

## Anexo A

# Set Partitioning In Hierarchical Trees (SPIHT)

El algoritmo SPIHT fue introducido por primera vez por Said y Pearlman [3] para la compresión de imágenes. Este algoritmo se puede ver como una extensión del algoritmo Embedded Zero-tree Wavelet (EZW) diseñado por J. M. Shapiro [14]. El algoritmo EZW supuso un importante avance en la codificación de imágenes dado que rompía con la tónica general de incremento en la eficiencia obtenida gracias a un incremento en la complejidad. Esta técnica no solamente era competitiva en cuanto a prestaciones de compresión comparada con la técnicas más complejas de compresión de imagen sino que además, era extremadamente rápida en cuanto a su ejecución y producía un flujo de bits embebido. Esto es muy importante dado que gracias a esta propiedad la codificación se podía parar en cualquier instante, controlándose de forma precisa la cantidad de información que se utilizaba en la codificación de la imagen. Siguiendo este trabajo, A. Said y W. A. Pearlman introdujeron el algoritmo SPIHT para la compresión de imágenes. El algoritmo SPIHT se aplica directamente a los coeficientes obtenidos realizando la transformada wavelet a la imagen.

Los principios de SPIHT son el ordenamiento parcial de los coeficientes transformados por magnitud con un algoritmo de ordenamiento basado en la partición en conjuntos, transmisión ordenada del plano de bits y explotación de la autosimilitud entre los diferentes niveles de la transformada. Siguiendo estos principios, el codificador siempre envía los bits más significativos al decodificador.

### A.1 Transformada Wavelet

La aplicación de la transformada wavelet permite pasar las muestras en el dominio del tiempo a un dominio transformado donde resulta más eficiente su codificación. De esta forma se consigue una gran eficiencia en cuanto a operaciones y tiempo de ejecución.

Las Wavelets, funciones bases de la Transformada Wavelet, son generadas a partir

de una función Wavelet madre, mediante traslaciones y dilataciones. Estas funciones permiten reconstruir la señal original a través de la Transformada Wavelet inversa. La Transformada Wavelet no es solamente local en tiempo, sino también en frecuencia. Dentro de los usos de esta poderosa herramienta podemos nombrar, además del análisis local de señales no estacionarias, el análisis de señales electrocardiográficas, sísmicas, de sonido, de radar, así como también es utilizada para la compresión y procesamiento de imágenes y reconocimiento de patrones.

En una imagen cualquiera, existe una relación directa entre los colores que esta presenta en cada pixel y la distribución del espectro en frecuencia. De esta manera, se sabe que las variaciones suaves en el color se corresponden con variaciones de baja frecuencia y las variaciones fuertes de color se corresponden con variaciones de alta frecuencia. La transformación del espacio de intensidades de color al espacio frecuencial se puede llevar a cabo de muchos modos. Uno de los posibles es la DCT, otra de las posibles técnicas es la descomposición de una imagen por medio de la Transformada Discreta Wavelet o Discrete Wavelet Transform (DWT).

La forma general de una transformada wavelet unidimensional (1-D) se puede apreciar en la Figura A.1

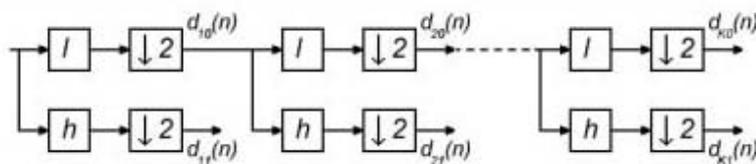


Figura A.1: Descomposición de una wavelet de 1-D, de K niveles.

Aquí una señal pasa a través de un filtro pasa bajos y de otro pasa altos, denotados por l y h respectivamente, y luego se submuestra en un factor de dos - esto causa un proceso con pérdida-, logrando una transformada de un nivel. Se pueden realizar múltiples niveles repitiendo los procesos de filtrado y eliminación (submuestreo), únicamente sobre la rama del filtro pasa bajos. Este proceso se lleva a cabo para un número K finito de niveles, obteniendo los coeficientes wavelet:  $d_{i1}(n)$ ,  $i = \dots, 1, \dots, K$  y  $d_{K0}(n)$ . Cuando no se necesita tener conocimiento de la escala o de la frecuencia, el conjunto entero de coeficientes wavelet se expresa como  $w(n)$ . La transformada wavelet de 1-D puede extenderse a una transformada wavelet bidimensional (2-D) utilizando filtros wavelet separables. Con los filtros separables, la transformada 2-D puede calcularse aplicando una transformada 1-D a cada fila de los datos de entrada, y luego aplicándola a cada columna de los datos de entrada.

Así, como la transformada wavelet se utiliza para separar una imagen en varias clases de importancia, la transformada inversa o antitransformada se utiliza para recomponer las distintas clases de datos, y de este modo poder reconstruir la imagen. Aquí, también se utiliza un filtro pasa bajos y otro pasa altos, pero el filtrado se realiza en forma

opuesta. Es decir, se comienza desde el nivel más alto, aplicando primero los filtros por columna y luego por fila, hasta llegar al primer nivel.

La DWT aplicada a imágenes proporciona una matriz de coeficientes, conocidos como coeficientes wavelet. Si a una imagen le aplicamos la DWT obtenemos cuatro tipos de coeficientes (ver Figura A.2): aproximaciones (LL), detalles horizontales (HL), detalles verticales (LH) y detalles diagonales (HH). La aproximación (LL) contiene la mayor parte de la energía de la imagen, es decir, la información más importante, mientras que los detalles tienen valores próximos a cero.

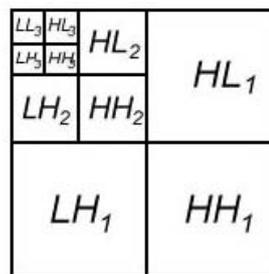


Figura A.2: Esquema de la organización de los coeficientes wavelet.

La elección de las wavelets analizadoras juega un papel muy importante en los resultados finales. Entre las características más importantes a tener en cuenta se encuentran: soporte compacto, simetría, etc. Las wavelets biortogonales son las más eficientes para un posterior proceso de compresión, y en particular, aquellas con pocos coeficientes, ya que el coste de obtención de los coeficientes se incrementa con su número.

Generalmente, la energía de las imágenes se concentra en las frecuencias bajas. Una imagen tiene un espectro que se reduce con el incremento de las frecuencias. Estas propiedades de las imágenes quedan reflejadas en la transformada wavelet discreta de la imagen. Los niveles más bajos de compresión se corresponden con las bandas de alta frecuencia. En particular, el primer nivel representa la banda de más alta frecuencia y el nivel más fino de resolución. A la inversa, el último nivel ( $n$ ) de descomposición corresponde con la banda de frecuencia más baja y la resolución más tosca. Así, al desplazarse de los niveles más altos a los más bajos, o sea, de baja resolución a alta resolución, se observa una disminución de la energía contenida en las subbandas recorridas.

Si los coeficientes wavelet obtenidos por medio de la transformada wavelet discreta (DWT) para un nivel concreto poseen pequeñas magnitudes (valores próximos a cero), se espera que esos coeficientes wavelet estén en los primeros niveles de descomposición. El aumento del nivel de descomposición wavelet produce unos coeficientes con mayores magnitudes. Adicionalmente, se puede comprobar cómo existen similitudes espaciales a través de las subbandas.

En la Figura A.3 se puede observar los contornos de Bárbara (base de imágenes para



(a) Imagen original de Bárbara.

(b) Organización de los coeficientes wavelet

*Figura A.3: Ejemplo de la transformada wavelet.*

investigación científica) en los distintos niveles y cómo son más bastos en el primer nivel de descomposición, además de cierta similitud entre los distintos niveles. El cálculo de la transformada wavelet para todas las posibles escalas supone una gran cantidad de información. Escoger solo aquellas escalas y posiciones que resulten interesantes para ciertos estudios es una tarea difícil. Si se escogen aquellas escalas y posiciones basadas en potencias de dos, los resultados serán más eficaces. Este análisis se denomina DWT.

Para muchas señales la información más importante se encuentra en las frecuencias bajas, mientras que en las altas frecuencias se encuentran los detalles o matices de la señal. Por ejemplo, en el caso de la voz humana, si eliminamos los componentes con altas frecuencias, la voz suena diferente pero se sigue entendiendo su mensaje. En cambio, si lo que se elimina son las componentes de bajas frecuencias, el mensaje se vuelve irreconocible. Por eso el análisis wavelet permite descomponer la señal en aproximaciones y detalles, a éste proceso se le conoce con el nombre de análisis. Este filtrado nos proporciona el doble de datos de los que son necesarios, este problema se soluciona con la operación de downsampling. La transformada wavelet discreta (DWT) se emplea para obtener una nueva representación de la imagen, más apropiada para el proceso de compresión.

## A.2 Algoritmo de ordenamiento de partición de grupos

Las mismas reglas de partición se definen tanto en el codificador como en el decodificador. Se dice que un conjunto de coeficientes subbanda  $c_i$  en el subconjunto  $T$  es significativo para una profundidad de bit  $n$  si  $\max_{i \in T} \{|c_i|\} \geq 2^n$ , en caso contrario se dice que es no significativo. Si el subconjunto es no significativo, se manda un cero

al decodificador. Si es significativo, se manda un uno al decodificador y después el subconjunto es dividido de acuerdo con el árbol de ordenamiento (ver Figura A.4) hasta que todos los grupos significativos son únicamente puntos significativos. En esta fase de la comunicación, llamada etapa de ordenamiento, los índices de los coeficientes son distribuidos dentro de tres listas, la lista de puntos no significativos (LIP), la lista de grupos no significativos (LIS) y la lista de puntos significativos (LSP). En esta etapa, sólo los bits relacionados con las entradas en LSP y las salidas binarias de los tests de magnitud son transmitidos al decodificador.

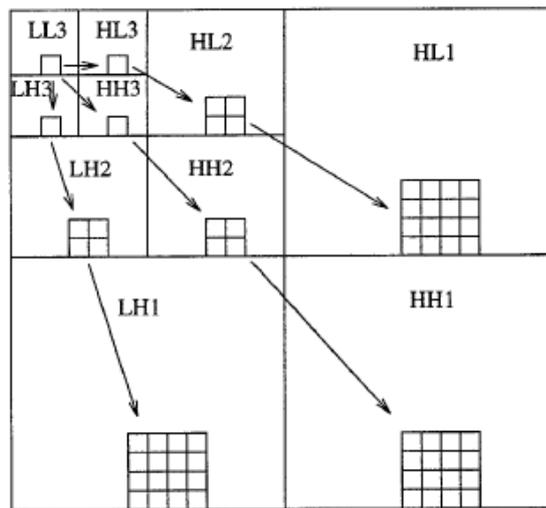


Figura A.4: Coeficientes wavelet organizados en árboles jerárquicos.

### A.3 Etapa de refinado

Después de la etapa de ordenamiento, se toman los coeficientes significativos para el umbral  $2^n$  y se envía al decodificador el  $n$ -ésimo bit más significativo de cada uno de los coeficientes que se encontraron relevantes con un umbral superior. Transmitiendo el flujo de bits con este ordenamiento en el plano de bits, siempre se transmiten los bits más significativos que quedan por transmitir. El boceto del proceso completo de codificación sería el siguiente:

1. Inicialización: LSP está vacía. Establecimiento de las raíces de los árboles de nivel superior en LIP y en LIS. Establecimiento del umbral  $2^n$  con  $n = \lfloor \log_2 (\max_i |c_i|) \rfloor$ .
2. Etapa de ordenamiento: Usando el algoritmo de ordenamiento de partición en grupos, distribuir los índices de los coeficientes en las listas LIP, LIS y LSP de forma apropiada.
3. Etapa de refinado: Para cada entrada en LSP que haya sido significativa para umbrales mayores, mandar el  $n$ -ésimo bit más significativo al decodificador.

4. Decrementar  $n$  en uno y volver al paso 2 hasta que el objetivo que se desee cumplir (un determinado error o una tasa de transmisión determinada) se alcance.

## A.4 Ejemplo del funcionamiento del algoritmo

Se va a considerar un ejemplo simple para mostrar y así entender mejor el funcionamiento del algoritmo. Se va a aplicar un paso del algoritmo a la descomposición wavelet en 3 niveles de una imagen de  $8 \times 8$  muestras produce 64 coeficientes distribuidos que se muestra en la Figura A.5. Todo el procedimiento de codificación se muestra en la Tabla A.1. A continuación se exponen algunas definiciones y explicaciones importantes para interpretar la Tabla A.1.

	0	1	2	3	4	5	6	7
0	63	-34	49	10	7	13	-12	7
1	-31	23	14	-13	3	4	6	-1
2	15	14	3	-12	5	-7	3	9
3	-9	-7	-14	8	4	-2	3	2
4	-5	9	-1	47	4	6	-2	2
5	3	0	-3	2	3	-2	0	4
6	2	-3	6	-4	3	6	3	6
7	5	11	5	6	0	3	-4	4

Figura A.5: Coeficientes wavelet de una imagen utilizada como ejemplo. Los números fuera de la caja indican las coordenadas usadas.

**LIS** Contiene grupos de coeficientes de la transformada wavelet que son definidos por las estructuras de árbol, y que se ha encontrado que tienen una magnitud más pequeña que un umbral (son no significativos). Los grupos son designados por el coeficiente raíz pero lo excluyen. Estos árboles tienen al menos dos elementos.

**LIP** Contiene coeficientes individuales cuya magnitud es más pequeña que el umbral.

**LSP** Contiene los coeficientes con magnitud mayor que el umbral (los coeficientes significativos).

**O(i,j)** En las estructuras de árbol, es el conjunto de descendientes directos de un nodo definido por la localización (i,j).

**D(i,j)** Conjunto de descendientes del nodo definido por la localización (i,j).

$L(i,j)$  Conjunto definido por  $L(i,j) = D(i,j) - O(i,j)$ .

Las siguientes referencias se refieren a las entradas en la Tabla A.1 :

- (1) Tenemos las condiciones iniciales de SPIHT. El coeficiente de mayor magnitud es 63, así que el umbral se establece en 32. La notación  $(i,j)_A$  o  $(i,j)_B$  indica que la entrada a LIS es de tipo 'A' o 'B' respectivamente. Notar la duplicación de los coeficientes en las listas. El coeficiente  $(0,0)$  no es considerado raíz.
- (2) SPIHT empieza codificando la importancia de los píxeles individuales de LIP (1 cuando es significativo y 0 cuando no lo es). Cuando un bit es significativo se mueve a LSP y su signo es también codificado. Se usa la notación  $1+$  y  $1-$  para indicar cuando un bit significativo está seguido de un bit signo.
- (3) Después de comprobar los píxeles se empieza con los conjuntos, siguiendo las entradas en LIS (las entradas activas están en negrita). En este ejemplo  $D(0,1)$  es el conjunto de 20 coeficientes  $\{(0,2), (0,3), (1,2), (1,3), (0,4), (0,5), (0,6), (0,7), (1,4), (1,5), (1,6), (1,7), (2,4), (2,5), (2,6), (2,7), (3,4), (3,5), (3,6), (3,7)\}$ . Al ser  $D(0,1)$  significativo, los siguientes en comprobarse son sus cuatro hijos  $\{(0,2), (0,3), (1,2), (1,3)\}$ .
- (4) Después de que todos los hijos sean comprobados,  $(0,1)$  es movido al final de LIS y el tipo cambia de 'A' a 'B', es decir, que la nueva entrada de LIS cambia de  $D(0,1)$  a  $L(0,1)$ .
- (5) Se aplica el mismo procedimiento que en (3) y (4) al conjunto  $D(1,0)$ . Notar que aunque el hijo de  $(0,1)$  es no significante,  $D(0,1)$  es significativo porque  $L(0,1)$  es significativo.
- (6) Desde que  $D(1,1)$  es significativo, no se necesita tomar ninguna acción. El algoritmo mueve el próximo elemento a LIS.
- (7) El próximo elemento de LIS,  $(0,1)$ , es de tipo 'B', y por eso se comprueba  $L(0,1)$ . Notar que la coordenada  $(0,1)$  se mueve del principio de LIS en este paso. Esta es ahora comprobada otra vez, pero con otra interpretación del algoritmo.
- (8) Lo mismo que arriba, pero  $L(0,1)$  es significante, así que el conjunto es dividido en  $D(2, 0)$ ,  $D(2, 1)$ ,  $D(3, 0)$ , y  $D(3, 1)$ , y las correspondientes entradas son añadidas a LIS. Al mismo tiempo, la entrada  $(1,0)_B$  es movida de LIS.
- (9) El algoritmo permanece evaluando los conjuntos de entradas tal como aparecen en LIS.
- (10) Cada nueva entrada es tratada como en los casos anteriores. En este caso se comprueba el hijo de  $(2,1)$ .

Paso	Pixel o Grupo probado	Bit de Salida	Acción	Lista de Control
(1)				LIS = $\{(0,1)A,(1,0)A,(1,1)A\}$ LIP = $\{(0,0),(0,1),(1,0),(1,1)\}$ LSP = 0.
(2)	(0, 0)	1+	(0,0) to LSP	LIP = $\{(0,1),(1,0),(1,1)\}$ LSP = $\{(0,0)\}$
	(0, 1)	1-	(0,1) to LSP	LIP = $\{(1,0),(1,1)\}$ LSP = $\{(0,0),(0,1)\}$
	(1, 0)	0	none	
	(1, 1)	0	none	
(3)	D(0, 1)	1	test offspring	LIS = $\{(\mathbf{0},1)A,(1,0)A,(1,1)A\}$
	(0, 2)	1+	(0,2) to LSP	LSP = $\{(0,0),(0,1),(0,2)\}$
	(0, 3)	0	(0,3) to LIP	LIP = $\{(1,0),(1,1),(0,3)\}$
	(1, 2)	0	(1,2) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2)\}$
	(1, 3)	0	(1,3) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3)\}$
(4)			type changes	LIS = $\{(1,0)A,(1,1)A,(0,1)B\}$
(5)	D(1, 0)	1	test offspring	LIS = $\{(\mathbf{1},0)A,(1,1)A,(0,1)B\}$
	(2, 0)	0	(2,0) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0)\}$
	(2, 1)	0	(2,1) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1)\}$
	(3, 0)	0	(3,0) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0)\}$
	(3, 1)	0	(3,1) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1)\}$
			type changes	LIS = $\{(1,1)A,(0,1)B,(1,0)B\}$
(6)	D(1, 1)	0	none	LIS = $\{(\mathbf{1},1)A,(0,1)B,(1,0)B\}$
(7)	L(0, 1)	0	none	LIS = $\{(1,1)A,(\mathbf{0},1)B,(1,0)B\}$
(8)	L(1, 0)	1	add new sets	LIS = $\{(1,1)A,(0,1)B,(2,0)A,(2,1)A,(3,0)A,(3,1)A\}$
(9)	D(2, 0)	0	none	LIS = $\{(1,1)A,(0,1)B,(\mathbf{2},0)A,(2,1)A,(3,0)A,(3,1)A\}$
(10)	D(2, 1)	1	test offspring	LIS = $\{(1,1)A,(0,1)B,(2,0)A,(\mathbf{2},1)A,(3,0)A,(3,1)A\}$
	(4, 2)	0	(4,2) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2)\}$
	(4, 3)	1+	(4,3) to LSP	LSP = $\{(0,0),(0,1),(0,2),(4,3)\}$
	(5, 2)	0	(5,2) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2)\}$
	(5, 3)	0	(5,3) to LIP	LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2),(5,3)\}$
(11)	(2,1)		removed	LIS = $\{(1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A\}$
(12)	D(3, 0)	0	none	LIS = $\{(1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A\}$
	D(3, 1)	0	none	LIS = $\{(1,1)A,(0,1)B,(2,0)A,(\mathbf{3},0)A,(3,1)A\}$
(13)				LIS = $\{(1,1)A,(0,1)B,(2,0)A,(3,0)A,(\mathbf{3},1)A\}$ LIP = $\{(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2),(5,3)\}$ LSP = $\{(0,0),(0,1),(0,2),(4,3)\}$

Tabla A.1: Ejemplo de codificación SPIHT.

- (11) En este caso, porque  $L(2,1) = 0$  (sólo tiene descendientes hijos), la entrada  $(2,1)A$  es movida de LIS.
- (12) Finalmente, las dos últimas entradas de LIS corresponden a un conjunto insignificante, y no se toma ninguna acción. La tapa de ordenamiento termina después de que la última entrada de LIS sea comprobada.
- (13) La lista final en etapa de ordenamiento forma la lista inicial en el siguiente etapa de ordenamiento, con un umbral de 16.

Sin utilizar ningún otro forma de codificación, SPIHT usa 29 bits en este primer paso.



## Anexo B

# DICOM

DICOM *Digital Imaging and Commucation in Medicine* es el estándar reconocido mundialmente para el intercambio de imágenes médicas, pensado para el manejo, almacenamiento, impresión y transmisión de imágenes médicas. Ha sido adoptado por una amplia mayoría de fabricantes de equipo médico como formato para almacenamiento y transmisión de las pruebas.

Todas las imágenes DICOM deben incluir una cabecera o campo *metadata*, este campo es necesario para interpretar de forma correcta la prueba médica y recuperarla con éxito el contenedor DICOM, además de la información para la correcta interpretación de la prueba médica, la cabecera suele incluir información relativa al paciente y las condiciones en las que se realiza la prueba.

En nuestro caso nos centraremos en el estudio de pruebas ecocardiográficas de distintos fabricantes. Los campos y el tamaño de las cabeceras por lo general no es constante para todos los aparatos, pero si existirán unos campos que aparecerán siempre y que necesitaremos para la extracción de la ecocardiografía en el caso de la nueva compresión. Estos campos son:

- **FileMetaInformationGroupLength:** Tamaño en Bytes de la cabecera DICOM de la prueba.
- **Width:** número de columnas de la imagen.
- **Height:** número de filas de la imagen.

A continuación mostraremos algunos de los principales campo de cabeceras DICOM para los distintos aparatos. Philips Envisor: tabla B.1, Siemmens SC2000: tabla B.2, Siemmens Acuson Sequoia: tabla B.3, Agilent Sonos: tabal B.4.

Tabla B.1: Cabecera DICOM para una ecocardiografía de un aparato Philips Envisor

CAMPO	CONTENIDO
Filename	[1x64 char]
FileModDate	'20-ene-2012 12
FileSize	454594
Format	'DICOM'
FormatVersion	3
Width	800
Height	564
BitDepth	8
ColorType	'indexed'
ImplementationVersionName	'OFFIS_DCMBP_354'
SpecificCharacterSet	'ISO_IR 100'
StudyDate	'20101223'
AcquisitionDate	'20101223'
ContentDate	'20101223'
Modality	'US'
Manufacturer	'Philips Medical Systems'
InstitutionName	'HOSPITAL CLINICO'
ReferringPhysicianName	[1x1 struct]
PatientName	[1x1 struct]
PatientID	'10-12-23-103454'
PatientBirthDate	"
PatientSex	"
HeartRate	79
PatientOrientation	"
SamplesPerPixel	1
PhotometricInterpretation	'PALETTE COLOR'
Rows	564
Columns	800
RedPaletteColorLookupTableDescriptor	[3x1 uint16]
GreenPaletteColorLookupTableDescriptor	[3x1 uint16]
BluePaletteColorLookupTableDescriptor	[3x1 uint16]
RedPaletteColorLookupTableData	[256x1 uint16]
GreenPaletteColorLookupTableData	[256x1 uint16]
BluePaletteColorLookupTableData	[256x1 uint16]
LossyImageCompression	'00'
Private_0031_60xx_Creator	'Philips US Imaging 60'

Tabla B.2: Cabecera DICOM para una ecocardiografía de un aparato Siemens SC2000

CAMPOS	CONTENIDO
Filename	[1x65 char]
FileModDate	'12-nov-2010 12
FileSize	2361370
Format	'DICOM'
FormatVersion	3
Width	1024
Height	768
BitDepth	8
ColorType	'truecolor'
FileMetaInformationGroupLength	204
FileMetaInformationVersion	[2x1 uint8]
ImplementationVersionName	'OFFIS_DCMBP_354'
SpecificCharacterSet	'ISO_IR 100'
StudyDate	'20101001'
AcquisitionDate	'20101001'
ContentDate	'20101001'
Modality	'US'
Manufacturer	'SIEMENS'
InstitutionName	'HC LOZANO BLESA'
ReferringPhysicianName	[1x1 struct]
StationName	'SC2000SC2'
StudyDescription	'Cardiaco'
ManufacturerModelName	'ACUSON SC2000'
PatientName	[1x1 struct]
PatientID	'101001092851781'
PatientBirthDate	'
PatientSex	'O'
HeartRate	88
TransducerData	'4V1c\\'
SequenceOfUltrasoundRegions	[1x1 struct]
TransducerType	'VECTOR_PHASED'
StudyID	'a6d4dc06-3e2d-41'
PatientOrientation	'
SamplesPerPixel	3
PhotometricInterpretation	'RGB'
PlanarConfiguration	0
Rows	768
Columns	1024
BitsAllocated	8
BitsStored	8
LossyImageCompression	'00'
Private_0149_10xx_Creator	'SIEMENS Ultrasound SC2000'

Tabla B.3: Cabecera DICOM para una ecocardiografía de un aparato Siemens Acuson Sequoia

CAMPOS	CONTENIDO
Filename	[1x63 char]
FileModDate	'05-may-2012 11
FileSize	147412
Format	'DICOM'
FormatVersion	3
Width	576
Height	456
BitDepth	8
ColorType	'truecolor'
FileMetaInformationGroupLength	256
PrivateInformation	[12x1 uint8]
IdentifyingGroupLength	416
SpecificCharacterSet	'ISO_IR 100'
StudyDate	'19990219'
ContentDate	'19990219'
Modality	'US'
Manufacturer	'ACUSON'
ReferringPhysicianName	[1x1 struct]
PatientName	[1x1 struct]
PatientID	'AEGIS DEMO'
PatientBirthDate	"
PatientSex	'O'
AcquisitionGroupLength	910
DeviceSerialNumber	'51445'
HeartRate	60
SequenceOfUltrasoundRegions	[1x1 struct]
TransducerType	'VECTOR_PHASED'
RelationshipGroupLength	142
SeriesNumber	1
PatientOrientation	"
ImagePresentationGroupLength	106
SamplesPerPixel	3
PhotometricInterpretation	'YBR_FULL'
Rows	456
Columns	576
BitsAllocated	8
BitsStored	8
LossyImageCompression	'00'
Private_7fdf_GroupLength	108
Private_7fdf_10xx_Creator	'ACUSON'

Tabla B.4: Cabecera DICOM para una ecocardiografía de un aparato Agilent Sonos

CAMPO	CONTENIDO
Filename	[1x64 char]
FileModDate	'05-may-2012 11
FileSize	114984
Format	'DICOM'
FormatVersion	3
Width	600
Height	430
BitDepth	8
ColorType	'indexed'
FileMetaInformationGroupLength	138
StudyDate	'19931129'
StudyTime	'123656.0'
Modality	'US'
Manufacturer	'ACME Products'
ReferringPhysicianName	[1x1 struct]
StudyDescription	'Echocardiogram'
PatientGroupLength	70
PatientName	[1x1 struct]
PatientID	'123-45-6789'
PatientBirthDate	'19231016'
PatientSex	'F'
RelationshipGroupLength	120
PatientOrientation	"
ImageComments	'SINUS VAL ANEUR'
ImagePresentationGroupLength	1732
SamplesPerPixel	1
PhotometricInterpretation	'PALETTE COLOR'
Rows	430
Columns	600
RedPaletteColorLookupTableDescriptor	[3x1 uint16]
GreenPaletteColorLookupTableDescriptor	[3x1 uint16]
BluePaletteColorLookupTableDescriptor	[3x1 uint16]
RedPaletteColorLookupTableData	[256x1 uint16]
GreenPaletteColorLookupTableData	[256x1 uint16]
BluePaletteColorLookupTableData	[256x1 uint16]



## Anexo C

# Modos de operación

En este anexo se realiza una descripción más detallada de los principales modos de operación existentes en los ecógrafos, describiremos de forma breve la utilidad de cada uno de los modos y las características que presentan en su representación.

- **MODO B:** Este es el modo de operación básico de la ecografía. Lo que muestra este modo es un corte del corazón desde una perspectiva dada. Existen tres de esas ventanas de visualización distintas: desde el esternón, desde la boca del estómago y desde las costillas, bajo el brazo. Pero esto no afectará al aspecto de la imagen por lo que no haremos ninguna diferencia entre ellas.
- **MODO DOPPLER COLOR:** A partir del modo B, podemos crear una ventana de modo Doppler color. En dicha ventana, y mediante un código de colores, se muestra información sobre la velocidad, dirección y sentido del flujo de la sangre. El color rojo significa que el flujo se acerca, el azul que se aleja. En este modo la información más relevante corresponde a la información en color.
- **MODO M:** Este modo, consiste trata de una representación especial del modo B. Consiste en la evolución temporal del corte dado por el plano transversal a la ecografía, nos proporciona la capacidad de medir los volúmenes sistólicos y diastólicos de las cavidades cardíacas, las paredes existentes entre ellas y otras longitudes de importancia para el diagnóstico de ciertas patologías. En cuanto a su distribución presentan la información propia del modo en la parte inferior y en la mayor parte de los casos cuentan con una imagen de apoyo que se corresponde con un modo B o bien un Doppler color.
- **MODOS DOPPLER PULSADO Y DOPPLER CONTINUO:** Estos modos son, aproximadamente, al Doppler Color lo que el modo M es al modo B. Sirven para medir velocidades de la sangre en zonas localizadas ya sean pequeñas con el modo Doppler Pulsado, o elevadas con el Doppler Continuo. En cuanto a la estructura de la imagen no presentaran ninguna diferencia a la hora de la segmentación con respecto a al modo M.

- **MODOS DOPPLER TISULAR Y DOPPLER TISULAR PULSADO:**

Estos modos, sirven para evaluar la función miocárdica, permitiendo obtener medidas sobre velocidades, desplazamientos y deformaciones en una región miocárdica. Los modos Tisulares son los que mas información contienen en color. El Doppler Tisular es muy similar a un modo Doppler Color pero con la información a color menos localizada en la imagen. En el caso del modo Doppler Tisular Pulsado la prueba consta de dos partes en la parte inferior un ultrasonido de gran tamaño parecida a un modo M pero con una gran cantidad de información en color y un auxiliar en la parte superior que se corresponde con un Doppler Tisular.

## Anexo D

# OCR

OCR o ROC en castellano son las siglas para *Reconocedor Óptico de Caracteres*, se conoce por este nombre a los programas que sirven para digitalizar de forma automática textos de imágenes o documentos escaneados. Con la digitalización del documento, conseguimos tener la posibilidad de tratar la información de forma cómoda con un editor de textos y reducir el espacio que ocupa el documento. Recientemente este tipo de software ha cobrado mayor interés para aplicaciones destinadas al público en general, un ejemplo de esto es la aparición de aplicaciones para móviles de nueva generación que consiguen resultados relativamente exitosos. Uno de los mejores ejemplos de este tipo de software es el incluido por la versión Adobe Acrobat Pro y su herramienta *Reconocer texto*.

Los resultados obtenidos con los OCRs son muy dispares, la mayoría presentan deficiencias importantes en cuanto a las condiciones que se deben presentar para digitalizar un documento. Pero por lo general suelen contener los mismos bloques funcionales.

1. Binarización de la imagen: la mejor forma de obtener un buen resultado en la digitalización pasa por transformar la imagen original en una imagen binaria en la cual los pixel que se identifiquen con algún carácter estén el blanco y el resto de la imagen este en negro.
2. Segmentación y extracción de contorno: una vez obtenidos los pixels que pertenecen a un carácter deberemos separar las distintas formas o caracteres, posteriormente cada forma se reduce a su contorno para su posterior identificación.
3. Identificación de carácter: evaluaremos, cada forma cotejándola con la base de datos de caracteres, y finalmente identificaremos cada forma con el carácter correspondiente que supere un umbral de semejanza configurable.

En el desarrollo de este proyecto, tras el análisis de varios OCRs freeware, se decidió el uso del software GORC [11] por ofrecer, en nuestro caso los mejores resultados.



# Anexo E

## Manual de uso de la aplicación

Este anexo pretende ser un manual de uso de la aplicación desarrollada, así como describir la funcionalidad de cada uno de los botones que aparecen en la interfaz. Dividiremos la explicación de la aplicación en tres bloques que se corresponder con cada una de las ventanas de la aplicación. Interfaz de compresión, herramienta de modificación y herramienta de medida. Para el desarrollo de la aplicación se ha usado la herramienta GUI [15] de Matlab.

### E.1 Interfaz de compresión

En esta figura E.1 mostramos la interfaz de compresión, incluiremos una breve descripción de la funcionalidades de cada uno de los bloques que la forman.

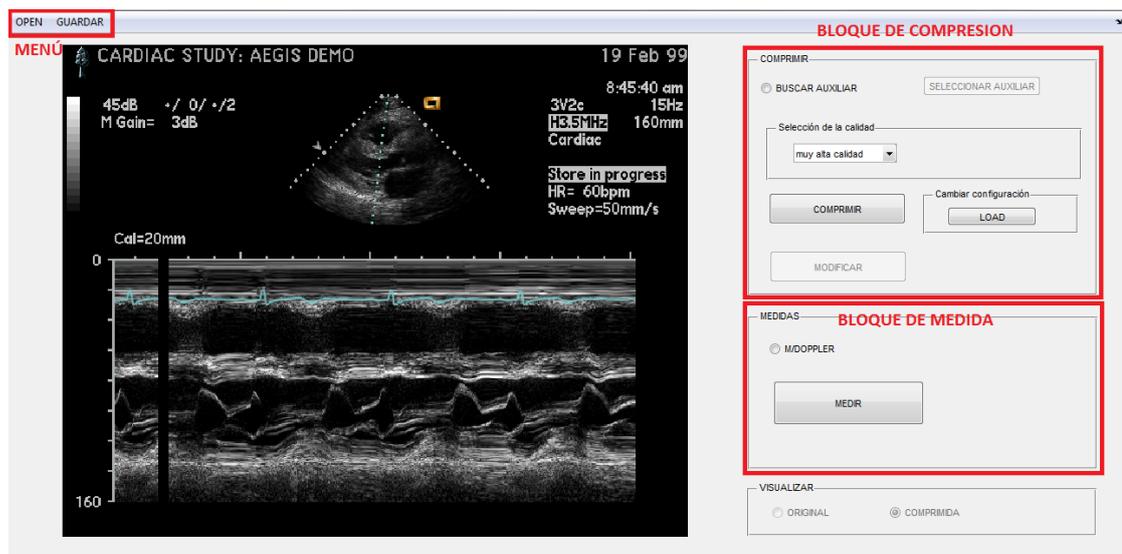


Figura E.1: Interfaz de compresión

### E.1.1 Menú

Al inicio de la aplicación pulsando en **OPEN** se abre un explorador que nos permite seleccionar el archivo que deseamos cargar, los formatos de fichero disponibles, son archivos DICOM o archivos previamente comprimidos con el nuevo formato de compresión. Una vez seleccionado el archivo que se desea abrir, se debe comprobar con cual de los 2 formatos permitidos se corresponde en el caso de ser una imagen DICOM, se extrae la imagen y se representa por pantalla. En el caso de imágenes comprimidas el proceso es un poco mas complicado, ya que antes de representar la imagen deberemos descomprimirla y reconstruirla. Al final cualquiera que sea el tipo de archivo tenemos en la ventana de visualización una imagen de la prueba médica, en el caso de haber abierto una prueba en formato DICOM se activan los objetos correspondientes al bloque 2 características de la compresión.

Desde el menú también podremos almacenar tanto una imagen comprimida como una imagen DICOM sobre la que se haya realizado alguna medida nueva, para ello pulsamos en **GUARDAR** donde primero se selecciona el formato a guardar y despues mediante un explorador se elige el nombre del archivo.

### E.1.2 Bloque de compresión

Dentro de este bloque como vemos en la figura E.2, aparecen numerosos botones, el primero de ellos es un botón de selección (**BUSCAR AUXILIAR**, pulsándolo seleccionamos que deseamos buscar figuras auxiliares, por defecto se buscara la única que existe en nuestro bando de pruebas, perteneciente al modelo Philips Envisor, pero se deseamos buscar otro auxiliar, debemos pulsar sobre **SELECCIONAR AUXILIAR** y seleccionar la figura a buscar.

A continuación desde el menú desplegable **Selección de calidad** cuanto queremos comprimir la imagen, por defecto estará fijado a muy alta calidad. Desde el botón **LOAD** podemos cargar algunos un fichero de configuración que contiene ciertos parámetros para la segmentación, la configuración actual funciona para todo nuestro banco de pruebas, pero cargar una nueva configuración podría ser necesario si no obtenemos el resultado deseado para un aparato no probado.

Una vez seleccionadas las características pulsaremos en **COMPRIMIR** para iniciar la segmentación y compresión de la imagen, al finalizar se muestra la imagen comprimida. desde el menú visualizar podemos seguir visualizando la imagen original.

Por último una vez realizada la compresión se activa el botón **MODIFICAR** pulsando este botón dejamos suspendida la ventana actual y se abre la herramienta de modificación en una nueva ventana.

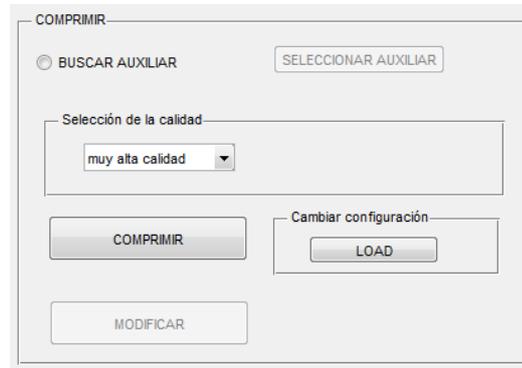


Figura E.2: Bloque de compresión

### E.1.3 Bloque de medida

El bloque de medida que vemos en la figura E.3, es mucho más sencillo ya que el único parámetro configurable es el tipo de medidas que se desea realizar, está especificado según el modo. De nuevo si pulsamos en el botón **MEDIR** dejaremos suspendida la ventana actual y en otra ventana se abre la herramienta de medida.



Figura E.3: Bloque de medida.

## E.2 Herramienta de modificación

La herramienta de modificación, en la figura E.4, nos da posibilidad de editar bloques de texto, eliminarlos, crear nuevos o importarlos. Además también podemos modificar la calidad del ultrasonido visualizando los cambios. Una vez finalizados los cambios y pulsando sobre **DONE** los cambios se reflejan al volver a la interfaz de compresión.

### E.2.1 Bloque de edición

Desde este bloque, que vemos en la figura E.5, podemos realizar varias acciones sobre los bloques existentes. Primero desde **SELECCIONAR BLOQUE**, elegimos sobre qué bloque queremos realizar las acciones, el número del bloque elegido debe coincidir con el que aparece al lado de cada bloque en la imagen. Una vez seleccionado el bloque su contenido se muestra en la ventana **TEXTO SELECCIONADO**, y se pueden realizar una de las siguientes acciones:

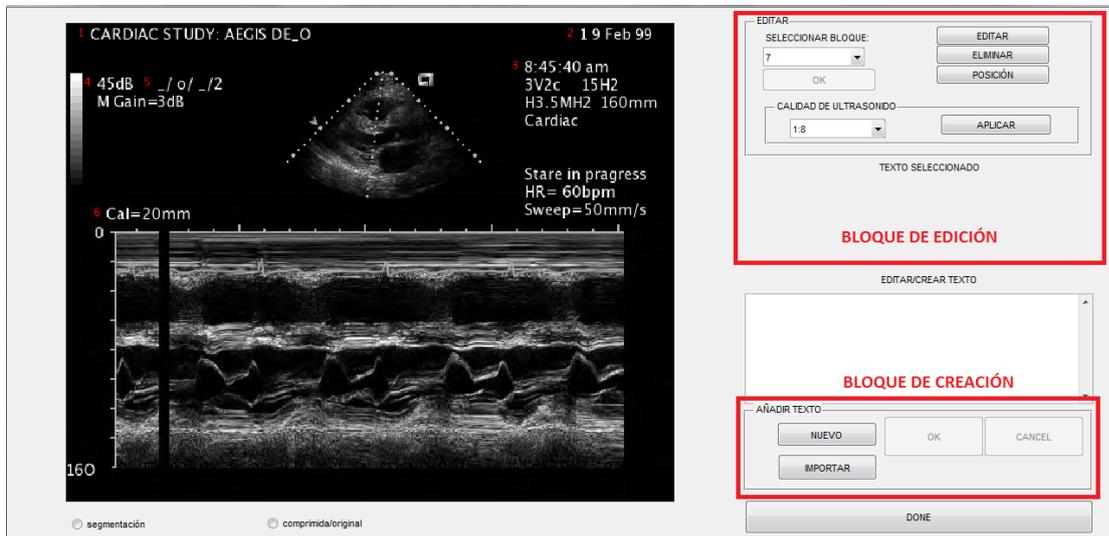


Figura E.4: Herramienta de modificación

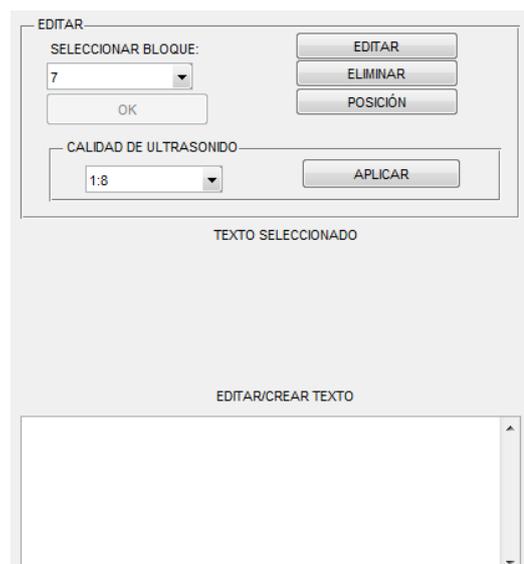


Figura E.5: Bloque de edición

- Editar: Pulsando en **EDITAR** en contenido del bloque se copiará en la ventana **EDITAR/CREAR TEXTO**, desde allí lo modificaremos hasta que su contenido sea el deseado. Para finalizar y modificar el bloque pulsaremos en **OK**
- Eliminar: Una vez seleccionado el bloque que deseamos eliminar, pulsaremos en **ELIMINAR**.
- Posición: es una forma rápida de mover un bloque de texto, una vez seleccionado el bloque pulsamos sobre **POSICIÓN** y aparece un cursor con el que elegimos el punto de la imagen donde se desea desplazar el texto.

Como ya hemos dicho desde este bloque podremos también variar la calidad de la compresión, simplemente seleccionaremos el valor deseado en el menú desplegable y ejecutaremos la nueva compresión con la calidad deseada pulsando en **APLICAR**. Al seleccionar la calidad seleccionamos el número de bits por pixel que deseamos, por ejemplo 1:8 significa que comprimiremos hasta obtener un bit por cada pixel.

### E.2.2 Bloque de creación

El bloque de creación que vemos en la figura E.6, podemos crear un nuevo bloque de texto tanto escribiendo su contenido nosotros mismos o importándolo de la imagen original, es decir seleccionando una zona en la imagen original de la volveremos a extraer los datos y a digitalizarlos.

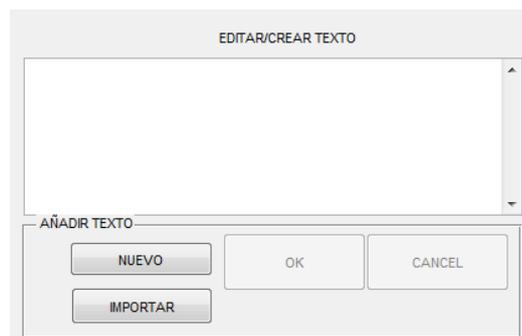


Figura E.6: Bloque de creación

Para crear un nuevo bloque, pulsaremos en **NUEVO**, en ese momento aparece un cursor con el que seleccionamos la posición del nuevo bloque, después escribimos el texto deseado en la ventana **EDITAR/CREAR TEXTO** y por último si queremos escribirlo pulsaremos en **OK** y si deseamos cancelar, pulsaremos en **CANCEL**.

Para importar un texto pulsaremos en **IMPORTAR**, aparece la imagen original y un cursor para seleccionar un cuadro en el que realizar el análisis una vez finalizado el nuevo cuadro aparece en la posición en la que hemos buscado.

### E.3 Herramienta de medida

En la figura E.7 podemos observar la herramienta de medida para el modo M. En el caso de los modos Doppler la ventana es idéntica solo que podremos realizar 2 medidas distintas.

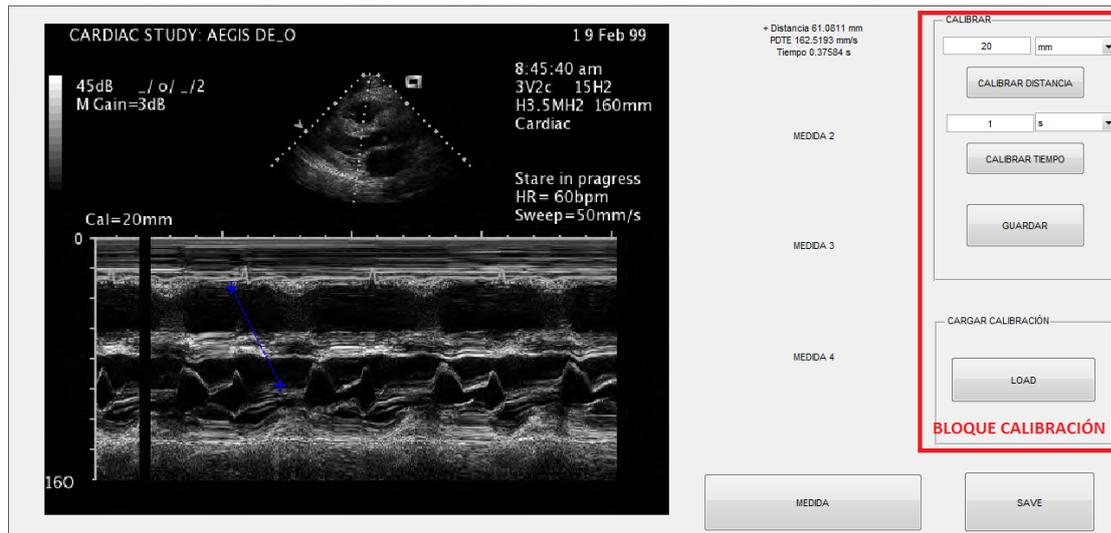


Figura E.7: Herramienta de medida

Lo primero que debemos hacer es calibrar la medida. Podemos cargar una calibración existente pulsando sobre **LOAD**, o bien crear una nueva sobre la imagen, para ello seleccionamos la unidad de medida y el valor en estas unidades, después pulsando en **CALIBRAR DISTANCIA** y **CALIBRAR TIEMPO** aparece un cursor con el que se selecciona el punto inicial y final de la medida en la imagen. Una vez realizada la calibración solo deberemos pulsar sobre **MEDIDA** y realizar las medidas. Las medidas se muestran sobre la imagen y los valores numéricos calculados se muestran a la derecha de a pantalla.

Una vez finalizadas las medidas deseadas pulsamos sobre **SAVE** y volveremos a la ventana principal donde en la imagen comprimidas aparecen las nuevas medidas, como podemos ver en la figura E.8.

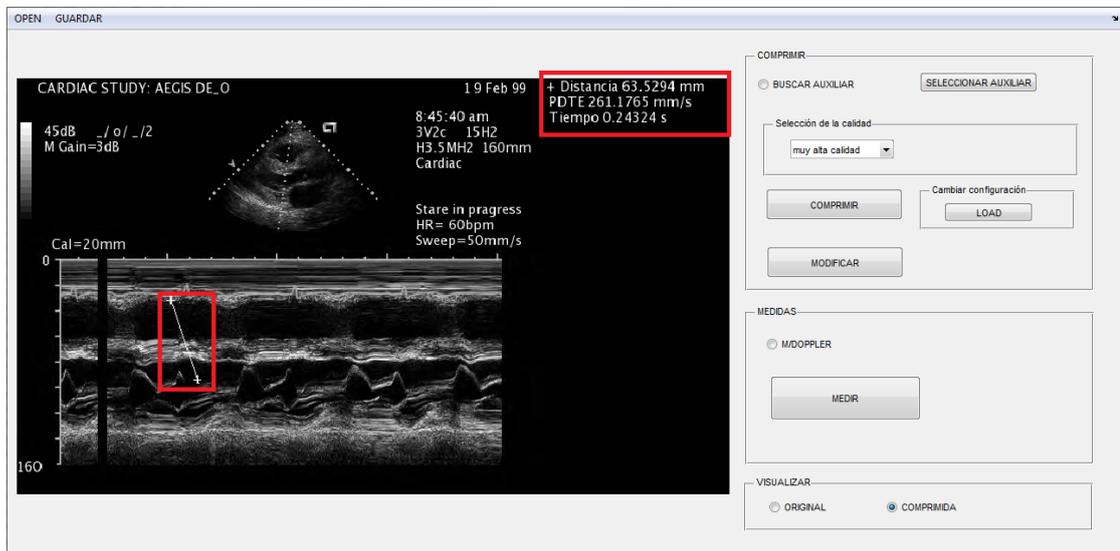


Figura E.8: Imagen comprimida con nueva medida añadida