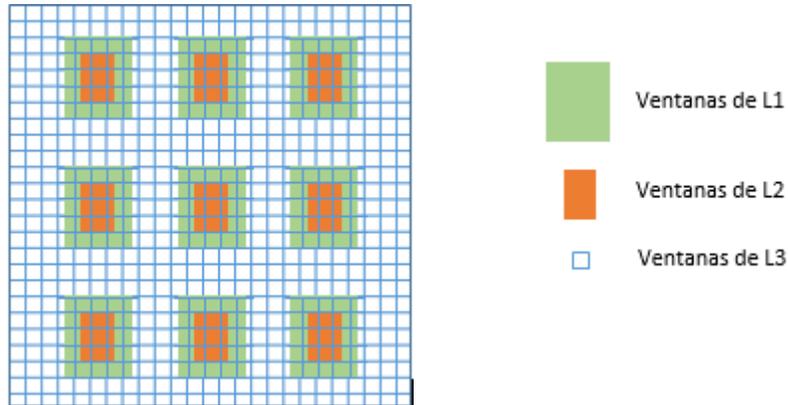
La imagen contiene datos pobremente correlacionados donde no se podrá estimar desplazamientos correctos, representados por una franja azul, que separa los buenos datos de la imagen en dos. (a) a manera de ejemplo son mostradas solo cuatro semillas distribuidas en toda la imagen. (b) todas las semillas propagan sus propiedades o valores a su alrededor, pero la semilla de color verde no lo hace muy bien, esto indica que posee un valor incorrecto de desplazamiento que provoca malas estimaciones de las ventanas vecinas que terminan rápidamente por limitar la propagación de la semilla. (c) Las regiones propagadas por las malas semillas son detectadas por su tamaño reducido y son remplazadas por valores estimados a partir de las semillas vecinas de buena calidad. Al final se reconstruyen todas las regiones de la imagen a pesar de los datos poco correlacionados.

### 3.1.2.4 Multi-resolución

Existen varios algoritmos de búsqueda de ventanas, el más básico es el algoritmo de búsqueda exhaustiva, pero no es ni eficiente ni rápido. En el método híbrido se ha escogido un algoritmo de multi-resolución por su eficiencia y costo computacional bajo.

Multi-resolución es una estrategia de búsqueda rápida de ventanas. Lo que se quiere con esta estrategia es encontrar la ubicación de una ventana en tres pasos. En el primer paso las ventanas tienen resolución pobre, su correspondencia es buscada en la imagen de post-deformación en una región de búsqueda y con un salto de muestra predeterminados. En el segundo paso se toma la estimación anterior para inicializar la búsqueda de la ventana, pero esta vez, con una resolución mayor y una región de búsqueda y salto de muestra menor. Para el tercer paso se repite el mismo proceso.

### 3.1.2.5 Descripción del método Híbrido



**Figura 19. Ilustración del Método Híbrido de Estimación de Desplazamiento.**

El método híbrido es descrito a continuación de acuerdo a como se explica en [10], tomando algunas pequeñas derivaciones basadas en los siguientes trabajos [16] y [18].

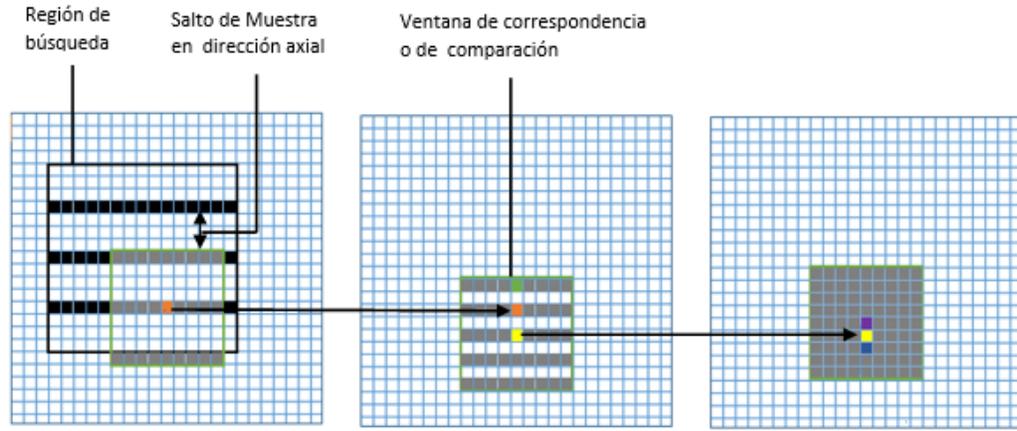
De acuerdo a [16] se construye una pirámide de tres niveles, que van desde L0 hasta L2. En el primer nivel de procesamiento L0, las imágenes de pre-deformación y post-deformación que se utilizan para la estimación de desplazamientos son matrices de datos RF. En L1 el procesamiento se hace con sub-muestras de imágenes en modo B (envolvente de datos RF) de pre-deformación y post-deformación construidas a partir de las matrices de datos RF. En L2 el procesamiento se hace con sub-muestras de las imágenes en modo B del nivel anterior.

En el L1 y L2 se evita utilizar de forma directa sub-muestras de las matrices de datos RF para no violar el criterio de muestreo de Nyquist y por qué la suavidad de las imágenes en modo B son preferibles en el momento de hacer el sub-muestreo.

En L2 son distribuidas ventanas o bloques de gran tamaño, separados unas de otras en la imagen de pre-deformación tal como se muestra en la Figura 19, el ancho de cada ventana es controlado tomando como base que a lo ancho de imagen deben distribirse solo tres ventanas, el largo de cada ventana es definida dependiendo de qué tan quebrada o discontinuo sea el tejido.

La ubicación de estas ventanas se estiman con una búsqueda en la imagen de post-deformación haciendo una comparación de bloques con una métrica de correlación cruzada.

Para buscar cada ventana en L2 no se utiliza ningún tipo de estrategia de seguimiento, solo se implementa una estrategia de multi-resolución rápida que encuentra la estimación de los desplazamientos en tres pasos. En la Figura 20 es mostrada una representación detallada del algoritmo.



**Figura 20. Ilustración de algoritmo multi-resolución modificado de: [11]**

La primera figura corresponde al primer paso del algoritmo, La ubicación del botón naranja representa la mejor estimación del desplazamiento axial y lateral. En el segundo paso solo habrá tres ventanas candidatas, naranja, amarillo y verde. El botón amarillo es la ubicación de la mejor estimación de desplazamiento. En el tercer paso las ventanas candidatas también son solo tres, amarillas, azules y moradas. El botón morado es la ubicación de la mejor estimación de desplazamiento. En el primer paso la resolución de la ventana ira de acuerdo al valor de  $k$ , en nuestro caso  $k=4$ , entonces, la ventana se compone de datos tomados cada cuatro muestras en dirección axial (botones grises) y partiendo desde el centro como se muestra en a) de la Figura 19. Como  $k=4$  se toman ventanas candidatas (botones negros son los centros de las ventanas candidatas) cada cuatro muestras en dirección axial en una región de búsqueda y se comparan con la ventana buscada a través de una métrica de correlación cruzada, la ubicación de la ventana con mejor coeficiente de correlación es registrada. La región de búsqueda es definida de acuerdo a la compresión máxima estimada, aplicada para deformar el tejido.

En el segundo paso  $k=2$ , por lo que la resolución de la ventana aumenta como se muestra en b), pero a diferencia del paso anterior, solo habrá tres ventanas candidatas, independientemente de la región de búsqueda, la primera ventana corresponde a la misma posición de la ubicación o desplazamiento estimado registrado en el paso anterior, la segunda y tercera ventana estarán ubicadas arriba y abajo, en salto de dos de acuerdo al valor de  $k$ . se hace la comparación de ventanas como en el paso anterior y se registra el nuevo valor de desplazamiento.

En el tercer paso se repite el mismo proceso del paso anterior pero con un valor de  $k=1$ , y el valor hallado será el resultado final del algoritmo de multi-resolución rápida, definido como el desplazamiento estimado axial y lateral.

Hay estimaciones de desplazamiento que pueden resultar erróneas o de mala calidad, identificadas por un coeficiente de similitud pobre. Las estimaciones de desplazamiento cuyo coeficiente de similitud se encuentren por debajo de un umbral serán rechazadas y remplazadas por estimaciones de desplazamiento vecinas. El valor del umbral dependerá de la correlación de los datos y de la discontinuidad del tejido.

Después de terminar la estimación de los desplazamientos de todas las ventanas en nivel 2, pasamos al nivel 1.

En L1 las ventanas son de un tamaño y separación menor con respecto a L2, parte de estas ventanas estarán centradas y alineadas con las ventanas de L2 como se muestra en la Figura 19. Para buscar cada ventana en L1 aparte de usar una estrategia de multi-resolución rápida, es utilizado el algoritmo de Seguimiento Guiado por la Calidad.

El algoritmo de Seguimiento Guiado por la Calidad en base a [19], puede ser implementado con una semilla o múltiples semillas. En este trabajo se utilizan múltiples semillas. Los desplazamientos estimados de L2 son las semillas que se usan en L1, estas semillas necesitan ser interpoladas en las ventanas de L1, para eso se estiman los desplazamientos de cada una de las ventanas en L1 centradas en las ventanas de L2 tomando como punto de partida de búsqueda los desplazamientos estimados de L2.

Luego de tener las semillas interpoladas, son registradas en una tabla llamada Ranking, aquí se guardan los coeficientes de similitud, los desplazamientos y la ubicación de la ventana buscada. Después se estiman los desplazamientos de las

ventanas faltantes de L1. Esta estimación se hace con ayuda del algoritmo guiado por la calidad explicado a continuación.

Del Ranking es tomado el desplazamiento lateral y axial estimado de la ventana con mejor coeficiente de similitud, es decir, de mejor calidad, para que sea un punto de inicio de búsqueda de las ventanas vecinas más próximas. Luego de estimar los desplazamientos de estas ventanas vecinas, son registradas en el Ranking, que se actualiza eliminando los datos de la ventana que ya se ha utilizado. Del Ranking es tomado de nuevo la ventana de mejor calidad y se repite el mismo proceso anterior hasta haber estimado los desplazamientos de todas las ventanas.

Las estimaciones de desplazamiento erróneas son rechazadas por medio de un umbral que puede ser diferente del de L2 y serán remplazadas o interpoladas por datos vecinos, pero este proceso a diferencia del de L1 solo será ejecutado cuando a todas las ventanas se les haya estimado el desplazamiento. El dato erróneo se remplazara por el vecino con mejor coeficiente de similitud.

Una vez terminado la interpolación de todos los datos, se desciende al nivel L0.

En L0 son distribuidas ventanas pequeñas si ningún tipo de separación tal como se muestra en la Figura 19. En este nivel se repiten los mismo pasos del nivel anterior pero añadiendo un proceso de corrección al algoritmo de Seguimiento de calidad.

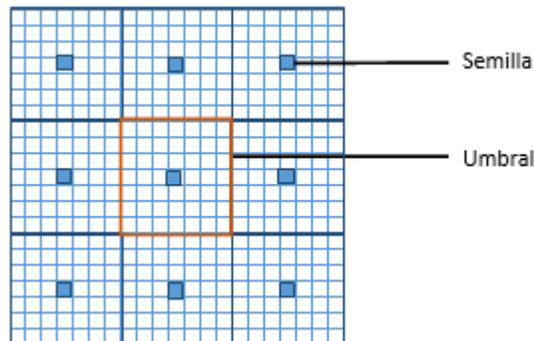
La corrección permite, como se ve en c) de la Figura 18, eliminar regiones de estimaciones de desplazamiento erróneas debido a malas semillas y remplazarlas por estimaciones correctas.

Para hacer la corrección es usado un vector llamado ID-semilla, en este vector se llena el número de veces que el desplazamiento estimado de una semilla ha sido utilizado de forma directa o indirecta para estimar el desplazamiento de otras ventanas.

La ID-semilla se recorre para buscar las malas semillas. Una semilla será mala, si el número de sus propagaciones está por debajo de un umbral. Después las ubicaciones de las malas semillas y de sus propagaciones son detectadas y eliminadas.

Las ventanas con estimaciones de buena calidad que limitan con las regiones eliminadas son identificadas y registradas en un Ranking nuevo. A partir de los valores llenados en el Ranking se vuelven a estimar los desplazamientos de las regiones o ventanas que se eliminaron por sus estimaciones erróneas.

El umbral será escogido según el número de semillas que se utilicen, es decir según el número de ventanas en L1, en la Figura 21 se muestra un ejemplo de cómo escoger el umbral.



**Figura 21. Ilustración del mínimo número de propagaciones que hacen que una semilla sea buena.**

En este ejemplo la imagen contiene 9 semillas representadas en color azul, distribuidas en la matriz de estimaciones de desplazamiento de tamaño 21x21. El marco del color naranja contiene el número de ventanas mínimas en las que se debe propagar una semilla, para que sea considerada como buena. En este caso el umbral tendría un valor de 49.

### 3.1.3 AM2D

El método de elastografía en tiempo real AM2D [1] se basa en la minimización analítica (AM) de las funciones de costo que incorporan similitud de la intensidad de los datos RF y continuidad de desplazamiento que lo hace robusto a de-correlaciones pequeñas. Para que el método sea resistente a de-correlaciones grandes son usadas algunas técnicas de estadística. El método no está basado en ventanas, por lo que la estimación del desplazamiento se hace para cada uno de las muestras de los datos-RF. También, un filtro de Kalman es usado para la obtención de las imágenes de deformación a partir de los campos de desplazamiento.

### 3.2 Calculo del Gradiente

Las imágenes de deformación permiten analizar el grado de rigidez o de deformación de un tejido blando, una forma de obtener estas imágenes consiste en conseguir el gradiente de las estimaciones de desplazamiento, como se muestra a continuación:

$$\varepsilon_{i,i}(x_i, x_j) = \left( \frac{\partial d_i(x_i, x_j)}{\partial x_i} \right) \text{Coeficiente de deformación axial} \quad (18)$$

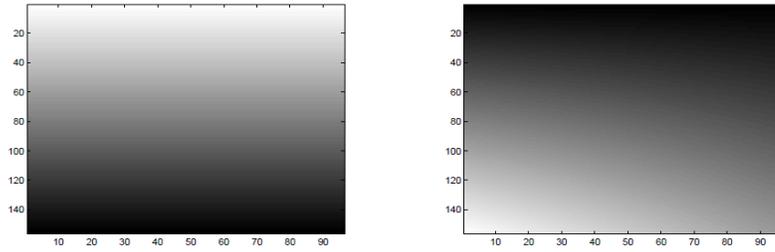
$$\varepsilon_{j,j}(x_i, x_j) = \left( \frac{\partial d_j(x_i, x_j)}{\partial x_j} \right) \text{Coeficiente de deformación lateral} \quad (19)$$

Existen diferentes técnicas para obtener el gradiente de las imágenes de desplazamiento, en [51] comparan dos técnicas, que muestran resultados similares y buenos. Una de ellas solo requiere una derivada simple, seguido de un filtro Gaussiano. Esta técnica es usada en este proyecto.

En [1] es desarrollado un algoritmo para el cálculo de deformación a partir de la imagen de desplazamiento usando un filtro de Kalman. Este algoritmo es utilizado para comparar el desempeño de la técnica que usa filtro Gaussiano.

### 3.3 Normalización

La normalización hace referencia a encontrar una escala apropiada para los coeficientes de deformación en cada imagen de modo que la información no sea ambigua. La persistencia hace referencia a que en tiempo real, en cada nuevo marco de datos RF que origina nuevos valores de deformación, exista persistencia visual en las regiones donde no hay deformaciones, ya sea mediante procesos de promediado en el tiempo o mediante procesos de promediado espacial. El despliegue hace referencia a que, de forma natural, en las imágenes de ultrasonido las regiones oscuras corresponden a zonas donde la intensidad de las señales es baja y no hay información, mientras que las regiones claras corresponden a zonas donde la intensidad de señal es alta y es donde hay mayor información.



**Figura 22. Planos de normalización. Izquierda: Plano según (2.2). Derecha: Plano según (2.4). Blanco para deformaciones grandes, negro para deformaciones pequeñas.**

El módulo despliegue busca hacer un balance entre luminosidad o intensidad de la imagen, que debe ser más intensa donde hay información de precisión (cambios de deformación), y cambios en el color que indican la deformación. En este trabajo, se exploró el proceso de normalización descrito en [52], el cual es una solución particularizada de [53]. Allí se plantea que un método robusto para obtener una escala apropiada para las deformaciones (axiales) en cada imagen consiste en ajustar un plano  $\hat{s}(x, y)$  al conjunto total de los desplazamientos en la forma:  $d(x, y) \approx \hat{d}(x, y) = \alpha + \hat{s}y$ , donde  $x$  es el eje de la dirección lateral y el eje en dirección axial. De este modo, al encontrar el coeficiente de deformación axial

$$\varepsilon_{yy}(x, y) = \frac{\partial d(x, y)}{\partial y} \quad (20)$$

Con  $d(x, y)$  los desplazamientos en la dirección axial; si se aplica el mismo gradiente a  $\hat{d}(x, y)$  se obtiene

$$\frac{\partial \hat{d}(x, y)}{\partial y} = \hat{s} \frac{\partial \hat{s}(x, y)}{\partial y} \quad (21)$$

Entonces, el coeficiente de deformación axial debería ser un múltiplo del coeficiente de deformación “promedio”  $\hat{s}$ . El coeficiente normal en la dirección axial entonces se calcula como:

$$\varepsilon'_{yy}(x, y) = \begin{cases} \varepsilon_{yy}(x, y) / \hat{s}(x, y) & \hat{s}(x, y) \neq 0 \\ 1 & \hat{s}(x, y) = 0 \end{cases} \quad (22)$$

Los parámetros de ajuste del plano se encuentran por algún algoritmo de regresión de mínimos cuadrados. El plano  $\hat{s}$  está predefinido dependiendo del tipo de problemas que se esperan encontrar en el cálculo de las deformaciones. En [53] se exponen dos tipos de planos y quizás algunas variaciones de las ecuaciones de los desplazamientos así:

Para ajustar para tensión de deformación reducida en profundidades grandes lejos de la prueba, ya que la tensión se esparce en el tejido circundante,

$$\begin{aligned}\hat{d}(x, y) &= \alpha_0 + \alpha_1 x + y(\beta_1 + \beta_2 y) \\ \hat{s}(x, y) &= \beta_1 + 2\beta_2 y\end{aligned}\tag{23}$$

Para ajustar para tener en cuenta que la prueba de ultrasonido puede rotar en el eje de elevación durante la exploración,

$$\begin{aligned}\hat{d}(x, y) &= \alpha_0 + \alpha_1 x + \beta_1 y(1 + \beta_2 y)(1 + \beta_3 x) \\ \hat{s}(x, y) &= \beta_1(1 + 2\beta_2 y)(1 + \beta_3 x)\end{aligned}\tag{24}$$

## 4 IMPLEMENTACIÓN Y DESARROLLO SOFTWARE

En este capítulo se realiza una descripción del diseño e implementación de los algoritmos asociados al software Elax1, además de las herramientas software utilizadas en el proceso de desarrollo. La herramienta permite mostrar elastogramas a partir del cálculo de la deformación de los tejidos. Para ello tiene la opción de escoger entre tres algoritmos, mencionados en el capítulo 3.

### 4.1 Herramientas de desarrollo software utilizadas

Esta descripción es tomada de las páginas web asociadas a las herramientas que se mencionan a continuación.

#### 4.1.1 QT [54]

Es un framework<sup>13</sup> para desarrolladores, que se usa para hacer aplicaciones con GUI<sup>14</sup>. Esta biblioteca incluye un conjunto de widgets (“controles” en la terminología de Windows) encargados de proporcionar las funciones estándar de una interfaz o GUI. QT presenta una alternativa para la comunicación inter-objeto denominada “signals y slots” que reemplaza al antiguo sistema de callbacks utilizado por otros sistemas, además de proporcionar un modelo de eventos convencional para la captura de las pulsaciones del ratón, teclas u otras entradas del usuario haciendo posible la creación de aplicaciones multiplataforma mediante funciones estándar, incluyendo drivers para Linux, Windows, e incluso drivers nativos para bases de datos como Oracle®, Microsoft® SQL Server, IBM DB2®, Borland.

##### 4.1.1.1 Principios de Qt (Signals and slots)

“Señales y ranuras” es un mecanismo desarrollado para la comunicación inter-objeto con el fin de reemplazar a los conocidos callbacks. Este sistema es flexible, totalmente orientado a objetos esta implementado en C++.

Una señal es emitida por un objeto cuando cambia su estado de alguna forma. Sólo aquellas clases que definan una señal y las subclases que deriven de ésta pueden emitir la señal. Cuando se emite una señal, el slot conectado a ella es ejecutado de forma inmediata como si se tratara de una función de llamada

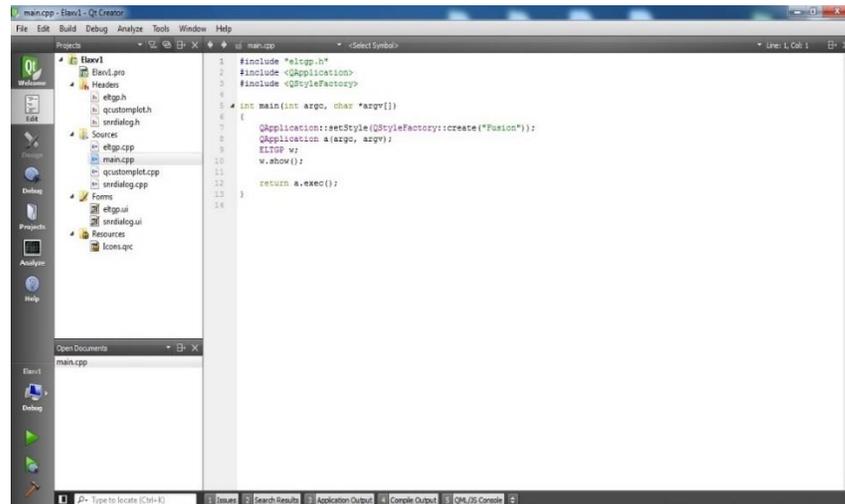
---

<sup>13</sup>Framework: herramienta software para desarrolladores.

<sup>14</sup>GUI: *Graphical User Interface* (Interfaz Gráfica de Usuario).

normal. Si varios slots están conectados a una misma señal todos ellos serán ejecutados uno después de otro pero en un orden arbitrario.

Los slots son funciones en C++ y por lo tanto también podrán ser llamadas de forma normal. Su única característica especial es que las señales pueden ser conectadas a una señal. En comparación con los callbacks, los signals y slots son mucho más sencillos y rápidos de utilizar ya que la diferencia con las aplicaciones reales es insignificante. La emisión de una señal es de cuatro a diez veces más lenta que trabajar directamente con los callbacks debido al tiempo requerido para localizar el objeto, pero su velocidad relativa aumenta al necesitar muchas menos operaciones de tipo new y delete. Además QT trabaja con un conjunto de widgets que poseen funciones típicas para el diseño de interfaces de usuario, los widget son capaces de recibir los eventos del teclado o ratón junto a otros eventos del sistema [21].



**Figura 23. Interfaz de desarrollo Qt**

#### 4.1.1.2 Qt Designer

Es un diseñador de interfaces gráficas para aplicaciones basados en las bibliotecas QT. Es posible escribir aplicaciones completamente en código fuente, o utilizando Qt Designer para un desarrollo más rápido. La arquitectura basada en componentes hace posible a los desarrolladores ampliar Qt Designer con widgets personalizados y extensiones. Qt Designer presenta el inconveniente de no poderse programar directamente sobre él y tener que recurrir a entornos de desarrollo para poder conectar los eventos del computador con otras aplicaciones.

Diseñar un archivo con Qt Designer es un proceso simple, consiste en arrastrar los ítems desde un menú a un formulario, utilizando herramientas de edición para seleccionar, cortar, pegar y cambiar el tamaño de los diferentes componentes. A la salida de este programa se obtiene un fichero con la extensión .ui que contiene una descripción de la interfaz indicando sus características, componentes y forma de comportamiento.

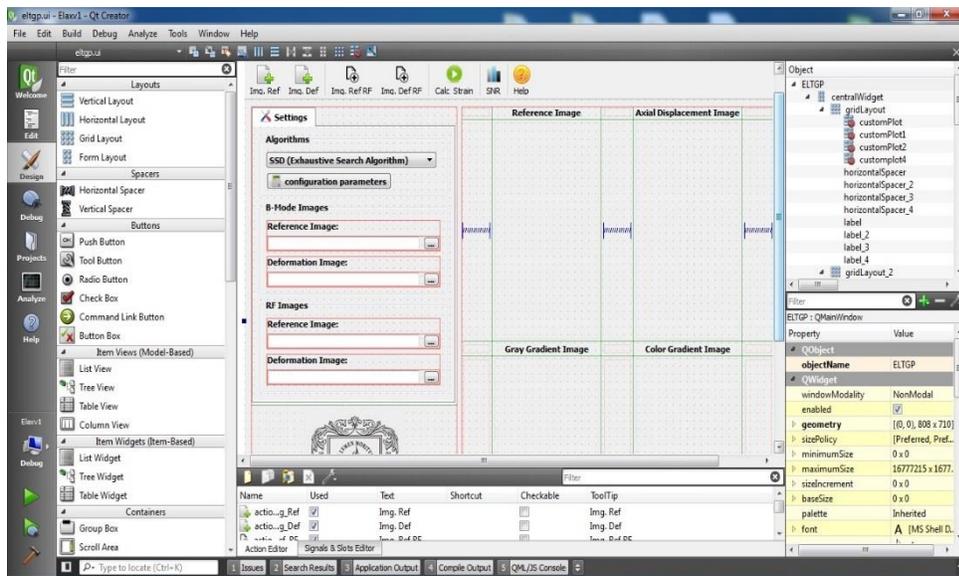


Figura 24. Interfaz de Qt Designer.

#### 4.1.2 OpenCV [55]

OpenCV es un código abierto de C ++ para procesamiento de imágenes y visión por ordenador, desarrollado originalmente por Intel y ahora apoyado por Willow Garage.

Es gratuito tanto para uso comercial y no comercial. Por lo tanto no es obligatorio para sus aplicaciones de OpenCV sean abiertos o libres.

Es una biblioteca de muchas funciones incorporadas dirigidas principalmente a procesamiento de imágenes en tiempo real. Ahora tiene varios cientos de algoritmos de procesamiento de imágenes y visión por ordenador que hacen el desarrollo de aplicaciones avanzadas de visión por ordenador fácil y eficiente.

#### Características principales

- Optimizado para tiempo real de procesamiento de imágenes y aplicaciones de visión de computadora
- Interfaz principal de OpenCV es en C ++
- También hay interfaces de plenos C, Python y Java
- Aplicaciones OpenCV se ejecutan en Windows, Android, Linux, Mac y iOS
- Optimizado para procesadores de Intel

## **Módulos OpenCV**

OpenCV tiene una estructura modular. Los principales módulos de OpenCV se enumeran a continuación.

### **Core**

Este es el módulo básico de OpenCV. Incluye estructuras básicas de datos (estructura de datos Mat EG-) y las funciones básicas de procesamiento de imágenes. Este módulo también se utiliza ampliamente por otros módulos como highgui, etc.

### **highgui**

Este módulo proporciona herramientas sencillas de interfaz de usuario, varios codecs de imagen y de vídeo, herramientas de captura de video, la manipulación de las ventanas de imagen, etc. Si desea funciones más avanzadas de interfaz de usuario, usted tiene que utilizar los marcos de la interfaz de usuario como Qt, WinForms, etc.

### **imgproc**

Este módulo incluye algoritmos de procesamiento de imágenes básicas, incluyendo el filtrado de imágenes, transformaciones de imagen, las conversiones de espacio de color y etc.

### **Vídeo**

Este es un módulo de análisis de vídeo que incluye los algoritmos de seguimiento de objetos, algoritmos de sustracción de fondo y etc.

## **Objdetect**

Esto incluye los algoritmos de detección y de reconocimiento de objetos para objetos estándar.

OpenCV ahora se utiliza ampliamente para el desarrollo de aplicaciones avanzadas de procesamiento de imágenes y visión por computador. Ha sido una herramienta para estudiantes, ingenieros e investigadores en todos los rincones del mundo.

### **4.1.3 Matlab Compiler Runtime [56]**

Ejecute aplicaciones o componentes de MATLAB compilados sin instalar MATLAB.

MATLAB Runtime es un conjunto independiente de bibliotecas compartidas que permite la ejecución de aplicaciones o componentes de MATLAB compilados en ordenadores que no tienen instalado MATLAB. Cuando se utilizan conjuntamente, [MATLAB](#), [MATLAB Compiler](#) y MATLAB Runtime le permiten crear y distribuir aplicaciones numéricas o componentes de software de forma rápida y segura.

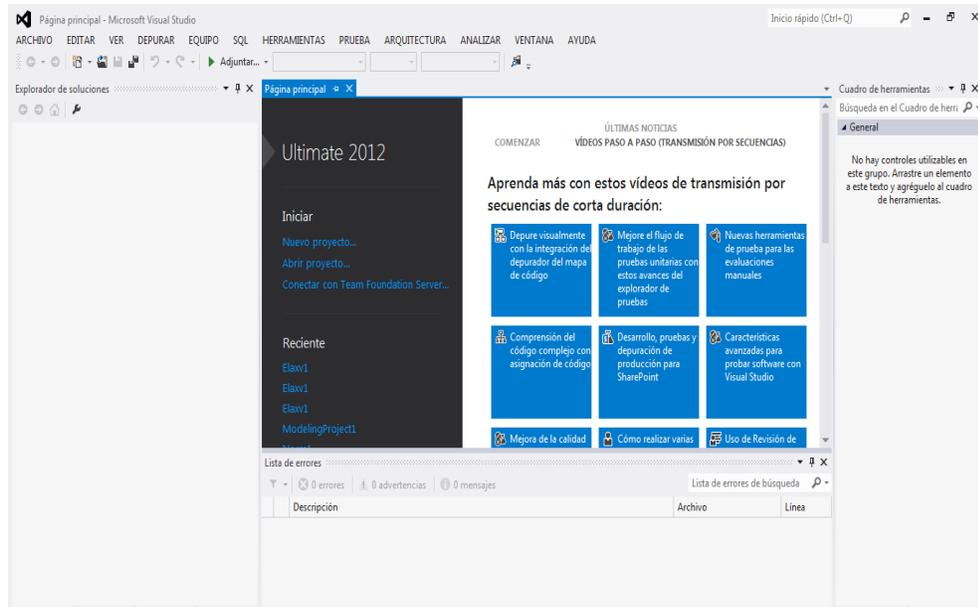
En caso de tener instalado MATLAB, no es necesaria la instalación del Runtime.

### **4.1.4 Microsoft Visual Studio [57]**

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones cliente y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

La versión usada es Microsoft Visual Studio 2012.



**Figura 25. Interfaz Visual Studio 2012.**

Para mayor información acerca del uso y configuración de las herramientas mencionadas anteriormente ver anexo XX.

## 4.2 Diseño e implementación.

Con el fin de diseñar de manera correcta la herramienta Elax1, se elaboró un diagrama de casos de uso, donde se definen los requisitos del usuario.

■

### **Figura 26. Diagrama de casos de uso Elax1**

La figura 26, muestra el diagrama de casos de uso para la herramienta desarrollada, donde se definen las acciones básicas operadas por el usuario y las desarrolladas por el sistema. Para entender la estructura y componentes de las clases que interactúan en cada caso, es necesario el uso de un diagrama de relaciones entre casos de uso.

La figura 27, muestra cómo se presentan las relaciones y la interacción del usuario en la herramienta.

▼

### **Figura 27. Diagrama de relaciones de casos de uso de Elax1**

#### **4.3 Diagrama de clases**

A continuación se hace una descripción del diagrama de clases en UML, donde se muestra la relación de las clases que se involucran en el sistema.

Las clases involucradas en el diagrama de la figura 28, son las que se han considerado como principales, pero estas a su vez están compuestas por otras sub-clases, las cuales no fueron incluidas en el diagrama principal pero se procederán a explicar más adelante.

### **Figura 28. Diagrama de clases de Elax1**

Para entender un poco lo que se realiza en cada clase, se explicara cómo funciona la relación entre ellas, de acuerdo al algoritmo que se esté utilizando en el momento de la ejecución de la herramienta.

Cabe resaltar que la clase gradiente, como la clase strain y NormA1 se utilizan en los tres algoritmos aquí mencionados, por lo tanto se describirá generalmente una sola vez en el método SSD.

## 4.4 Método SSD

En el diagrama de clases de la figura 29 se ilustra la relación de clases para el método SSD.

88

### Figura 29. Diagrama de clases del método SSD

Cada una de las clases y algunas partes importantes del código son explicadas a continuación.

#### 4.4.1 Strain

Esta clase se utiliza para crear los objetos y variables, que permiten fijar y mostrar las imágenes de referencia, de deformación, de desplazamientos axial y lateral y de deformaciones axial y lateral.

A continuación se incluye el código de cabecera de la clase strain.

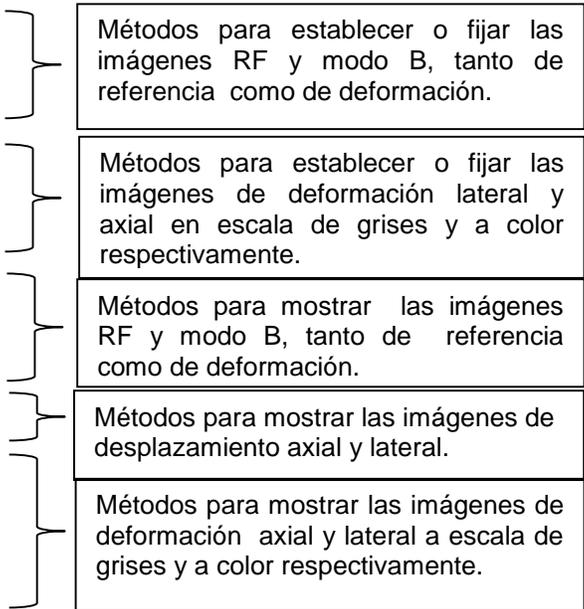
```

#ifndef __STRAIN_H__
#define __STRAIN_H__
#include "stdafx.h"
#include "opencv/cv.h"
using namespace cv;

class strain
{
private:
    Mat Ga; //imagen de gradiente axial
    Mat Gl; //imagen de gradiente lateral
    Mat GaRgb; //imagen a color de gradiente axial
    Mat GlRgb; //imagen a color de gradiente lateral
protected:
    IplImage* imgref; // Imagen de referencia
    IplImage* imgdef; // Imagen de deformación
    IplImage* imgref_RF; // Imagen de referencia RF
    IplImage* imgdef_RF; // Imagen de deformación RF
    int num_rows; // Número de filas de la imagen
    int num_cols; // Número de columnas de la imagen
public:
    // Imágenes de OpenCv para Desplazamientos
    Mat Da; // Imagen de desplazamiento axial
    Mat Dl; // Imagen de desplazamiento lateral

public:
    // Constructores
    strain(IplImage* imr, IplImage* imd, IplImage* imr_RF, IplImage* imd_RF);
    // SetThings
    void SetImgRef(IplImage* imr);
    void SetImgDef(IplImage* imd);
    void SetImgRef_RF(IplImage* imr_RF);
    void SetImgDef_RF(IplImage* imd_RF);
    void SetImgGa(Mat _Ga);
    void SetImgGl(Mat _Gl);
    void SetImgGaColor(Mat _GaRgb);
    void SetImgGlColor(Mat _GlRgb);
    // GetThings
    IplImage* GetImgRef(void);
    IplImage* GetImgDef(void);
    IplImage* GetImgRef_RF(void);
    IplImage* GetImgDef_RF(void);
    Mat GetImgDa(void);
    Mat GetImgDl(void);
    Mat GetImgGa(void);
    Mat GetImgGl(void);
    Mat GetImgGaColor(void);
    Mat GetImgGlColor(void);
    // Operaciones

```



```

        // Destructor
        ~strain(){};
};
#endif

```

#### 4.4.2 Gradiente

Esta clase obtiene imágenes de deformación aplicando un gradiente a las imágenes de desplazamiento encontradas previamente. En esta clase se hace uso de una “librería compartida” creada en matlab llamada **am2dlib.h**, que permite el cálculo del gradiente a través de una función de Kalman implementada en Matlab por medio de la función **Cpp\_Kal\_LSQ**. Para mayor información sobre la creación de las librerías compartidas en Matlab leer anexo C.

```
#include "am2dlib.h"
```

El llamado a esta clase es hecho desde la interfaz Elax1 como se muestra a continuación.

```

//calcular el gradiente
gradiente grd(str.Da,nrR,ncR);

```

El constructor de la clase **gradiente** recibe como entrada el desplazamiento axial y el tamaño de filas y columnas de la matriz de desplazamiento. Una vez llamado el constructor, este inicializa los parámetros y crea las variables necesarias para el cálculo del gradiente.

Adicionalmente el constructor inicializa la aplicación amd2lib necesaria para que la librería de matlab funcione con el **MCR**<sup>15</sup>, para su posterior uso.

```
amd2libInitialize() //inicializa la aplicación am2dlib
```

El método **CalcStrain**, invoca una función llamada **strains\_algorithm** la cual convierte las variables que se tienen a un formato que se pueda interpretar en matlab haciendo uso de las variables cargadas con anterioridad tal como se muestra a continuación.

---

<sup>15</sup> “MATLAB COMPILER RUNTIME, es un conjunto independiente de bibliotecas compartidas que permite la ejecución de aplicaciones o componentes de MATLAB compilados en ordenadores que no tienen instalado MATLAB”

```

mxDouble* IMGDA = om::allocateMatrixFromOpencv<mxDouble>(da);
om::copyDataToMatlab(da,IMGDA);
mwArray m_Da( num_rows, num_cols, mxDOUBLE_CLASS); // Imagen Desplazamiento Axial
que retorna AM2D, formato Matlab
m_Da.SetData(IMGDA, num_rows*num_cols); // Imagen desplazamiento MatLab
mwArray mstk((num_rows-2*rowgap+1),(num_cols-2*colgap), mxDOUBLE_CLASS); // Define
imagen strains con kalman MatLab
const mwArray wDiff(43);

```

La función **Cpp\_Kal\_LSQ**, es la encargada de calcular la imagen de deformación.

```

Cpp_Kal_LSQ(1, mstk, m_Da,wDiff); // Obtiene en mstk la imagen de strains en formato
Matlab
mstk.GetData(matlab_strainK,(num_rows-2*rowgap+1)*(num_cols-2*colgap)); // Convierte
mstk a formato OpenCv en matlab_strainK

```

Una vez la imagen de deformación es calculada, se obtiene por medio de la función **GetStrain**, la cual retorna la variable que guarda los datos de la imagen de deformación.

```

Mat GetStrain(void)
{
    return(strainK); //retorna la variable strainK en formato Mat
}

```

#### 4.4.3 NormA1

Esta clase permite la normalización de la imagen de deformación por medio de un procedimiento que busca encontrar los parámetros, que mejor ajustan un plano de normalización de los Strains, a los datos de los desplazamientos. Se utiliza una función propia de Matlab para aproximación por mínimos cuadrados 'lsqcurvefit', por lo tanto se hace necesario el uso de una “librería compartida” como se explicó anteriormente en la clase **gradiente**. La librería que contiene la función que calcula la normalización se llama **normal.h**.

El constructor de la función **NormA1**, recibe como parámetros el desplazamiento axial y la imagen de deformación, además de los tamaños de columnas y filas respectivamente. Como se explicó anteriormente, para el uso de las aplicaciones hechas en matlab es necesario su inicialización para su posterior uso.

```

NormA1 nrm(Da,Stk,nrow,ncol,nrow2,ncol2); //inicializa los parámetros iniciales

```

```
normalInitialize() //inicializa la aplicación normal
```

Hecho esto se procede a invocar el método **NormA1\_calc**, de donde se llama a la función **Calc\_Norm**, la cual convierte las variables que se tienen, a un formato que se pueda interpretar en matlab haciendo uso de las variables cargadas con anterioridad tal como se muestra a continuación.

```
mxDouble * p_Da = om::allocateMatrixFromOpencv<mxDouble>(Da1); //Define memoria
marco de referencia Im1 en formato matlab
mxDouble * p_Stk = om::allocateMatrixFromOpencv<mxDouble>(Stk1); //Define memoria
marco de referencia Im1 en formato matlab

om::copyDataToMatlab(Da1,p_Da); // Copia los datos desde OpenCv a formato de
MatLab
om::copyDataToMatlab(Stk1,p_Stk);

mwArray m_Da(num_rows, num_cols, mxDOUBLE_CLASS); // Imagen Desplazamiento Axial que
retorna AM2D, formato Matlab
m_Da.SetData(p_Da, num_rows*num_cols); // Imagen desplazamiento MatLab
mwArray m_Stk( num_rows2, num_cols2, mxDOUBLE_CLASS); // Imagen Desplazamiento Axial
que retorna AM2D, formato Matlab
m_Stk.SetData(p_Stk, num_rows2*num_cols2); // Imagen desplazamiento MatLab
mwArray NS(num_rows2, num_cols2, mxDOUBLE_CLASS);
const mwArray met(1);
```

La función **normalización**, es la que se encarga de realizar el cálculo de la normalización a la imagen de deformación. Recibe como parámetros de entrada el desplazamiento axial **m\_Da**, la imagen de deformación **m\_Stk** y la variable entera **met** que determina el plano que se aplicara para la normalización, la cual tiene asociado el valor de 1 por defecto. Como parámetro de salida retorna los valores de la imagen de deformación normalizada en **NS**.

```
normalizacion(1,NS,m_Da,m_Stk,met);
NS.GetData(matlab_NS,(num_rows2)*(num_cols2)); // Convierte NS a formato OpenCv en
matlab_NS
```

Una vez la imagen de deformación es normalizada, es obtenida por medio de la función **GetNS**, la cual retorna la variable que guarda los datos de la imagen normalizada.

```
Mat NormA1::GetNS(void){
    return(NS);
}
```

#### 4.4.4 BMA

En esta clase es implementado un algoritmo de correspondencia de bloques con búsqueda de ventanas exhaustivas, tal como se explica en la sección 3.1.2 Estimación de desplazamiento híbrido.

El llamado a esta clase es echa desde la interfaz Elax1 como se muestra a continuación.

```
BMA str(imr,imd,_k1,_k2,comp,_kv,_lv)  
str.alg_SSD();
```

El constructor de la clase **BMA** pide o solicita los parámetros necesarios para que el método SSD pueda ser ejecutado, a continuación son descritos cada uno de estos parámetros.

imr, imd: Imagen de referencia y deformación en modo B.

\_k1, \_k2: Ancho y alto de la ventana o bloque de correspondencia.

comp: valor en porcentaje que indica el grado de compresión, que se estima, fue aplicado para deformar el tejido, este valor generalmente se encuentra en un rango de 0% a 10%.

\_kv, \_lv: indica respectivamente la separación axial y lateral entre cada ventana. Sí los valores de separación insertados son menores al ancho y alto de las ventanas, las ventanas se solaparan una con otras, el grado de solapamiento depende de que tan pequeños sean los valores suministrados.

Estos parámetros se encuentran inicializados por defecto para los las imágenes de base con las que se trabajó en la aplicación. Si desea cambiar dichos valores selecciona la opción de configurar parámetros que se encuentra en la aplicación (este procedimiento se explicara más adelante).

Una vez llamado el constructor de la clase **BMA** para inicializar los parámetros, el método **alg\_SSD** es ejecutado para correr el algoritmo que estimara los desplazamientos axiales y laterales.

A continuación es mostrada la estructura básica del método **alg\_SSD**.

```
// Algoritmo SSD

for (i=0;i<=(Nv_a-1);i++){ // Las matrices de desplazamiento son recorridas
    for (j=0;j<=(Nv_l-1);j++){
        y1=iny+i*kv;//cada punto en la matriz de desplazamiento equivale a un punto
            en la imagen de referencia.
        x1=inx+j*lv;
        prevSSD = 4000000.0; // Se inicia un valor del prevSSD en un valor
            alto
        temp = 0.0;
        umin=0;
        vmin=0;

        for (v=-V1;v<=V2;v+=1){ // v y u desplazan la ventana en una región de
            en la imagen de referencia.
            for (u =-U1;u<=U2;u+=k){
                sad=0.0;
                for (a=-mk1;a<=mk1;a++){// ubicación de cada pixel de una ventana
                    for (b=-mk2;b<=mk2;b++){
                        //encontrar el valor cada uno de los pixeles de una ventana
                        float ref=((uchar*)(imgref->imageData+(y1+a)*imgref->
                            widthStep))[x1+b];
                        float def=((uchar*)(imgdef->imageData+(y1+a+u)*imgdef->
                            widthStep))[x1+b+v];
                        // Suma de las diferencias al cuadrado
                        temp=ref-def;
                        sad=sad+temp*temp;
                    }
                }
                // De todos los sad dentro de una región de búsqueda, se
                toma el mayor, y los valores de u y v son asignados como
                los valores de desplazamientos de la ventana.
                if (prevSSD > sad)
                    {
                        prevSSD = sad;
                        umin=u;
                        vmin=v;
                    }
            }
        }
        //los valores de desplazamiento axial y lateral son
        guardados en las matrices de desplazamiento axial y
        lateral.

        Da.ptr<float>(i)[j]=umin;
        Dl.ptr<float>(i)[j]=vmin;
    }
}
```

## 4.5 Método Híbrido

En esta sección se explica las clases del método híbrido y su funcionamiento, las cuales son **buscador\_piramidal**, **Bsc** y **Lista**. Las clases adicionales como **strain**, **gradiente** y **NormA1** fueron explicadas con anterioridad en la el método SSD.

48

**Figura 30. Diagrama de clases del método Híbrido**

### 4.5.1 Buscador\_piramidal

Esta clase corresponde al algoritmo Multinivel [11], Este algoritmo está compuesto de un proceso piramidal de tres niveles, como resultado arroja una matriz de desplazamiento axial y lateral de resolución fina.

El llamado a esta clase es hecho desde la interfaz Elax1 como se muestra a continuación.

```
//calcular el desplazamiento
    buscador_piramidal str(imgr,imgd,imgr_RF,imgd_RF, cmp,_umbral_L2, _umbral_L1,
    _umbral_L0);
    str.piramide(&str);
```

El constructor de la clase **buscador\_piramidal** pide o solicita los parámetros necesarios para que el método híbrido pueda ser ejecutado, a continuación son descritos cada uno de estos parámetros.

Imr, Imd: Imagen de referencia y deformación en modo B.

Imr\_RF, Imd\_RF: Imagen de referencia y deformación de Radio Frecuencia.

Cmp: valor en porcentaje que indica el grado de compresión, que se estima, fue aplicado para deformar el tejido, este valor generalmente se encuentra en un rango de 0% a 10%.

\_Umbral\_L2, \_Umbral\_L1, \_Umbral\_L0: Rechaza las ventanas con coeficiente de similitud que se encuentren por debajo de los valores asignados en estos parámetros, estos umbrales pueden tomar valores entre 0 y 1, y son asignados a cada uno de los niveles del proceso piramidal.

Una vez llamado el constructor de la clase **buscador\_piramidal** para inicializar los parámetros, el método **pirámide** es ejecutado para correr el algoritmo que estimara los desplazamientos axiales y laterales.

A continuación es mostrada la estructura básica del método **pirámide**.

```
void buscador_piramidal::piramide(buscador_piramidal* stt)
{
```

```
    for (level = 0; level <= maxlevel-1; level++)
    {
```

```
        //
        //
        //
```

```
    }
}
```

Aquí son guardadas tres pares de imágenes de referencia y deformación sub-muestreadas en 1, 1/2 y 1/4 del total de los datos (píxeles).

```
        // Procesamiento de cada nivel
        for (level = maxlevel; level >= 0; level--)
```

```
        {
            refer = refes[level-1];
            defor = defes[level-1];
```

Imágenes de referencia y deformación sub-muestreadas según el nivel del ciclo for.

```
        if (level == (maxlevel))
```

```
            {
                //
            }
```

Configuración o ajuste de variables según el nivel en la pirámide.

```

//
    }
    if(level==(maxlevel-1))
    {
        //
        //
    }
    if(level==(maxlevel-2))
    {
        //
        //
    }
    bsc primero(stt);
    primero.bsc_híbrido(&primero,stt);
}
}

```

El método `bsc_híbrido` de la clase `bsc` es llamado para ejecutar el algoritmo de búsqueda que calcula los desplazamientos axiales y laterales de según el nivel de la pirámide.

#### 4.5.2 Bsc

Esta clase estima los desplazamientos axiales y laterales para cada nivel de la pirámide tal como se explica en **3.1.2 Descripción del método**. A continuación se explica el siguiente método:

```
Void bsc_hibrido(bsc* prim,buscador_piramidal* stt);
```

Es utilizado para ejecutar el método **alg\_buscador** bajo ciertas condiciones, teniendo en cuenta el nivel en que se encuentra el proceso piramidal. En este método también es ejecutado los métodos **corrección**, **interpolacion1**, además de algunos métodos de la clase **Lista**.

A continuación es mostrado los llamados más importantes de la clase **Bsc**.

```
modo='0';
prim->alg_buscador(&lis, modo);
```

Aquí se llama a **alg\_buscador** teniendo como parámetros un objeto (&lis) creado por la clase **Lista** y un parámetro “modo” que si está en 0 indica que **alg\_buscador** interpolara las semillas encontradas en un nivel superior a un nivel inferior. Si el proceso piramidal se encuentra en el nivel máximo, este llamado es usado para estimar el total de las ventanas de ese nivel.

```
modo='1';  
prim->alg_buscador(&lis,modo);
```

En este caso **alg\_buscador** utiliza el algoritmo de calidad guiada para estimar el desplazamiento del total las ventanas, a partir de las semillas estimadas previamente (modo='0').

```
if(_level==1){  
prim->interpolacion1();// interpolar las ventanas erróneas  
puntos.clear();  
}
```

Este llamado permite interpolar las ventanas erróneas (con coeficiente de similitud por debajo del umbral) de la matriz de desplazamiento axial y lateral del nivel 1.

```
if(_level==0){  
prim->correccion(prim); //eliminar pixeles que vienen de malas semillas//  
prim->interpolacion1();// interpolar las ventanas erróneas  
}
```

En este llamado son remplazadas las ventanas de mala calidad que son consecuencia de la implementación del algoritmo de calidad guiada. La **corrección** solo se hace al final del nivel 0. **interpolacion1** también es aplicado en el nivel 0.

### 4.5.3 Lista

La clase **Lista** es implementada para crear la tabla Ranking que guardara los datos de cada ventana que ya ha sido estimada.

Esta clase tienen funciones que permiten insertar, mostrar, borrar datos de la tabla ranking, buscar datos en específico, y resetear la tabla.

A continuación se explica el llamado que se hace a algunas de las funciones de la clase **Lista**. Cabe aclarar que el llamado es hecho desde la clase **Bsc**.

```
lis->insertar(i,j,umin,vmin,prevCC,id_semilla);
```

Este método es llamado una vez estimado los desplazamientos de una ventana, para que sean guardados en la tabla Ranking. Los parámetros que se insertan en el Ranking son:

i, j: ubicación de la ventana en la imagen.

Umin, vmin: desplazamiento axial y lateral estimado de la ventana.

prevCC: Coeficiente de similitud que indica la calidad de la estimación.

Id\_semilla: Identificador que permite saber de qué semilla proviene la estimación de esta ventana.

```
lis-> buscar(i_0, j_0, umin_0, vmin_0,id_semilla);
```

Este método se usa para escoger de todas la estimaciones que están guardadas en la tabla Ranking, la que tenga el coeficiente de similitud más alto, es decir con mejor calidad, para que posteriormente los datos de esta estimación, sirvan de guía o punto de partida para estimar los desplazamientos de las ventanas vecinas.

Los parámetros que arroja o retorna este método son:

i\_0, j\_0: ubicación de la ventana con mejor calidad.

umin\_0, vmin\_0: desplazamiento axial y lateral estimado de la ventana con mejor calidad.

Id\_semilla: Identificador que permite saber de qué semilla proviene la estimación de la ventana con mejor calidad.

## 4.6 Método AM2D

El método AM2D o de programación dinámica se trabajó por medio de librerías compartidas de Matlab, que permitieron el uso de dicho algoritmo haciendo uso de un archivo Mex<sup>16</sup> asociado. Esto se hizo debido a que la inclusión y elaboración de los algoritmos y funciones resultaba más práctico, además que Matlab permite que la creación de dichas librerías puedan usarse en IDE's como Visual Studio para su compilación.

...

### Figura 31. Diagrama de clases método AM2D

A comparación de los dos métodos expuestos anteriormente, el método de programación dinámica no hace uso de las clases **strain** y **gradiente**. Para su funcionamiento, la clase **Strains\_AM2D** necesita declarar los valores iniciales de los parámetros que necesita para su funcionamiento.

El llamado y la declaración de valores de los parámetros para esta clase son hechos desde la interfaz Elax1 como se muestra a continuación.

```
double IRF[2] ={-100,0};  
double IA[2] = {4,4};  
int midA = 450;  
double alfa_DP = 0.15;  
double alfa = 5;  
double beta = 10;
```

---

<sup>16</sup> Mex, es una extensión de archivo de Matlab que contiene funciones y librerías del desarrollador

```

double gamma = 0.005;
double T = 0.2;
double a_t = 0.63;
double f_0 = 7.2727;
double f_s = 40;

// Initialize the MATLAB Compiler Runtime global state
strains_AM2D str(&Im3,&Im4, nrR, ncR, IRF, IA, midA, alfa_DP, alfa, beta, gamma, T,
a_t, f_0, f_s);

```

Una vez se cargan los parámetros y se crean las variables, se inicializa la aplicación **am2dlib** y de ahí se procede a hacer el cálculo del desplazamiento axial, para ello se invoca el método **CalcDisp**.

```

am2dlibInitialize() // inicialización de la aplicación am2dlib

...

// Cálculo de Desplazamiento Axial
str.CalcDisp();

```

Este método llama a la función **AM2D\_algorithm**, que es la encargada de convertir las variables que se tienen, a un formato que se pueda interpretar en matlab haciendo uso de las variables cargadas con anterioridad tal como se mostró en casos anteriores.

```

void strains_AM2D::CalcDisp(void)
{
    strains_AM2D::AM2D_algorithm();
    ...
}

```

Una vez se encuentren las variables listas, se invoca la función **Cpp\_AM2D** de matlab que se encarga de realizar el cálculo del desplazamiento axial y guardar los valores encontrados en la variable **m\_Da**. La función recibe como parámetros de entrada la imagen de referencia y deformación, además de los parámetros de entrada y retorna tres valores de los cuales destacamos el desplazamiento axial y el desplazamiento lateral.

```

Cpp_AM2D(3, m_Da, m_Dl, m_DPdisparity, m_Im1, m_Im2, m_IRF, m_IA, m_midA, m_alfa_DP,
m_alfa, m_beta, m_gamma, m_Tf, m_xx); // Función que envía parámetros a una función
creada en Matlab y retorna las variables con los desplazamientos axial y lateral

```

Terminado el proceso de cálculo de los desplazamientos, se procede a calcular la imagen de deformación. Para ello se invoca desde la interfaz el método **CalcStrain**.

```
// Cálculo de Strain Axial  
str.CalcStrain();
```

Este método llama a la función de Matlab **Cpp\_Kal\_LSQ**, que es la encargada de calcular la imagen de deformación. La función recibe como parámetros de entrada la imagen de desplazamiento axial **m\_Da** y una constante de tipo entero **wDIFF** que permite realizar un recorte a los bordes de la imagen resultante. Como parámetro de salida retorna en la variable **mstk** la imagen de deformación.

```
Cpp_Kal_LSQ(1, mstk, m_Da,wDIff); // Obtiene en mstk la imagen de strains en formato  
Matlab
```

Una vez haya terminado de realizar los cálculos correspondientes, se utilizan las funciones que retornan las variables del desplazamiento axial y la imagen de deformación para su posterior uso en la clase **normalización** y mostrar los resultados en la interfaz.

```
ImDA = str.GetImgDa(); //Función que retorna la imagen de desplazamiento  
...  
ImGA = str.GetStrain(); //Función que retorna la imagen de deformación
```

## 4.7 Funciones asociadas a la interfaz

Por último se explicaran las funciones más importantes que se encuentran asociadas a la interfaz desarrollada. Dichas funciones son las que permiten desde la adquisición de los datos de entrada, su configuración, uso y posterior visualización.

### 4.7.1 Método ELTGP

Este método está asociado a la interfaz de usuario. Es aquí donde se procede a crear y configurar las acciones de los botones y demás herramientas usadas en la interfaz de usuario. Tiene asociados los archivos **eltgp.h** y **eltgp.cpp**, los cuales son invocados desde el menú de inicio de la aplicación.

```

#include "eltgp.h"
...

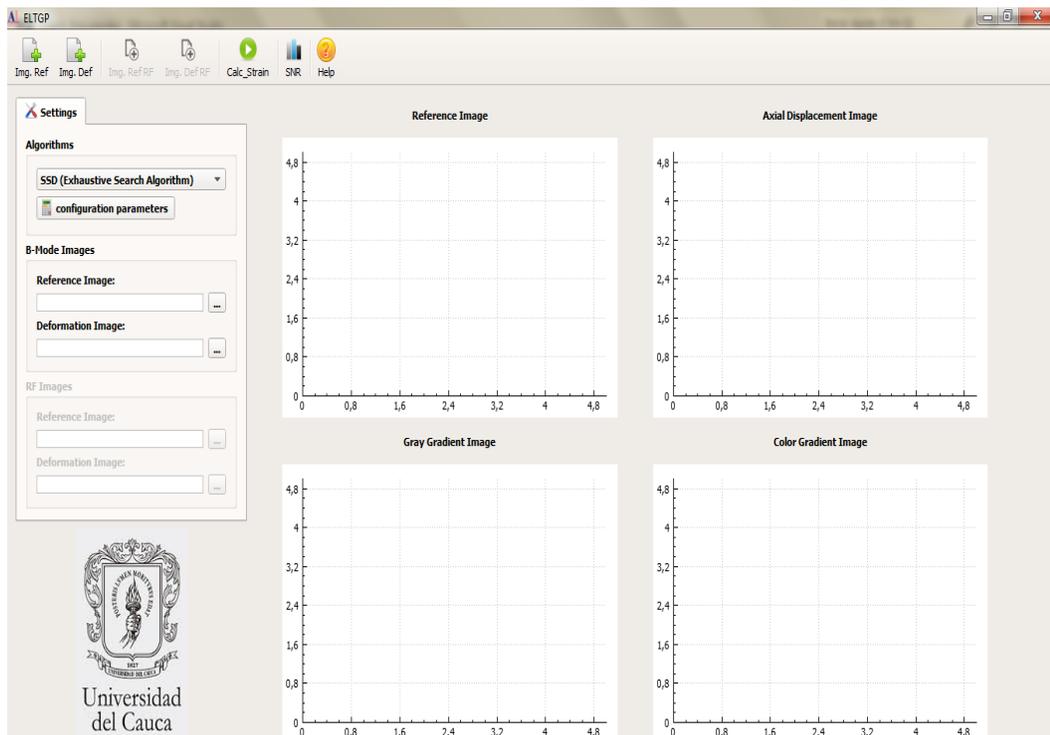
int main(int argc, char *argv[])
{
    QApplication::setStyle(QStyleFactory::create("Fusion"));
    QApplication a(argc, argv);
    ELTGP w;
    w.show();

    return a.exec();
}

```

Los archivos **.h** y **.cpp** que se mencionaron anteriormente, se crean automáticamente cuando se inicia un nuevo proyecto en qt-creator. Para mayor información ver anexo B.

Cuando se ha terminado de diseñar la interfaz de acuerdo a los requerimientos que se solicitaron, se procede a crear las acciones asociadas a cada una de los botones y demás herramientas.

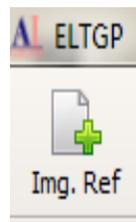


**Figura 32. Interfaz Elax1**

Las acciones que se crean, son funciones que se activan ya sea dando clic sobre un botón o cambiando la opción de selección en una lista. Como ejemplo, se tienen las funciones que permiten añadir los archivos de entrada de datos para el cálculo de las deformaciones.

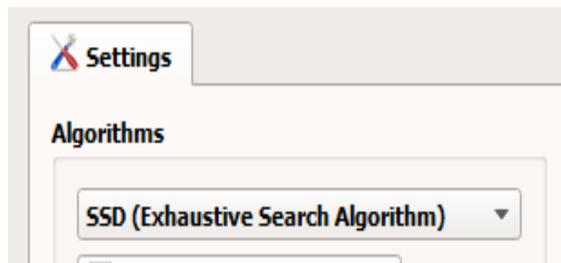
```
void ELTGP::on_toolButton_clicked(); //Función que permite la lectura de archivos  
//.YML que contiene los datos de la imagen de referencia.
```

Esta función se encuentra asociada al botón de la figura 33.



**Figura 33. Botón agregar imagen de referencia**

Existe otra función, que permite a partir de la posición donde se encuentre la pestaña (figura 34), la selección y uso de los algoritmos comentados en este capítulo.



**Figura 34. Pestaña para la selección del algoritmo**

El cambio de la pestaña para la selección del algoritmo, tiene asociada la siguiente función:

```
void ELTGP::calc_strain();
```

...

```

switch (ui->comboBox->currentIndex()){
    case 0:
        ...
    }else{
        //funcion para calcular el desplazamiento
        BMA str(&Im1,&Im2,71,41,6,48,28);
        str.alg_SSD();
        //calcular el gradiente
        gradiente grd(str.Da,nrR,ncR);
        grd.CalcStrain();
        ImDA = str.GetImgDa();
        ImGA = grd.GetStrain();
        ...
    }break;
}

```

Dentro de la función se invocan los métodos para el cálculo de la imagen de deformación como se explicó con anterioridad.

Dentro de la interfaz hay la facilidad de llamar procesos externos, que permiten la configuración o vista de resultados. Para la configuración de parámetros se hace uso de estos llamados, ofreciéndole al usuario la posibilidad de cambiar los parámetros asociados a cada algoritmo, logrando una mejora o no de los resultados obtenidos.

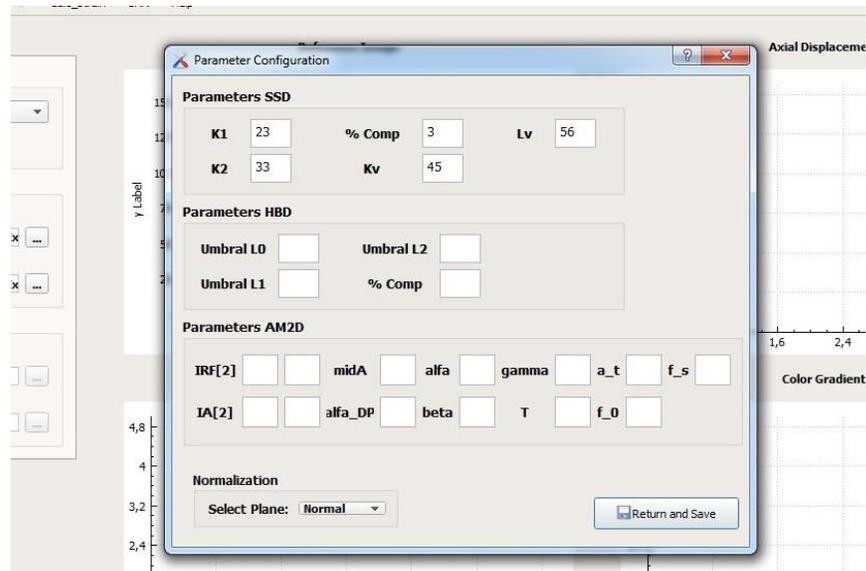


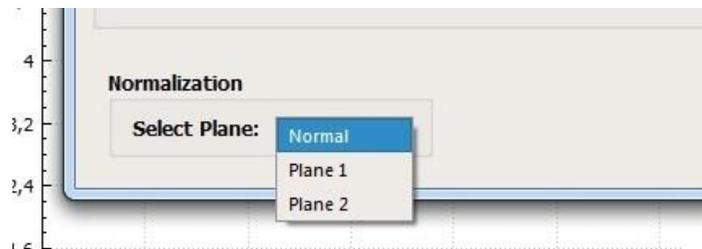
Figura 35. Ventana auxiliar para configuración de parámetros.

La ventana de configuración de parámetros tiene asociada una clase llamada **conf\_par**, creada en Qt. Para hacer uso de ella se pone el nombre de cabecera **conf\_par.h** y se invoca por medio de la siguiente función:

```
void ELTGP::help_win(){
    Help help;    //creacion del objeto tipo Help
    help.exec(); //ejecuta la acción y muestra la ventana creada
}
```

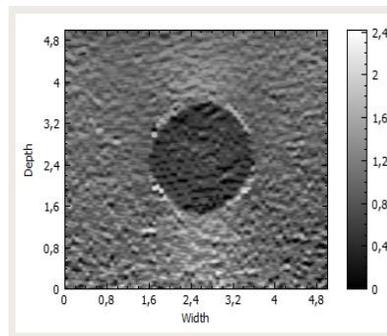
Esto permite que la ventana se abra externamente sin interrumpir los procesos de fondo como se observa en la figura 35.

Fuera de la configuración de parámetros en esta ventana se tiene la opción de seleccionar el plano de normalización que se desea aplicar a los resultados de deformación, tal como se ve en la figura 36.



**Figura 36. Selección del plano de normalización.**

Por ultimo hay una función que permite mostrar los resultados y parte de los archivos que cargamos en la interfaz como imágenes ploteadas en un plano figura 37.



**Figura 37. Imagen de deformación ploteada**

Para el uso de estos objetos es necesario la inclusión la librería **qcustomplot**<sup>17</sup> que trabaja sobre el entorno de Qt permitiendo mostrar los datos de las imágenes que calcula la interfaz, en un plano. Para ello se invoca la siguiente función:

```
void ELTGP::setup_graph(QCustomPlot *customPlot,int i,Mat Img);
```

La función recibe tres parámetros; **customplot** es un valor numérico entero que permite determinar el plano en el que se mostrará la imagen, **i** permite configurar el color de gradiente de la imagen de escala de grises a escala de color y por último **Img** es la variable que guarda los datos de las imágenes a mostrar.

#### 4.7.2 Método SNR\_Dialog

En este procedimiento se crea una ventana nueva que deja mostrar y configurar el cálculo de la función señal a ruido sobre la imagen de deformación. La creación de esta interfaz genera igual que en la interfaz principal los archivos **SNRDialog.h** y **SNRDialog.cpp** correspondientes. La figura 38 muestra la ventana en uso mostrando los resultados.

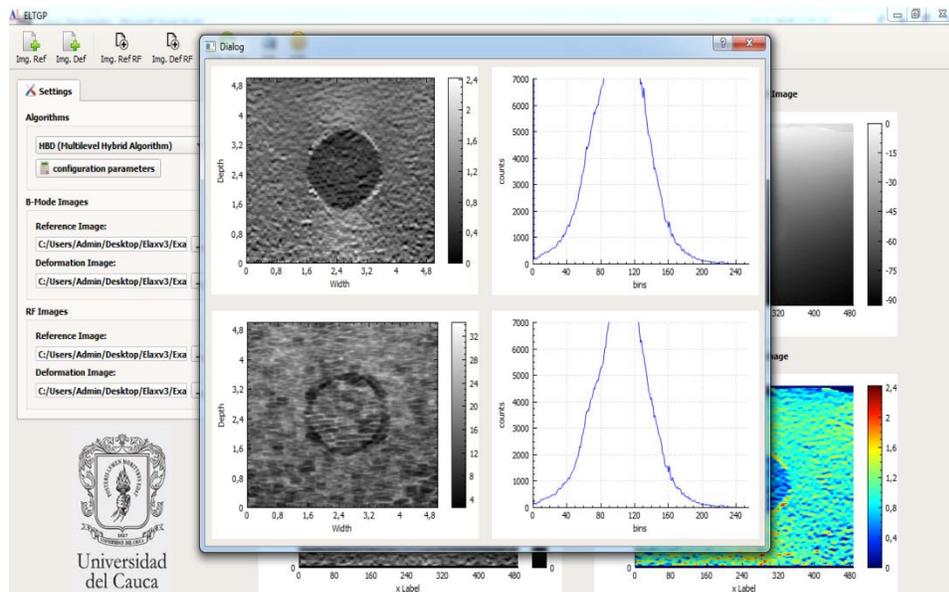


Figura 38. Ventana calculo señal a ruido de imagen de desplazamiento

<sup>17</sup> Librería de uso libre, que trabaja sobre el entorno de Qt permitiendo mostrar graficas de funciones e imágenes en un plano. Para mayor información diríjase a la página [<http://www.qcustomplot.com/>]

Se invoca a través de la siguiente función:

```
void ELTGP::calc_snr(){
    SNRDialog snrdialog; //creacion del objeto tipo Help
    snrdialog.copiar_data(ImGA); //parámetro necesario para el calculo
    snrdialog.exec(); //ejecuta la acción y muestra la ventana creada
}
```

Esto permite que la ventana se abra externamente sin interrumpir los procesos de fondo. Dentro de **SNRDialog.h** se crearon funciones para realizar el cálculo sobre la imagen de deformación y obtener los resultados requeridos.

```
void snr(Mat* ); //Función que calcula la SNR de la imagen de deformación
void plot_data(Mat,int); //Plotea los datos de las imágenes
void histogram(Mat); //Calcula el histograma de la imagen de deformación
```

Para mayor información sobre la creación de nuevas ventanas y su uso véase el anexo B.

## 5 RESULTADOS

El software Elax1 es utilizado en este capítulo para la obtención de imágenes de desplazamiento axial, imágenes de gradiente axial en escala de grises y a color, relaciones señal a ruido (SNR), imágenes de deformación con planos de normalización y relaciones de deformación. En Elax1 están disponibles los siguientes algoritmos de elastografía: Correspondencia de Bloques con Búsqueda Exhaustiva (SSD), AM2D e Híbrido (HBD). Los resultados arrojados por cada uno de estos algoritmos serán analizados y comparados más adelante.

Este proyecto no conto con máquinas de ecografía que permitan la obtención directa de datos RF de exploración en vivo de pacientes o de Phantoms. Los datos RF tanto de pacientes como de Phantoms se obtuvieron de la base de datos [1]. Los datos RF y modo B de estos pacientes, se denominan en [1] y en este trabajo como paciente 3 y paciente 4. La base de datos contiene los datos RF estudiados por Hassan Rivas en [1] usando el algoritmo AM2D.

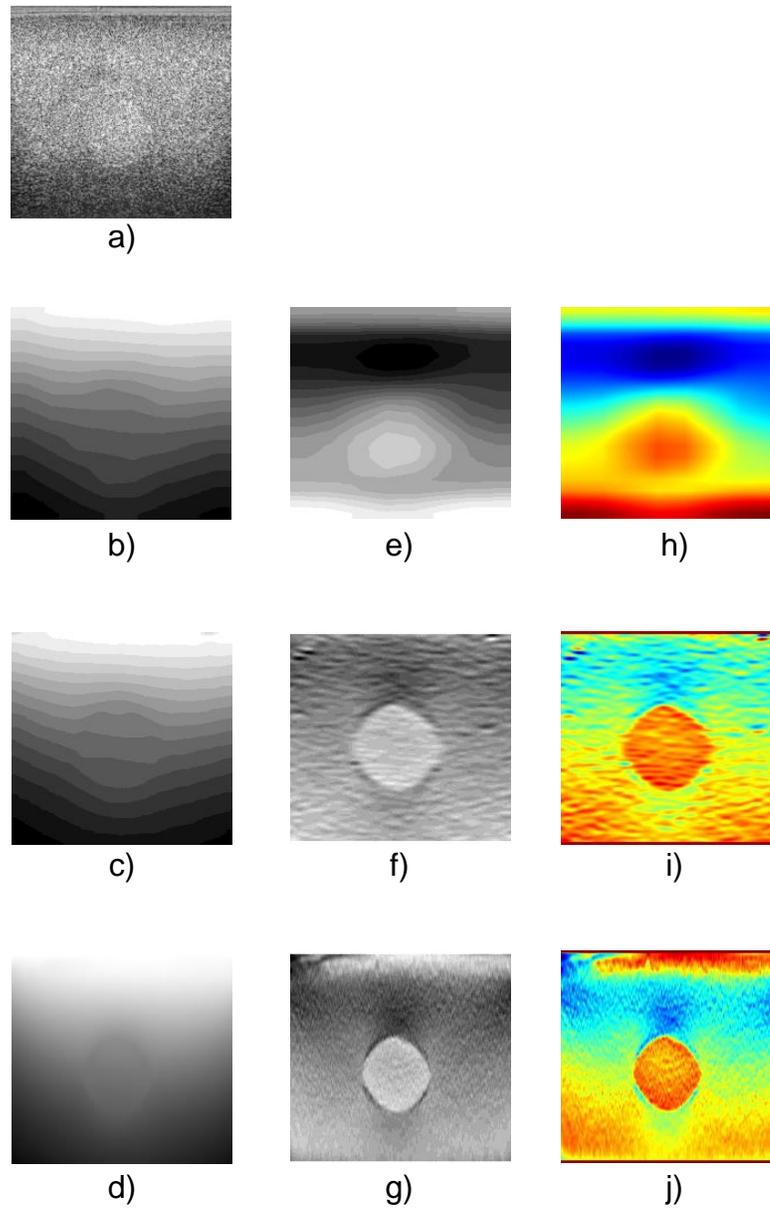
En 5.1 y 5.2 se evaluara el desempeño de los algoritmos SSD y HBD, para ello un algoritmo (AM2D) que ya ha sido evaluado en [1], es utilizado para hacer una comparación de resultados.

A continuación se muestran, analizan y comparan los resultados obtenidos con Elax1 usando la base de datos de [1].

### 5.1 Resultados Phantom

Un Phantom experimental obtenido de [1], es utilizado para la obtención de las primeras pruebas y resultados. Imágenes de deformación del Phantom son analizadas con y sin el uso de planos de normalización en los tres métodos de elastografía estudiados por este proyecto.

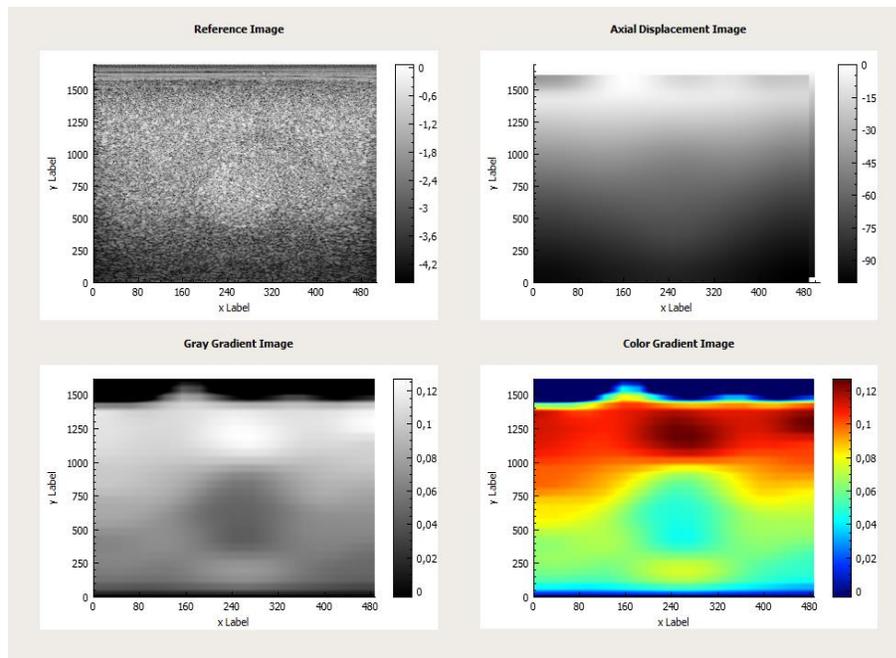
En la figura 39 se puede observar una compilación de los resultados obtenidos en Elax1 sin normalización, usando el Phantom de prueba.



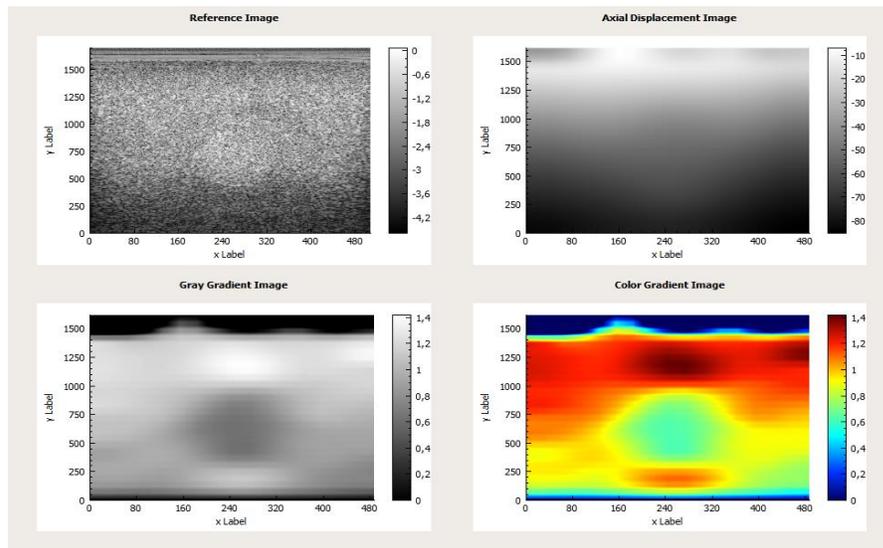
**Figura 39. Resultados obtenidos Phantom de prueba sin normalización**

En la figura 39, (a) es la imagen en modo B del Phantom de prueba, las imágenes correspondientes a (b), (c) y (d) son los desplazamientos axiales calculados por cada uno de los algoritmos mencionados anteriormente. Las imágenes (e), (f) y (g) son las imágenes de deformación o elastogramas producto del cálculo de la deformación aplicado en Elax1; las imágenes (h), (i) y (j) son las mismas imágenes de deformación solo que se les aplica un gradiente de color que permite hacer un análisis visual sobre los tejidos examinados.

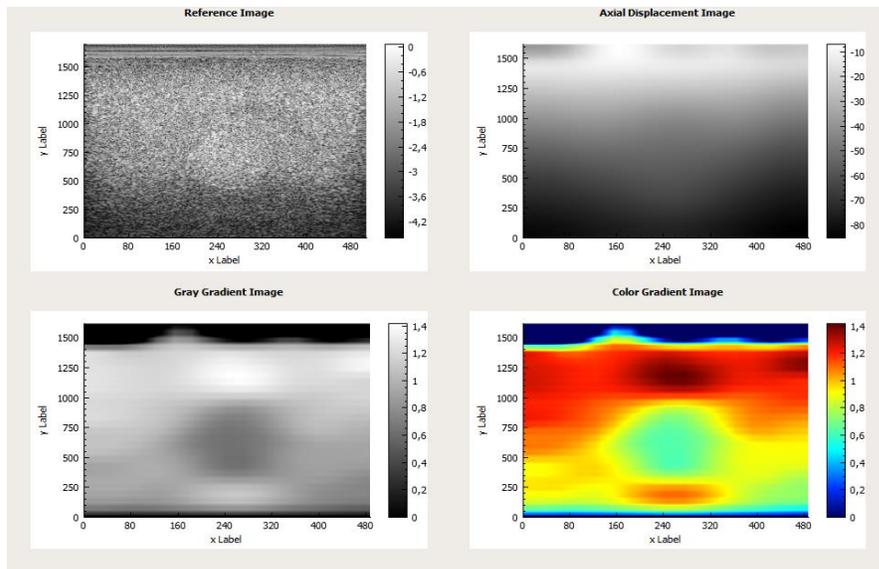
Ahora, desde las figura 40 hasta la figura 48 se procede a mostrar los resultados arrojados por la herramienta Elax1, haciendo énfasis en el uso de los planos de normalización que se pueden aplicar a los resultados.



**Figura 40. Resultados del algoritmo SSD sin plano de normalización.**

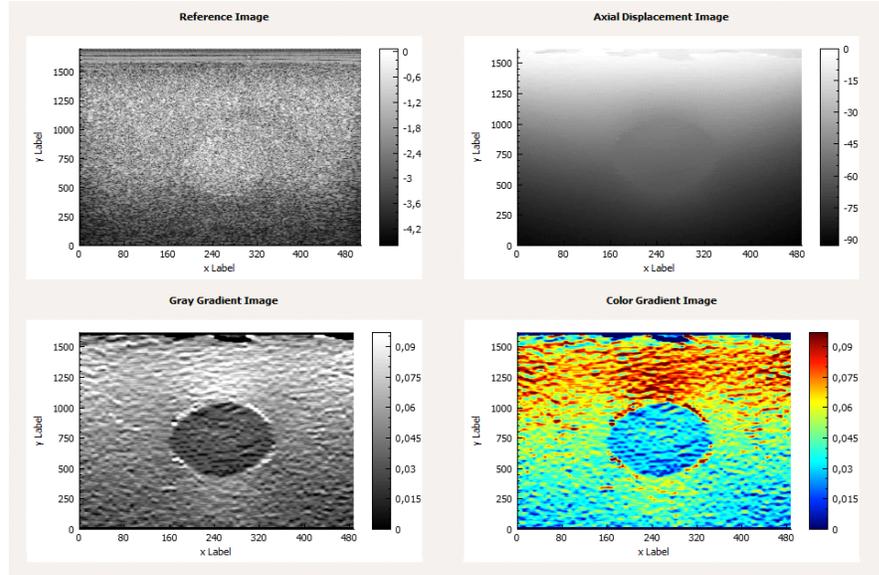


**Figura 41. Resultados del algoritmo SSD con el plano de normalización 1.**

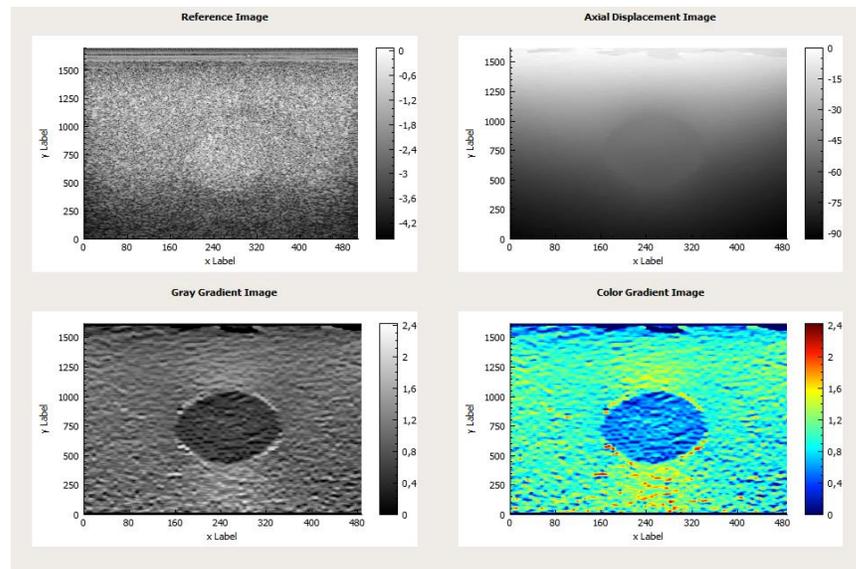


**Figura 42. Resultados del algoritmo SSD con el plano de normalización 2.**

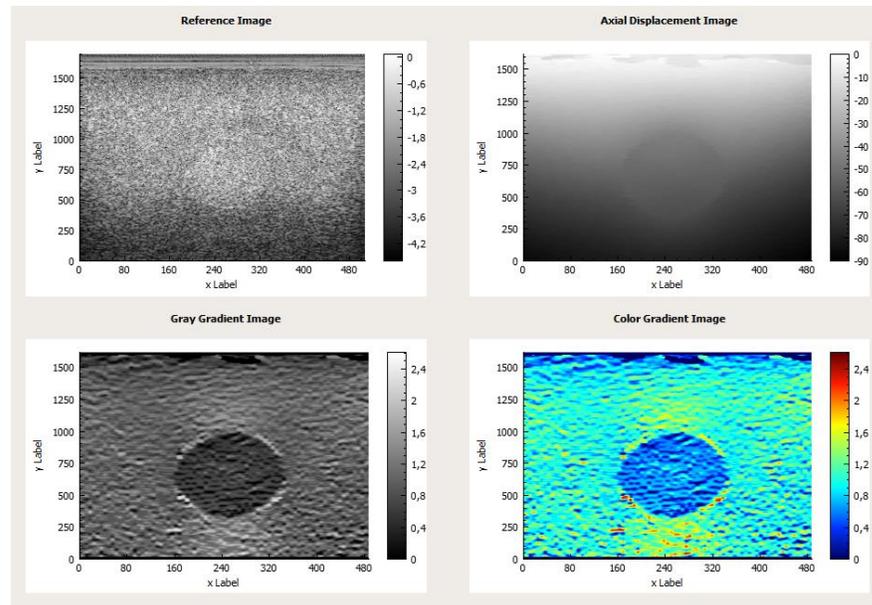
Como se puede observar los resultados que se obtienen después de la normalización, permiten ver una imagen con tonos suavizados y uniformes.



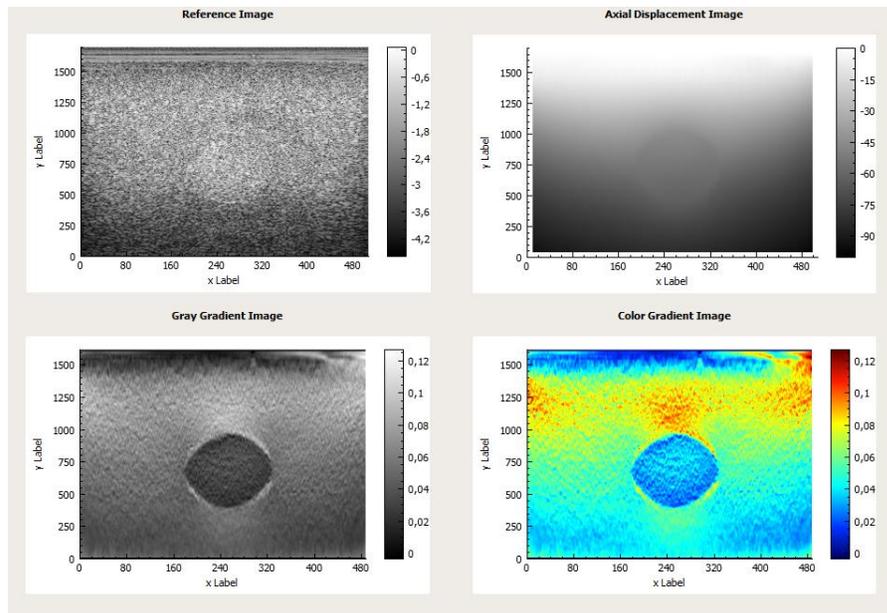
**Figura 43. Resultados del algoritmo HBD sin plano de normalización.**



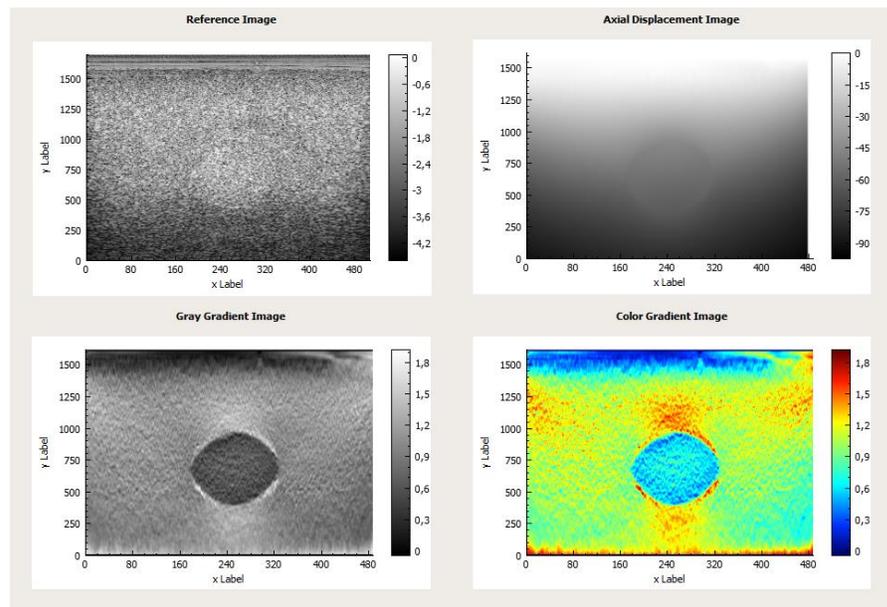
**Figura 44. Resultados del algoritmo HBD con el plano de normalización 1.**



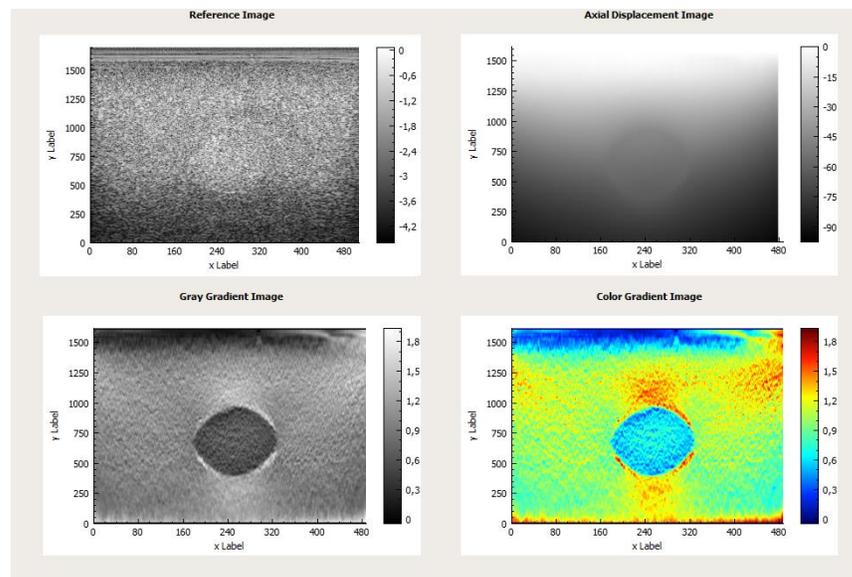
**Figura 45. Resultados del algoritmo HBD con el plano de normalización 2.**



**Figura 46. Resultados del algoritmo AM2D sin plano de normalización.**



**Figura 47. Resultados del algoritmo AM2D con el plano de normalización 1.**



**Figura 48. Resultados del algoritmo AM2D con el plano de normalización 2.**

En las anteriores figuras se puede observar cuatro ventanas de la interfaz de Elax1. La ventana Reference Image visualiza la imagen modo B del Phantom, la ventana Axial Displacement Image visualiza el desplazamiento axial, la ventana Gray Gradient Image visualiza la deformación axial del Phantom representada por el gradiente del desplazamiento en escala de grises, donde el color blanco hace referencia al valor de deformación más alto del tejido y el negro al valor de deformación más bajo del tejido, tal como se muestra en la paleta de colores. En la ventana Color Gradient Image se puede visualizar la imagen de deformación axial en escala de colores donde el color rojo representa el valor de deformación más alto del tejido, y el color azul el valor de deformación más bajo del tejido, tal como se muestra en la paleta de colores.

Las figuras 40, 43 y 46 muestran los resultados de los tres algoritmos sin el uso de un plano de normalización. Como se puede observar, las imágenes de deformación indican que el tejido superior del Phantom es más elástico que el tejido inferior, pero esta afirmación es errónea, el tejido superior del Phantom al estar más cerca al punto de contacto con el sonógrafo, recibe más fuerza de compresión y por consiguiente sufre mayor deformación. Para que sea posible una correcta interpretación de la imagen de deformación axial del Phantom, se debe hacer equitativa la distribución de fuerzas de compresión en el tejido, es por eso que un plano de normalización es necesario, ya que permite cancelar la distribución descendente de fuerzas de compresión con un aumento de intensidad en los valores de deformación en las regiones superiores de la imagen y una disminución de intensidades en los valores de deformación de las regiones inferiores de la imagen. En las figuras 41, 42, 44, 45, 47 y 48 se puede obtener

imágenes de deformación que permitan una interpretación más aproximada a la conformación elástica real del Phantom.

En las figuras 41, 44 y 47 se utiliza el plano de normalización 1, mientras que en las figuras 42, 45 y 47 se utiliza el plano de normalización 2. Una consecuencia de utilizar estos planos de normalización es un cambio considerable de proporción en los valores de deformación. En la paleta de colores de la figura 7 se puede observar un rango de deformación entre 0% y 12%, mientras que en la figura 8 se observa un rango de deformación entre 0 y 180%, pese a que estos valores de deformación son erróneos en términos de magnitud, siguen guardando un valor de deformación relativo con el tejido circundante, que es lo que permite una correcta interpretación cualitativa y semicuantitativa de la deformación de un tejido.

En las figuras 41, 42, 44, 45, 47 y 48 se pudo observar, en términos generales, buenas imágenes de deformación por parte de los 3 métodos del Elax1. Para los 3 métodos se determinó que el nódulo del Phantom es más rígido que en sus regiones contiguas, además fue posible visualizar claramente el tamaño y la geometría de la región dura, es decir del nódulo. Fue notable también la superioridad en calidad de resultados del método AM2D con respecto a los métodos HBD y SSD, y la superioridad de resultados del método HBD con respecto al método SSD.

Una herramienta muy útil para lograr un diagnóstico más objetivo de una enfermedad cancerígena es la **Relación De Deformación**. En las figuras 49, 50 y 51 es mostrado su funcionamiento para los tres métodos de elastografía estudiados en este proyecto.

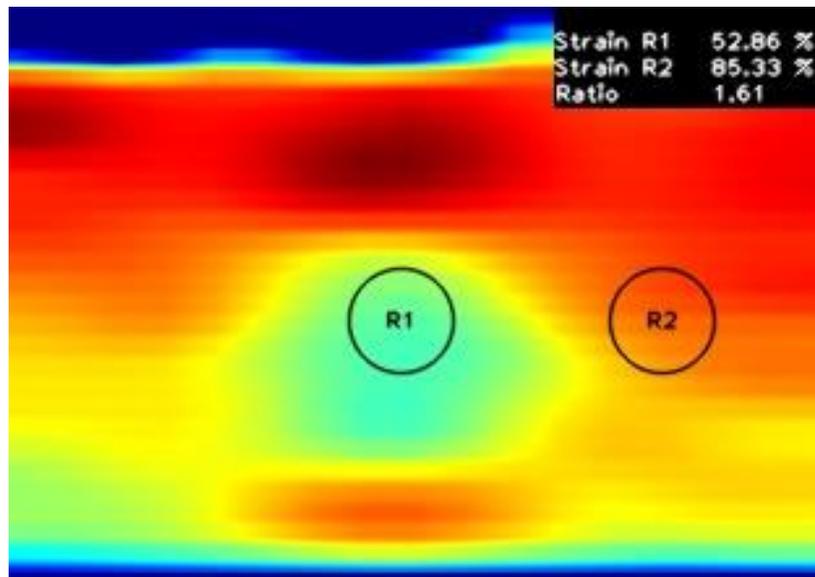


Figura 49. Relación De Deformación usado en el método de elastografía SSD.

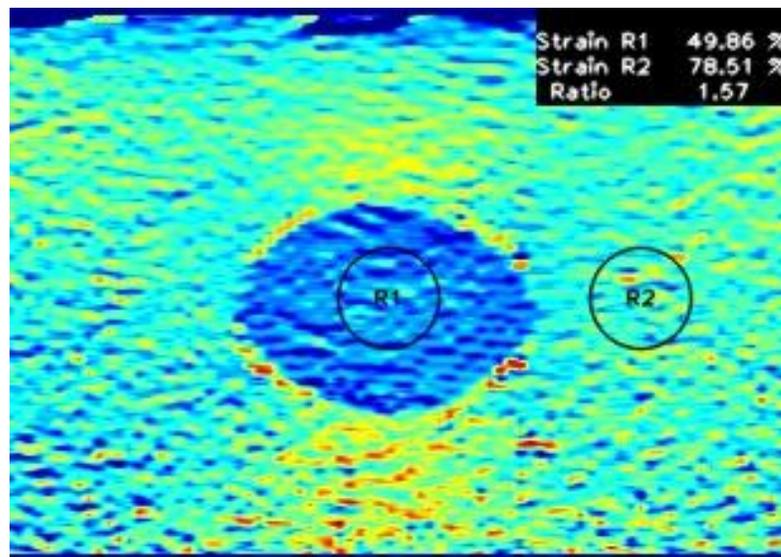
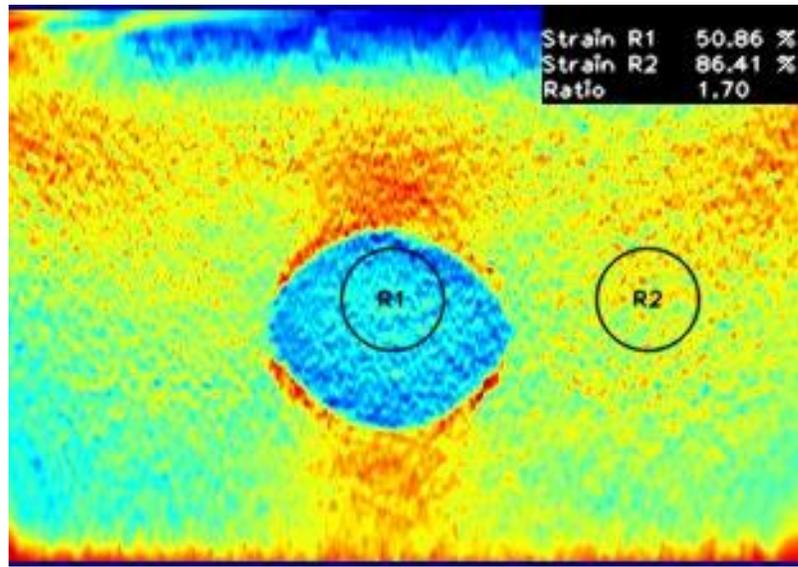


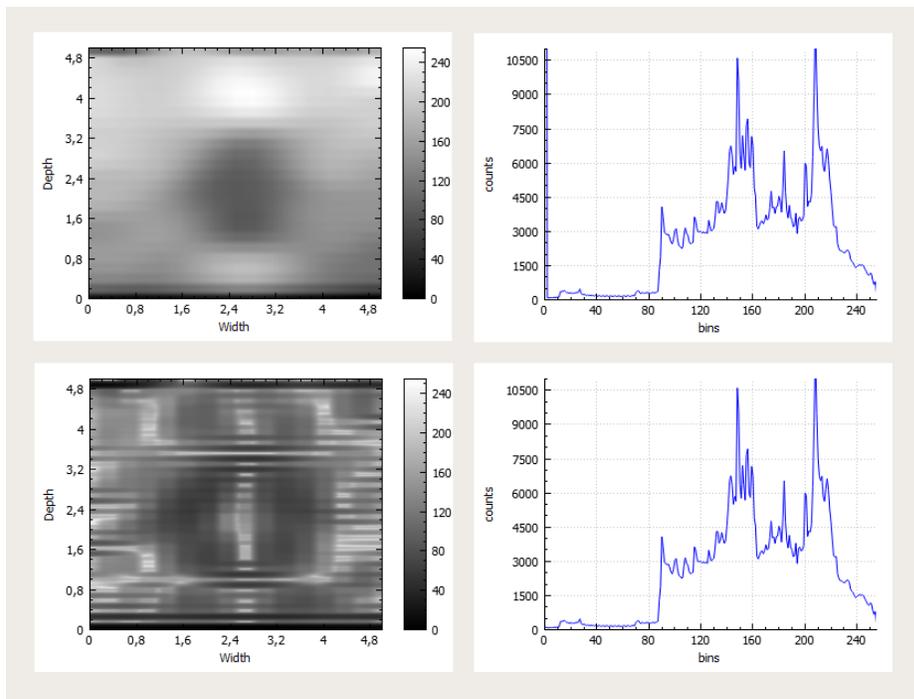
Figura 50. Relación De Deformación usado en el método de elastografía HBD.



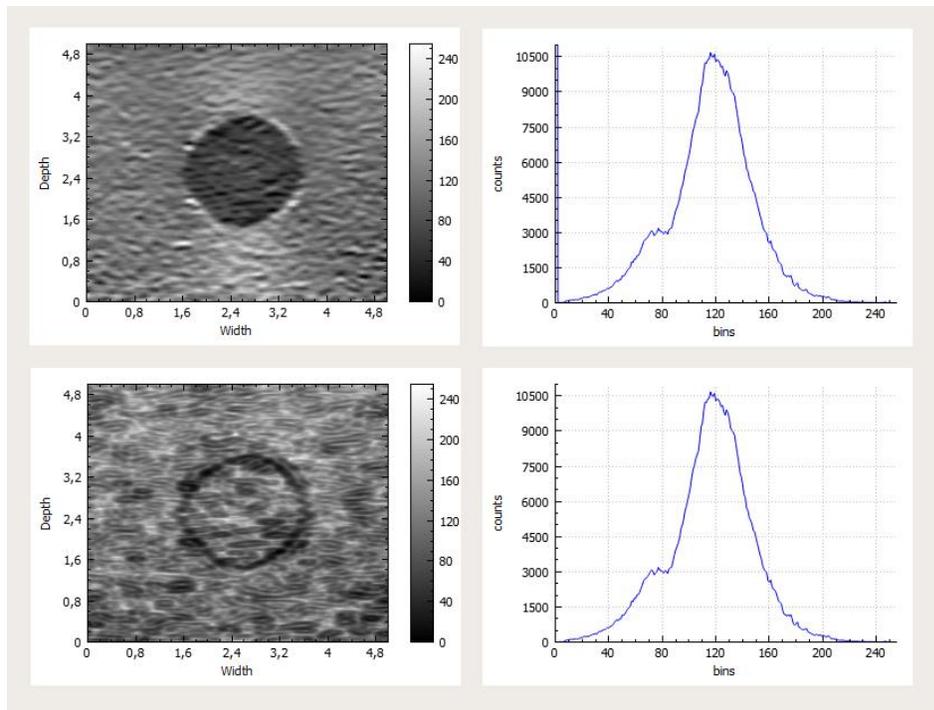
**Figura 51. Relación De Deformación usado en el método de elastografía AM2D.**

De las anteriores figuras se puede resaltar dos regiones de interés R1 y R2, siendo R1 una región del tejido situada en el nódulo del Phantom y R2 una región de interés situada en el tejido adyacente al nódulo. La razón de deformación para los métodos SSD, HBD indica que la región R1 es aproximadamente 1.6 veces más rígida que la región R2 y en el método AM2D R1 es aproximadamente 1.7 veces más rígida que la región R2.

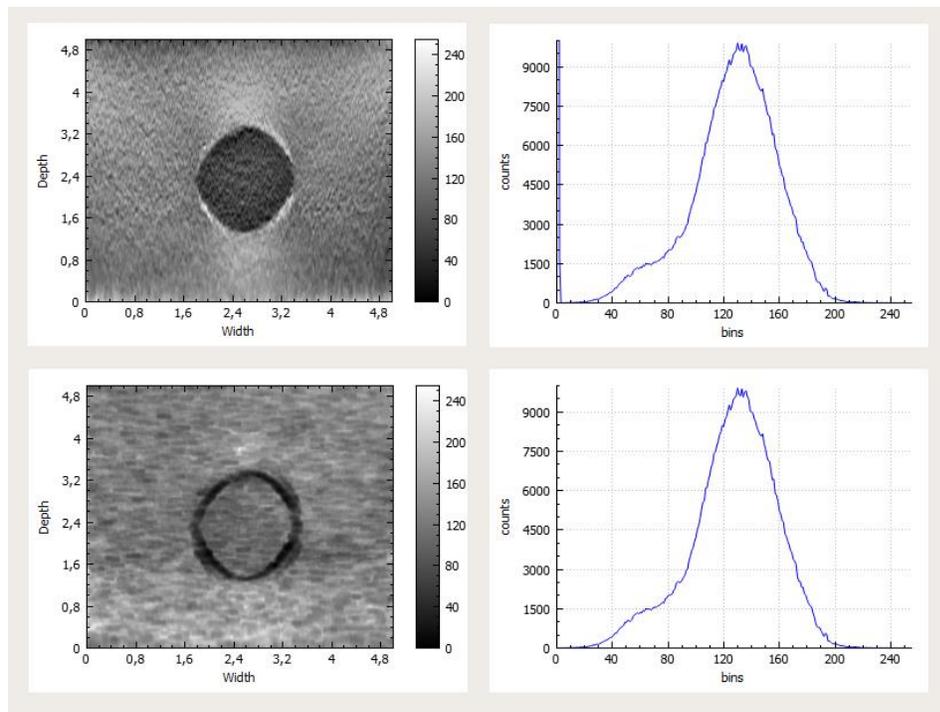
Otra herramienta que posee Elax1 es una opción que muestra la distribución de píxeles de la imagen de deformación (SNR). SNR puede ser útil para analizar cuáles son los píxeles o rangos de píxeles con mayor concentración en la imagen de deformación y por ende es posible analizar cuáles son las mayores concentraciones de deformación. En las figuras 52, 53 y 54 se muestra el cálculo de SNR de las imágenes de deformación del Phantom para cada uno de los métodos de elastografía del elax1.



**Figura 52. SNR de imagen de deformación del método SSD.**



**Figura 53. SNR de imagen de deformación del método HBD.**



**Figura 54. SNR de imagen de deformación del método AM2D.**

De las anteriores figuras, se puede ver que el SNR del método SSD es muy ruidoso y no hace posible un análisis adecuado de la imagen de deformación. El SNR de los métodos HBD y AM2D son muy parecidos, se puede decir que las mayores concentraciones de píxeles están aproximadamente entre 110 y 160 para el AM2D y entre 110 y 140 para el HBD y hacen referencia al tejido circundante al nódulo del Phantom, mientras que hay una concentración de píxeles entre 50 y 80 para el AM2D y 70 y 90 para el HBD que hacen referencia al nódulo del Phantom, tal como se observa en la imagen de deformación.

5.2 Resultados Paciente 3 y 4

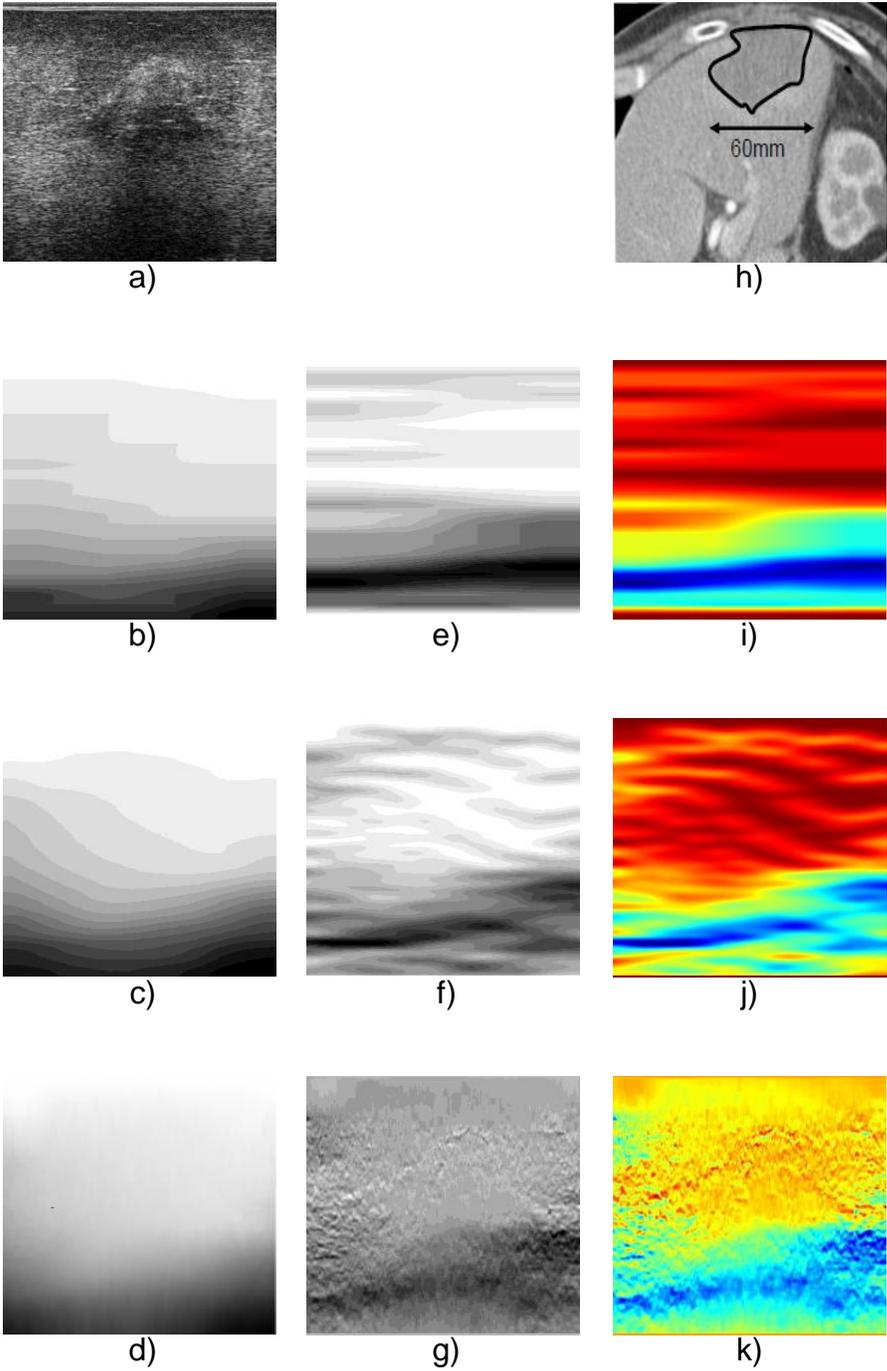
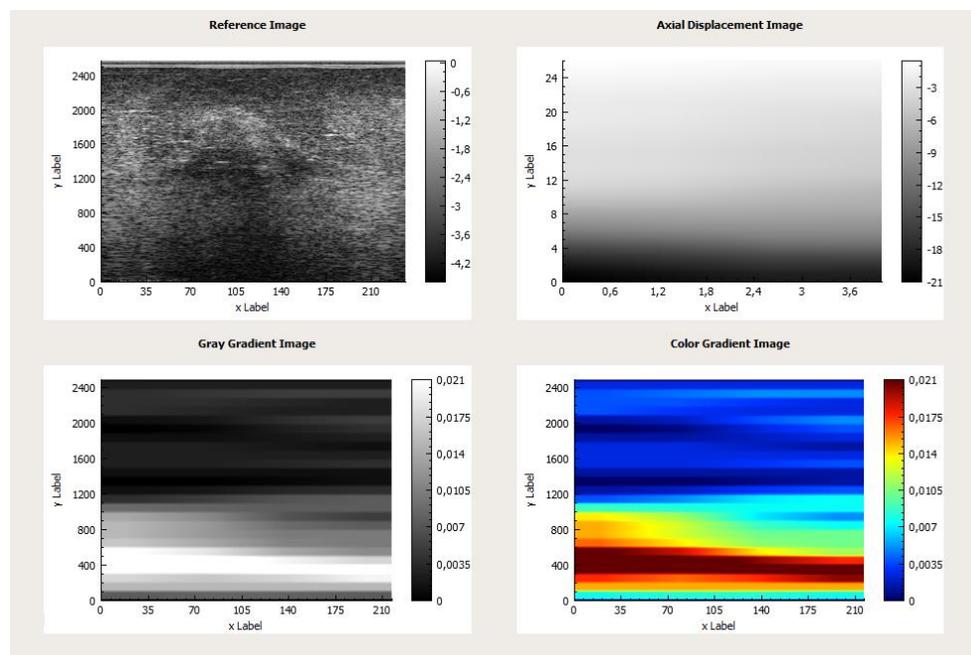


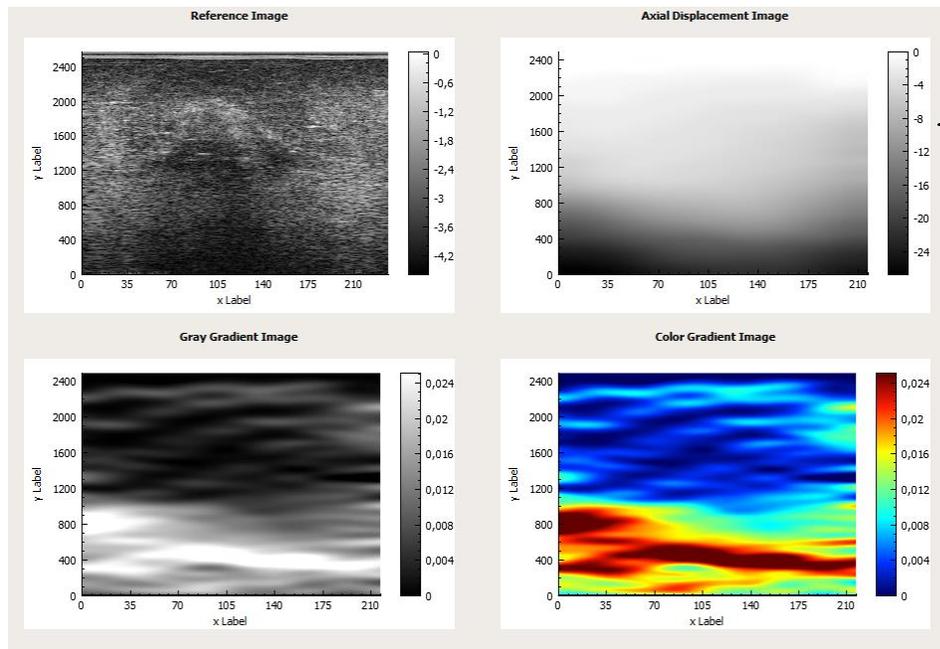
Figura 55. Resultados obtenidos Phantom de prueba sin normalización

La figura (a) es la imagen en modo B del paciente de prueba, las imágenes correspondientes a (b), (c) y (d) son los desplazamientos axiales calculados por cada uno de los algoritmos mencionados anteriormente. Las imágenes (e), (f) y (g) son las imágenes de deformación o elastogramas producto del cálculo de la deformación aplicado en Elax1; las imágenes (h), (i) y (j) son las mismas imágenes de deformación solo que se les aplica un gradiente de color que permite hacer un análisis visual sobre los tejidos examinados.

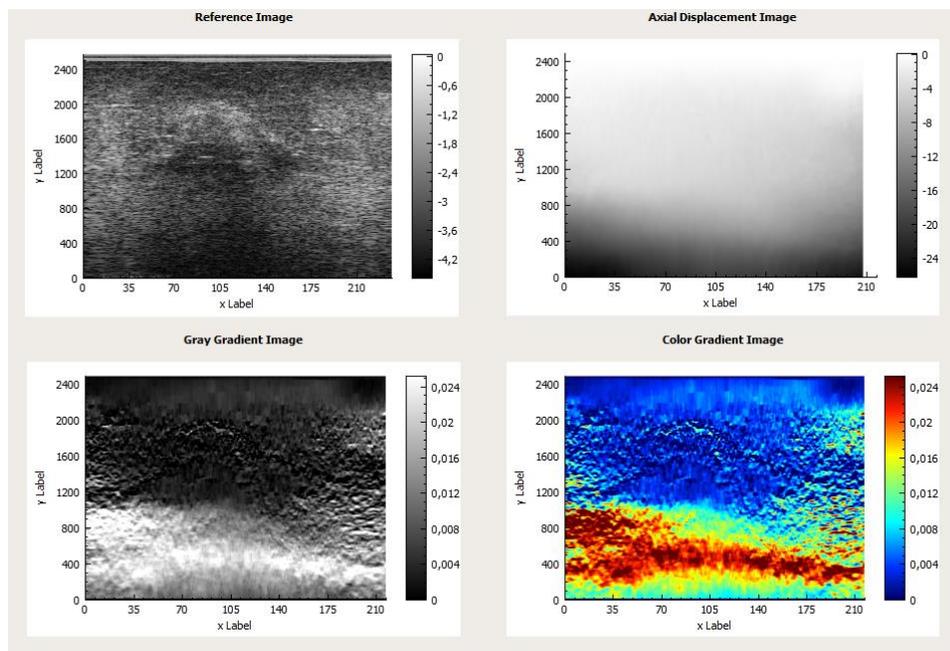
Ahora se procede a mostrar los resultados arrojados por la herramienta Elax1 como se muestra en las figuras 56, 57 y 58.



**Figura 56. Resultados de Elastografía del paciente 3 utilizando el algoritmo SSD.**



**Figura 57. Resultados de Elastografía del paciente 3 utilizando el algoritmo HBD.**

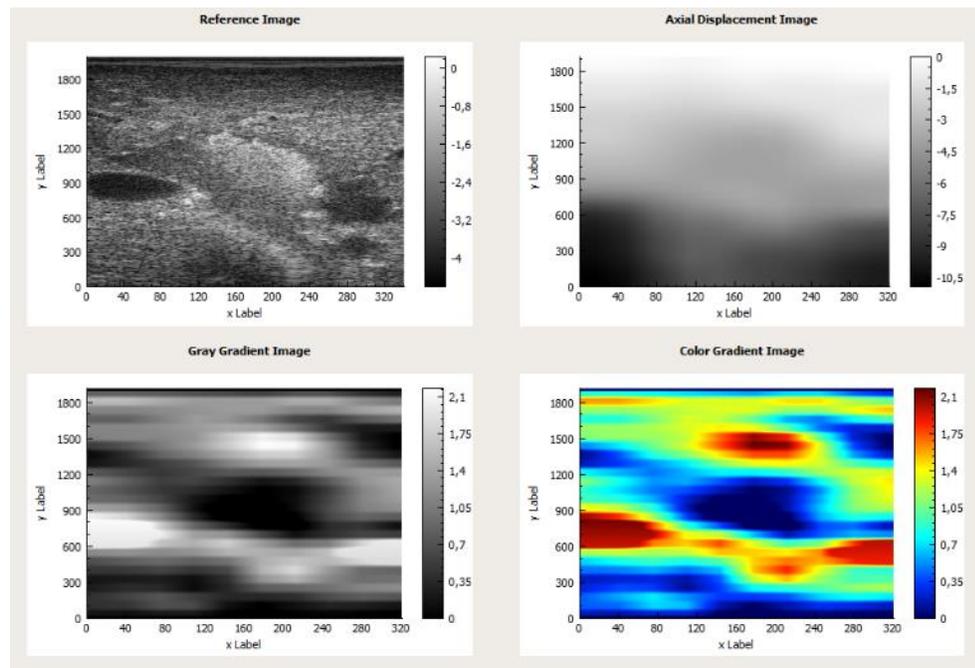


**Figura 58. Resultados de Elastografía del paciente 3 utilizando el algoritmo AM2D.**

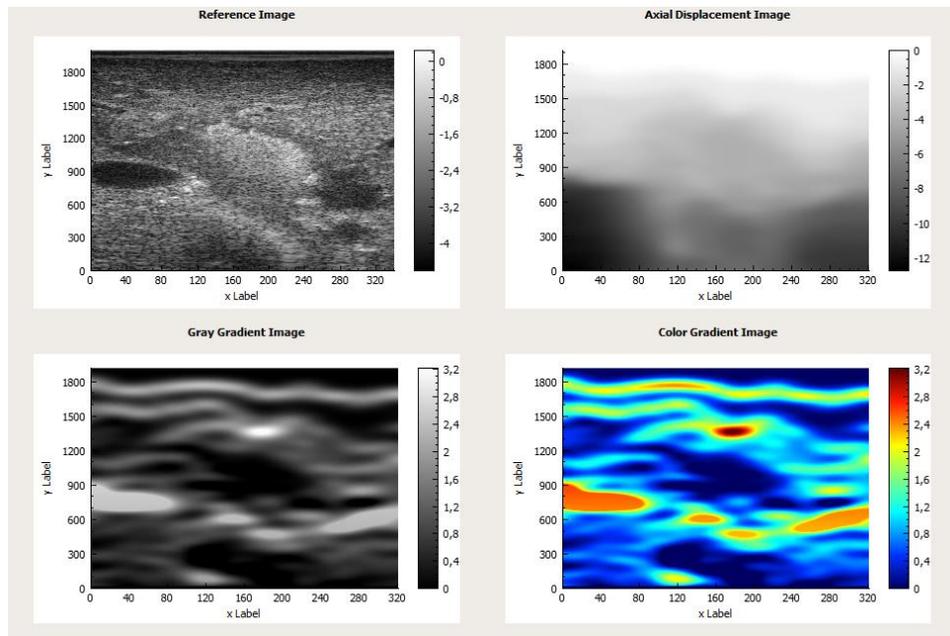
De las anteriores figuras se puede diferenciar una región rígida representada por el color azul, esta región que corresponde a la lesión térmica dejada por la terapia de ablación RF para el cáncer de hígado tiene más o menos la misma geometría de deformación en los tres métodos de elastografía utilizados, pese a que es posible detectar la lesión térmica del paciente 3 en los tres métodos de elastografía, la resolución y precisión del método SSD resulta muy deficiente, el método de elastografía HBD presenta una imagen de deformación muy similar a la imagen del método AM2D pero con una resolución menor.

Los planos de normalización no fueron utilizados para la obtención de resultados de deformación del paciente 3, debido a que las imágenes arrojadas no fueron buenas en ninguno de los métodos de elastografía. Esto se puede deber a que los valores máximos o mínimos de las imágenes de deformación se encuentran lejos de los valores medios de la imagen.

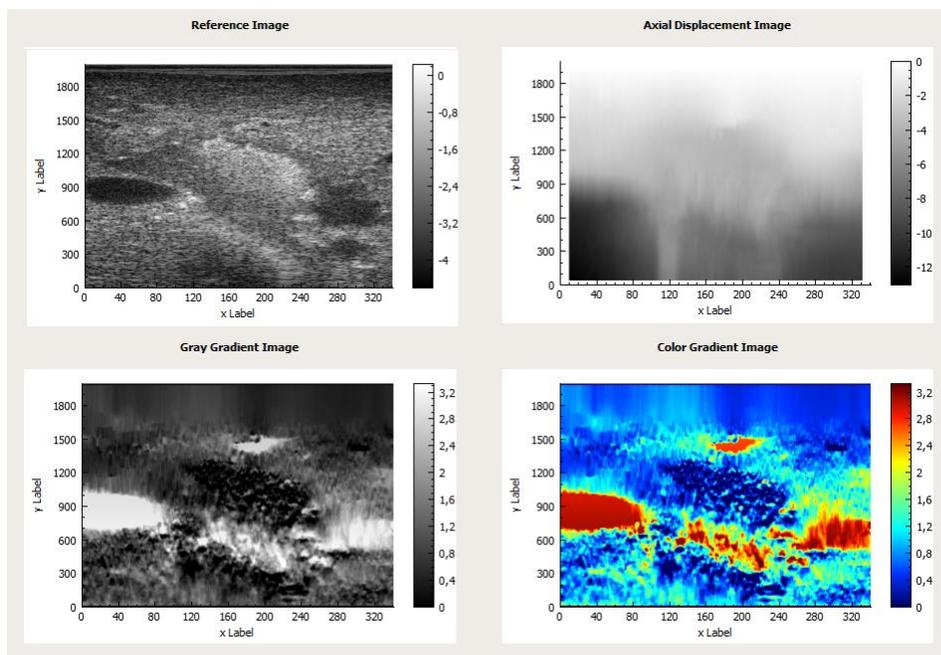
Los resultados del paciente 4 utilizando los tres métodos elastografía se pueden ver en las figuras 59, 60 y 61.



**Figura 59. Resultados de Elastografía del paciente 4 utilizando el algoritmo SSD.**



**Figura 60. Resultados de Elastografía del paciente 4 utilizando el algoritmo HBD.**



**Figura 61. Resultados de Elastografía del paciente 4 utilizando el algoritmo AM2D.**

De las anteriores figuras se puede diferenciar en el centro de las imágenes de deformación, una región rígida representada por el color azul, esta región que corresponde a la lesión térmica dejada por la terapia de ablación RF para el cáncer de hígado tiene una geometría de deformación similar en los tres métodos de elastografía utilizados, al comparar la imagen de deformación del AM2D con los otros métodos es posible apreciar ciertas diferencias, estas diferencias se pueden deber a artefactos que los métodos SSD y HBD no pudieron resolver. Al igual que en los resultados del paciente 3, se observan deficiencias de resolución y precisión del SSD comparada con los otros dos métodos.

En base a 5.1 y 5.2, se puede afirmar que en los resultados de los métodos de elastografía HBD y SSD se pudieron identificar la ubicación, geometría y valores de deformación de las regiones duras de un tejido; se pudo ver que los algoritmos son buenos tanto desde el punto de vista cualitativo y cuantitativo, más sin embargo, el método de elastografía AM2D fue superior en términos de resolución, robustez y precisión.

Cabe resaltar que los resultados obtenidos están sujetos a modificaciones que podrían contribuir con la mejora de los mismos, mediante la variación de los parámetros que recibe cada algoritmo para el cálculo de elastografía usado en este trabajo. A continuación se muestra tabla 1 con los parámetros utilizados para los resultados obtenidos.

**Tabla 1. Parámetros de configuración para los algoritmos SSD, HBD y AM2D**

<b>Métodos</b>	<b>Parámetros</b>	<b>Phantom</b>	<b>Paciente 3</b>	<b>Paciente 4</b>
AM2D	IRF: máximo rango de variación permitida en el desplazamiento axial dado en pixeles	[-100,0]	[-45,0]	[-11,0]
	IA: máximo rango de variación permitido en el desplazamiento lateral dado en pixeles	[4,4]	[0,0]	[0,0]
	Alfa_DP: peso de regularización de la programación dinámica	0.15	0.15	0.15

<b>Métodos</b>	<b>Parámetros</b>	<b>Phantom</b>	<b>Paciente 3</b>	<b>Paciente 4</b>
	MidA: semilla desde donde inicia el cálculo del desplazamiento dado en línea-RF	450	200	11
	Alfa: regularización axial	5	5	20
	Beta: regularización lateral	10	20	20
	Gamma: regularización lateral	0.005	0.005	0.005
	T: umbral para mínimos cuadrados iterados re ponderados (IRLS)	0.2	0.2	0.2
	A_t: atenuación dado en dB/cm/MHz	0.63	1	1
	F_0: frecuencia central de ultrasonido dado en MHz	8	8	8
	F_s: frecuencia de muestreo dado en MHz	40	40	40
SSD	Tv (ancho, alto): Tamaño de ventana dado en pixeles	[21,61]	[41,91]	[39,91]

Métodos	Parámetros	Phantom	Paciente 3	Paciente 4
	Sp (lateral, axial): Superposición entre ventanas dado en pixeles	[3,3]	[3,3]	[3,3]
	Cmp: Compresión máxima estimada del tejido dado en porcentaje (%)	7	1	3
HÍBRIDO	T2: umbral de la correlación cruzada normalizada en el nivel 2	0.7	0.7	0.7
	T1: umbral de la correlación cruzada normalizada en el nivel 1	0.7	0.7	0.7
	T0: umbral de la correlación cruzada normalizada en el nivel 0	0.8	0.8	0.8
	Cmp: compresión máxima estimada del tejido dado en porcentaje (%)	7	3	1
	Tav2 (arriba, abajo): aumento en tamaño del alto de la ventana con respecto al ancho en el nivel 2, dado en pixeles	[0,0]	[8, 8]	[7,7]
	Rb0 (ancho, alto): región de búsqueda en el nivel 0, dado en pixeles	[3,5]	[3,5]	[3,3]
	Tv0 (ancho, alto): Tamaño de ventana en el nivel 0, dado en pixeles	[5,5]	[7,21]	[5,11]

## 6 CONCLUSIONES Y TRABAJOS FUTUROS

### 6.1 Conclusiones

A continuación se dan a conocer las conclusiones en base a los objetivos planteados y a los resultados alcanzados en el desarrollo de este trabajo.

- En la presente investigación fue posible conocer y comprender gran variedad de los métodos y algoritmos de elastografía desarrollados a lo largo de los años, se pudieron conocer algunas de las ventajas y desventajas que los hacen de aplicación específica a ciertos campos de la medicina y a ciertas condiciones de trabajo.
- Se desarrolló e implementó la herramienta software Elax1. En ella, tres algoritmos principales se pueden utilizar: SSD, HBD y AM2D, siendo los dos primeros, algoritmos que se implementaron en este trabajo. El complemento de estos algoritmos o métodos con los algoritmos de Relación de Deformación, SNR y de Normalización, permitieron y permitirán tener mayores acercamientos a la elastografía y a los innumerables aportes que esta ofrece.
- Con Elax1, imágenes de deformación fueron obtenidas, estas imágenes permitieron determinar cualitativamente la dureza relativa de un Phantom y de los tejidos de dos pacientes de la base de datos [1]. También fue posible obtener un conocimiento semicuantitativo de la dureza relativa de un tejido, gracias a la implementación del algoritmo Relación de Deformación y un mejor análisis de la conformación elástica del tejido gracias al algoritmo SNR.
- Otro de los algoritmos de la herramienta Elax1 son los planos de normalización, estos planos permitieron el mejoramiento visual y por ende interpretativo de las imágenes de deformación.
- Tanto en 5.1 como en 5.2 se pudieron ver, en términos generales, buenos resultados de elastografía, usando los diferentes algoritmos de la herramienta software Elax1, pero fue AM2D, el algoritmo más complejo y de más resolución, debido al hecho de que no usa algoritmos de correspondencia de bloques, además fue apreciable su superioridad en términos de resolución y de precisión sobre los algoritmos SSD y HBD, mas sin embargo, fueron los tres algoritmos, capaces de identificar de forma

similar, la ubicación y geometría de las regiones duras de los tres tejidos evaluados en la sección del capítulo de resultados.

- El desarrollo de una herramienta de tipo libre permitirá al usuario tener la libertad de experimentar en la obtención de los resultados, además de tener la posibilidad de la inclusión de nuevos métodos para la obtención de elastogramas, permitiendo acercarse y dar mayor confianza a los diagnósticos obtenidos.

## **6.2 Trabajos futuros**

Se recomienda para futuras versiones de la herramienta, implementar más algoritmos de procesamiento para elastografía que permitan llegar a la confianza en el diagnóstico mediante la técnica.

Modificar el uso de la herramienta a pruebas in vivo, mediante la adquisición de datos en pacientes con lesiones en los tejidos, logrando un cálculo del elastograma en tiempo real y acercándose más al procedimiento real.

Lograr la búsqueda de métodos que mejoren la interpretación de los resultados, debido que los usuarios finales son profesionales de la salud y podría no ser muy claro leer los resultados mediante gráficos estadísticos.

## 7 REFERENCIAS BIBLIOGRAFICAS

- [1] H. Rivaz, «Ultrasound-Based Guidance for Partial Breast Irradiation Therapy,» U.S. Army Medical Research and Materiel Command, Maryland, 2012.
- [2] D. C. Dugdale, «MedlinePlus,» 9 marzo 2012. [En línea]. Available: <http://www.nlm.nih.gov/medlineplus/spanish/ency/article/001289.htm>.
- [3] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yazdi y X. Li, «Elastography: A quantitative method for imaging the elasticity of biological tissues,» *Ultrasonic Imaging*, vol. 13, nº 2, pp. 111-134, 1991.
- [4] F. Yeung, S. F. Levinson y K. J. Parker, «Multilevel and motion model-based ultrasonic speckle tracking algorithms,» *Ultrasound in Med. & Biol.*, vol. 24, nº 3, pp. 427-441, 1998.
- [5] A. Pesavento, A. Lorenz, S. Siebers y H. Ermert, «New real-time strain imaging concepts using diagnostic ultrasound,» *Physics in Medicine and Biology*, vol. 45, nº 6, pp. 1423-1435, 2000.
- [6] C. Balleyguier, L. Ciolovan, S. Ammari, S. Canale, S. Sethom, R. A. Rouhbane, P. Vielh y C. Dromain, «Breast elastography: The technical process and its applications,» *Diagnostic and Interventional Imaging*, vol. 94, nº 5, pp. 503-513, 2013.
- [7] P. Ricci, E. Maggini, E. Mancuso y P. Lodise, «Clinical application of breast elastography: State of the art,» *European Journal of Radiology*, vol. 83, nº 3, pp. 429-437, 2013.
- [8] A. Basarab, H. Liebgott, F. Morestin, D. Lyshchik, T. Higashi, R. Asato y P. Delachartre, «A method for vector displacement estimation with ultrasound imaging and its application for thyroid nodular disease,» *Análisis de Imágenes Médicas*, vol. 12, nº 3, pp. 259-274, 2008.
- [9] J. Lindop, «3D Elastography Using Ultrasound,» Cambridge, 2008.
- [10] K. Liu, P. Zhang, J. Shao, X. Zhu, Y. Zhang y J. Bai, «A 2D strain estimator with numerical optimization method for soft-tissue elastography,» *Ultrasonics*, vol. 49, nº 8, pp. 723-732,

2009.

- [11] L. Chen, R. J. Housden, G. M. Treece, A. H. Gee y R. W. Prager, «A Hybrid Displacement Estimation Method for Ultrasonic Elasticity Imaging,» *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 57, nº 4, pp. 866-882, 2010.
- [12] J. E. Lindop, G. M. Treece, A. H. Gee y R. W. Prager, «Estimation of Displacement Location for Enhanced Strain Imaging,» *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 54, nº 9, pp. 1751-1771, 2007.
- [13] K. Kaluzynski, X. Chen, S. Y. Emelianov, A. R. Skovoroda y M. O'Donnell, «Strain rate imaging using two-dimensional speckle tracking,» *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 48, nº 4, pp. 1111-1123, 2001.
- [14] M. Lysiansky, N. B. Hinenzon, H. Khamis, N. Smirin, P. Lysyansky, Z. Friedman, S. Shimoni, W. Fehske y D. Adam, «Measurements of transmural strain variations by two dimensional ultrasound speckle tracking,» *Journal of Biomedical Graphics & Computing*, vol. 2, nº 1, p. 15, 2012.
- [15] J. E. Lindop, G. M. Treece, A. H. Gee y R. W. Prager, «Phase-based ultrasonic deformation estimation,» *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 55, nº 1, pp. 94-111, 2008.
- [16] T. Shiina, M. Yamakawa, N. Nitta y E. Ueno, «Real time tissue elasticity imaging using the combined autocorrelation method,» *Journal of Medical Ultrasonics*, vol. 29, nº 3, pp. 119-128, 2002.
- [17] H. Shi y T. Varghese, «Two-dimensional multi-level strain estimation for discontinuous tissue,» *Physics in Medicine and Biology*, vol. 52, nº 2, pp. 389-401, 2007.
- [18] J. Lindop, «2D and 3D Elasticity Imaging Using Freehand Ultrasound,» Cambridge, 2004.
- [19] L. Chen, G. M. Treece, J. E. Lindop, A. H. Gee y R. W. Prager, «A quality-guided displacement tracking algorithm for ultrasonic elasticity imaging,» *Medical Image Analysis*, vol. 13, nº 2, pp. 286-296, 2009.
- [20] S. K. Alam y J. Ophir, «Reduction of signal decorrelation from mechanical compression of tissues by temporal stretching: applications to elastography,» *Ultrasound in Medicine and*

*Biology*, vol. 23, nº 1, pp. 95-105, 1997.

- [21] G. Parra, M. Fajardo, F. Vergara, I. Diaz y E. d. Nubila, «Elastografía: Un nuevo Método Diagnóstico,» *Revista Colombiana De Radiología*, vol. 17, nº 3, pp. 1982-1985, 2006.
- [22] B. Garra, E. Cespedes, J. Ophir, S. Spratt, R. Zuurbier, C. Magnant y M. Pennaen, «Elastography of breast lesions: initial clinical results,» *Radiology*, vol. 202, nº 1, pp. 79-86, 1997.
- [23] R. Barr, «Real-time ultrasound elasticity of the breast: initial clinical results,» *Ultrasound Q*, vol. 26, nº 2, pp. 61-66, 2010.
- [24] Q. Zhu, Y. Jiang, J. Liu, H. Liu, Q. Sun, Q. Dai y X. Chen, «Realtime ultrasound elastography: its potential role in assessment of breast lesions,» *Ultrasoun Med Biol*, vol. 34, nº 8, pp. 1232-1238, 2008.
- [25] A. Itoh, E. Ueno, E. Tōno, H. kamma, H. Takahashi, T. Shiina, M. Yamakawa y T. Matsumura, «Breast disease: clinical application of US elastography for diagnosis,» *Radiology*, vol. 239, nº 2, pp. 341-350, 2006.
- [26] C. A. Gaviria Lopez, «Elastografía por ultrasonido. Revisión General.,» Popayan, 2012.
- [27] «Clinica MAYO,» 13 Agosto 2013. [En línea]. Available: <http://www.mayoclinic.org/tests-procedures/mri/basics/definition/prc-20012903>.
- [28] «nano-mri,» 19 09 2011. [En línea]. Available: <https://nanomri.wordpress.com/2011/09/19/hoewerkt-mri/>. [Último acceso: 2014].
- [29] S. Parthasarathy, B. D. Bolster y F. H. Miller, «Magnetic Resonance Elastography: Proven Indications, Challenges and Future Considerations,» *Siemens Healthcare*, vol. 1, p. 8, 2013.
- [30] «Labotest,» 2012. [En línea]. Available: <http://www.labotest.cl/services/us.html>. [Último acceso: 2014].
- [31] «imgkid,» 2014. [En línea]. Available: <http://imgkid.com/gallstone-ultrasound.shtml>. [Último acceso: 2014].

- [32] «keebomed,» [En línea]. Available: <http://www.keebomed.com/>. [Último acceso: 2014].
- [33] «A.C.V,» 2012. [En línea]. Available: <http://acvexternal.com/page/14>. [Último acceso: 2014].
- [34] A. Loureiro y J. Rita, «Magnetic Resonance Elastography: Design and Implementation as a Clinical Tool,» Lisboa, 2012.
- [35] M. M. Doyle, «Model-based elastography: a survey of approaches to the inverse elasticity problem,» *PHYSICS IN MEDICINE AND BIOLOGY*, nº 57, pp. 35-73, 2012.
- [36] S. Tsuyoshi, «Ultrasound elastography: Development of novel technologies and standardization,» *Japanese Journal of Applied Physics*, vol. 53, pp. 1-7, 2014.
- [37] C. L. d. Korte, E. I. Cespedes, A. F. W. v. d. Steen, G. Pasterkamp y N. Bom, «Intravascular ultrasound elastography: assessment and imaging of elastic properties of diseased arteries and vulnerable plaque1,» *European Journal of Ultrasound*, vol. 7, nº 3, pp. 219-224, 1998.
- [38] C. Fleury, E. d. Faria, V. Fleury, J. Carlos, S. Piato y D. Roveda Jr, «New elastographic classification of breast lesions during and after compression,» *Diagnostic and Interventional Radiology*, vol. 15, pp. 96-103, 2019.
- [39] L. Leong, L. Sim, Y. Lee, F. Ng, C. Wan, S. Fook-Chong, A. Jara-Lazaro y P. Tan, «A prospective study to compare the diagnostic performance of breast elastography versus conventional breast ultrasound,» *Clinical Radiology*, vol. 65, nº 11, pp. 887-894, 2010.
- [40] R. Moan, «Pros and cons of state-of-the-art breast elastography,» 10 07 2013. [En línea]. Available: <http://www.auntminnieeurope.com/index.aspx?sec=ser&sub=def&pag=dis&ItemID=608428>.
- [41] F. A, W. S y D. F, «Diagnostic value of strain ratiomeasurement in the differentiation of malignant and benign breast lesions.,» *Ultraschall in der Medizin*, vol. 32, pp. 400-405, 2010.
- [42] X. X. Y. H. e. a. Zhi H, «Ultrasonic elastography in breast can-cer diagnosis: strain ratio vs 5-point scale,» *Academic Radiology*, vol. 17, pp. 1227-1233, 2010.
- [43] D. F. F. A. e. a. Thomas A, «Significant differentiation of focalbreast lesions: calculation of strain ratio in breast sonoelastography.,» *Academic Radiology*, vol. 17, nº 5, pp. 558-563,

2010.

- [44] P. U. F. S. e. a. Fischer T, «Significant differentiation of focal breast lesions: raw data-based calculation of strain ratio.,» *Ultraschall in der Medizin*, vol. 33, nº 4, pp. 372-379, 2012.
- [45] Y. T. K. T. e. a. Yerli H, «Qualitative and semiquantitative evaluation of solid breast lesions by sonoelastography.,» *Journal of Ultrasound in Medicine*, vol. 30, pp. 179-186, 2011.
- [46] R. DM, H. GK y H. NJ, «Breast lesions: evaluation with US strain imaging-clinical experience of multiple observers.,» *Radiology*, vol. 238, pp. 425-437, 2006.
- [47] R. Barr, «Real-time ultrasound elasticity of the breast: initial clinical results.,» *Ultrasound Quarterly*, vol. 26, pp. 61-66, 2010.
- [48] J. Mari y C. Cachard, «Acquire real-time rf digital ultrasound data from a commercial scanner.,» *Technical Acoustics*, 2007.
- [49] R. Souchon, «Books Google,» 2004. [En línea]. Available: <http://books.google.es/books?id=eT6NZwEACAAJ>.
- [50] Y. Zhu y T. J. Hall, «A Modified Block Matching Method for Real-Time Freehand Strain Imaging,» *Ultrason Imaging*, vol. 24, nº 3, pp. 161-176, 2002.
- [51] G. Treece, J. Lindop, A. Gee y R. Prager, «Uniform precision ultrasound strain imaging,» Cambridge, 2009.
- [52] J. Bai, C. Ding y Y. Fan, «A multi-scale algorithm for ultrasonic strain reconstruction under moderate compression.,» *Ultrasonics*, vol. 37, pp. 511-519, 1999.
- [53] M. Bilgen y M. F. Insana, «Deformation models and correlation analysis in elastography.,» *The Journal of the Acoustical Society of America*, vol. 99, nº 5, p. 3212–3224, 1996.
- [54] «Qt Project,» [En línea]. Available: [http://qt-project.org/resources/getting\\_started](http://qt-project.org/resources/getting_started) . [Último acceso: 2014].

- [55] F. Sermal, «OpenCV Tutorial C++,» [En línea]. Available: <http://opencv-srf.blogspot.com/2010/09/what-is-opencv.html>. [Último acceso: 2014].
- [56] «MatLab,» [En línea]. Available: [http://es.mathworks.com/products/compiler/mcr/index.html?s\\_tid=gn\\_loc\\_drop](http://es.mathworks.com/products/compiler/mcr/index.html?s_tid=gn_loc_drop). [Último acceso: 2014].
- [57] «Microsoft,» [En línea]. Available: [http://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio). [Último acceso: 2014].
- [58] D. E. Guzmán Villamarín, «HERRAMIENTA SOFTWARE PARA LA PRÁCTICA Y EXPERIMENTACIÓN DE LA ROBOTICA QUIRUGICA,» Popayán, 2013.
- [59] P. Webb, «MATLAB Central,» Mathworks blog, 2 Febrero 2011. [En línea]. Available: <http://blogs.mathworks.com/loren/2011/02/03/creating-c-shared-libraries-and-dlls/>. [Último acceso: 2014].