



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
UNIDAD ZACATENCO

"IMPLEMENTACIÓN DE FILTROS EN IMÁGENES TRIDIMENSIONALES PARA SUPRESIÓN DE RUIDOS Y PRESERVACIÓN DE DETALLES FINOS CON APLICACIONES EN MEDICINA"

TESIS QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS EN INGENIERÍA DE
TELECOMUNICACIONES

PRESENTA:

ING. RUBÉN DARÍO SANSORES PECH

ASESOR:

DR. VOLODYMYR PONOMARYOV

CO-ASESOR:

M. EN C. MARCO ANTONIO ACEVEDO MOSQUEDA



México D.F.

Abril 2004



INSTITUTO POLITECNICO NACIONAL
COORDINACION GENERAL DE POSGRADO E INVESTIGACION

ACTA DE REVISION DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 17 del mes de Mayo del 2004 se reunieron los miembros de la Comisión Revisora de Tesis designada Por el Colegio de Profesores de Estudios de Posgrado e Investigación de la E. S. I. M. E. Para examinar la tesis de grado titulada:

"IMPLEMENTACION DE FILTROS EN IMAGENES TRIDIMENSIONALES PARA SUPRESION DE RUIDOS Y PRESERVACION DE DETALLES FINOS CON APLICACIONES EN MEDICINA"

Presentada por el alumno:

SANSORES

Apellido paterno

PECH

materno

RUBÉN DARÍO

nombre(s)

Con registro:

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|

aspirante al grado de:

MAESTRO EN CIENCIAS

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISION REVISORA

Director de tesis

DR. VOLODYMYR PONOMARYOV

DR. VLADIMIR A. KAZAKOV ERASOVA

DR. JAIR GARCIA LAMONT

M. EN C. MARCO ANTONIO ACEVEDO MOSQUEDA

DR. VLADISLAV KRAVCHENKO

M. EN C. MIGUEL SANCHEZ MERAZ

EL PRESIDENTE DEL COLEGIO

DR. FLORENCIO SANCHEZ SILVA

SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION



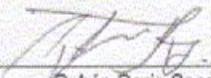
INSTITUTO POLITECNICO NACIONAL
COORDINACION GENERAL DE POSGRADO E INVESTIGACION

CARTA CESION DE DERECHOS

En la Ciudad de México, Distrito Federal, el día 20 del mes Mayo del año 2004 el que suscribe Rubén Dario Sansores Pech alumno del Programa de Maestría en Ciencias en Ingeniería de Telecomunicaciones con número de registro 011001, adscrito a la Sección de Estudios de Posgrado e Investigación de la ESIME Unidad Zacatenco, manifiesta que es autor intelectual del presente Trabajo de Tesis bajo la dirección del Dr. Volodymyr Ponomaryov y cede los derechos del trabajo intitulado: "IMPLEMENTACIÓN DE FILTROS EN IMÁGENES TRIDIMENSIONALES PARA SUPRESIÓN DE RUIDOS Y PRESERVACIÓN DE DETALLES FINOS CON APLICACIONES EN MEDICINA", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, graficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: dario_sansores@hotmail.com.

Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Rubén Dario Sansores Pech

ÍNDICE

| | |
|--|-------------|
| GLOSARIO..... | i |
| RELACIÓN DE TABLAS..... | ii |
| RELACIÓN DE FIGURAS..... | iii |
| NOMENCLATURA DE SÍMBOLOS..... | vi |
| RESUMEN..... | viii |
| ABSTRACT..... | ix |
| | |
| CAPITULO I. INTRODUCCIÓN..... | 1 |
| I.1. Objetivos y metas..... | 1 |
| I.2. Justificación..... | 1 |
| I.3. Estructura de la tesis..... | 2 |
| | |
| CAPITULO II. ESTADO DEL ARTE DE LAS IMAGENES EN TRES DIMENSIONES..... | 4 |
| II.1. Introducción..... | 4 |
| II.2. Los Rayos X..... | 4 |
| II.3. Cirugía Asistida por Ordenador (CAO)..... | 5 |
| II.4. La Tele Radiología..... | 7 |
| II.5. Ultrasonido 3D de Manos Libres..... | 8 |
| II.6. Fusión de Imágenes 3D..... | 9 |
| II.7. Reconstrucción 3D..... | 11 |
| II.8. Sistema de Navegación en 3D..... | 12 |
| II.9. Conclusiones..... | 13 |
| | |
| CAPITULO III. SISTEMA DESARROLLADO PARA CAPTURAR IMAGENES EN 3D..... | 15 |
| III.1. Introducción..... | 15 |
| III.2. Bloque de Captura..... | 15 |
| III.3. Bloque de Sincronización..... | 16 |
| III.4. Bloque de Exploración..... | 17 |
| III.5. Proceso para Capturar Imágenes en 3D..... | 18 |
| III.6. Imágenes 3D Obtenidas del Sistema Experimental..... | 23 |
| III.7. Conclusiones..... | 28 |
| | |
| CAPITULO IV. FILTROS NO LINEALES EXISTENTES APLICADOS A IMÁGENES 2D.... | 30 |
| IV.1. Introducción..... | 30 |
| IV.2. Filtros Lineales Versus Filtros No Lineales..... | 30 |
| IV.3. Filtro α - Trimmed Mean..... | 30 |
| IV.4. Filtro Modified Trimmed Mean..... | 31 |
| IV.5. Filtro de Orden de Rango..... | 32 |
| IV.6. Filtro Mediano de Multietapa..... | 32 |
| IV.7. Filtro de Comparación y Selección..... | 33 |
| IV.8. Filtro de Selección de Promedio y Filtro de Selección de Mediana..... | 33 |
| IV.9. Filtros de Selección de Rango..... | 34 |
| IV.9.1. Filtro Lower Upper Middle Smoother..... | 34 |

| | |
|--|-----------|
| IV.9.2. Filtro Lower Upper Middle Sharpener..... | 34 |
| IV.9.3. Filtros Lower Upper Middle..... | 35 |
| IV.10. Filtro Wilcoxon..... | 35 |
| IV.11. Estimadores M..... | 35 |
| IV.12. Criterios Usados para Obtener el Rendimiento del Filtro Propuesto..... | 38 |
| IV.13. Conclusiones..... | 38 |
| | |
| CAPITULO V. IMPLEMENTACIÓN DE LOS FILTROS DE ESTADÍSTICA DE ORDEN EN IMÁGENES MÉDICAS 3D..... | 40 |
| V.1. Introducción..... | 40 |
| V.2. Ventana y Cubo..... | 40 |
| V.3. Criterios usados para obtener el rendimiento de los filtros 3D..... | 41 |
| V.4. Filtros de Estadística de Orden en 3D..... | 41 |
| V.4.1. Filtro α - Trimmed Mean..... | 41 |
| V.4.2. Filtro Modified Trimmed Mean..... | 42 |
| V.4.3. Filtro Ranked – Order..... | 42 |
| V.4.4. Filtros Multistage Median..... | 42 |
| V.4.5. Filtro Comparison y Selection..... | 43 |
| V.4.6. Filtro Selective Average..... | 43 |
| V.4.7. Filtro Selective Median..... | 43 |
| V.4.8. Filtro LUM Smooth..... | 43 |
| V.4.9. Filtro LUM Sharpener..... | 43 |
| V.4.10. Filtro LUM..... | 43 |
| V.4.11. Filtro WILCOXON..... | 44 |
| V.4.12. Filtro MM-KNN..... | 44 |
| V.4.12.1. Filtro WM-KNN..... | 45 |
| V.4.12.2. Filtro ABSTM-KNN (Ansari-Bradley-Siegel-Tukey)..... | 46 |
| V.5. Resultados de las Simulaciones..... | 46 |
| V.5.1. Imágenes Resultantes al Suprimir 30% de Ruido Impulsivo..... | 52 |
| V.5.2. Imágenes Resultantes al Suprimir 20% de Ruido Impulsivo..... | 56 |
| V.5.3. Imágenes Resultantes al Suprimir 10% de Ruido Impulsivo..... | 58 |
| V.5.4. Imágenes del Músculo del Antebrazo..... | 61 |
| V.6. Ventana versus Cubo..... | 62 |
| V.7. Conclusiones..... | 66 |
| | |
| CAPITULO VI. IMPLEMENTACION DE LOS FILTROS DE ESTADÍSTICA DE ORDEN EN EL DSP TMS320C6711 PARA PROCESAR IMÁGENES 3D..... | 68 |
| VI.1. Introducción..... | 68 |
| VI.2. Procesamiento de Imágenes 3D..... | 68 |
| VI.3. Dispositivos Usados para el Procesamiento de Imágenes 3D..... | 69 |
| VI.3.1. TMS320C6711..... | 70 |
| VI.3.2. FPGA..... | 70 |
| VI.4. FPGA Versus DSP..... | 72 |
| VI.5. Programación del TMS320C6711..... | 72 |
| VI.6. Implementación del TMS320C6711 en imágenes 3D..... | 73 |
| VI.7. Conclusiones..... | 75 |
| | |
| CAPITULO VII. CONCLUSIONES..... | 77 |
| VII.1. Conclusiones..... | 77 |
| VII.2. Trabajos a Futuro..... | 78 |

BIBLIOGRAFÍA 79
ANEXO A 81
ANEXO B 83
ANEXO C 95
ANEXO D 105

GLOSARIO

| | |
|---------------|---|
| A/D: | Analógico a Digital. |
| Ablación: | Separación o extirpación de cualquier parte del cuerpo. |
| ABSTM: | Ansari Bradley Siegel Tukey. |
| ALU: | Unidad de Lógica Aritmética. |
| Biopsia: | Examen de un trozo de tejido perteneciente a un ser vivo, para determinar la naturaleza maligna o benigna de un tumor. |
| BMP: | Formato en Mapa de Bits. |
| BS: | Barrel Shifter para desplazar un dato varios bits a derecha/izquierda. |
| CAO: | Cirugía Asistida por Ordenador. |
| Coherente: | Las frecuencias generadas en el receptor y utilizadas para demodulación se sincronizan para oscilar a frecuencias generadas en el transmisor. |
| D/A: | Digital a Analógico. |
| DICOM 3.0: | DICOM, el Estándar en Imagen Digital y Comunicaciones en Medicina de ACR-NEMA se ha desarrollado para encontrar las necesidades que fabricantes y usuarios tienen con el equipamiento de imagen médica para la interconexión de dispositivos sobre redes estándares. DICOM también provee medios por los que los usuarios de equipamiento de imagen pueden intercambiar información de dispositivos diferentes. |
| DSP: | Digital Signal Processing. |
| DVD: | Digital Video Disc. |
| Estenografía: | Métodos para ocultar información en imágenes. |
| FOV: | Field Of View. |
| Frame Relay: | Frame Relay proporciona conexiones entre usuarios a través de una red pública, del mismo modo que lo haría una red privada con circuitos punto a punto. De hecho, su gran ventaja es la de reemplazar las líneas privadas por un sólo enlace a la red. |
| GE: | General Electric. |
| GFLOPS: | Giga Float Per Second. |
| LUM: | Lower Upper Middle. |
| MAC: | Multiplicadora/Acumuladora. |
| MAD: | The Median of Absolute deviations from Median. |
| Píxel: | Elementos que forman una imagen 2D. |
| Resección: | Extirpación quirúrgica, total o parcial, de un órgano o tejido. |
| RM: | Resonancia Magnética. |
| SVCD: | Super Video Compact Disc. |
| TAG: | Tomografía Axial Gráfica. |
| VCD: | Video Compact Disc. |
| Voxel: | Elementos que forman una imagen 3D. |
| Watermarking: | Marca de agua. |

RELACIÓN DE TABLAS

| | | |
|------------|---|----|
| Tabla 5.1. | Parámetros para los filtros KNN y las Funciones de Influencia..... | 46 |
| Tabla 5.2. | Resultados obtenidos de las pruebas realizadas con ruido Gaussiano..... | 47 |
| Tabla 5.3. | Resultados obtenidos de las pruebas realizadas con ruido impulsivo..... | 49 |
| Tabla 5.4. | Resultados obtenidos de las pruebas realizadas con la imagen de 640x480x90..... | 51 |
| Tabla 5.5. | Cubos implementados con el filtro MMKNN con F.I. Cut..... | 63 |
| Tabla 5.6. | Cubos implementados con el filtro MMKNN con F.I. Hampel..... | 63 |
| Tabla 5.7. | Cubos implementados con el filtro Alfa Trimmed Mean..... | 64 |
| Tabla 5.8. | Cubos implementados con el filtro Modified Trimmed Mean..... | 64 |
| Tabla 6.1. | Tiempo en segundos de los filtros 3D..... | 73 |
| Tabla 6.2. | Tiempo en segundos de los filtros 3D..... | 73 |
| Tabla 6.3. | Tiempo en segundos de los filtros 3D..... | 74 |
| Tabla 6.4. | Tiempo en segundos de los filtros 3D..... | 74 |
| Tabla 6.5. | Tiempo en segundos de los filtros 3D..... | 75 |
| Tabla 6.6. | Tiempo en segundos de los filtros 3D..... | 75 |

RELACIÓN DE FIGURAS

| | | |
|--------------|--|----|
| Figura 2.1. | Análisis de movimiento en 3D..... | 4 |
| Figura 2.2. | Esquema de un sistema CAO..... | 6 |
| Figura 2.3. | Instrumental quirúrgico adaptado y una vista general del primer prototipo del sistema..... | 7 |
| Figura 2.4. | Ultrasonido 3D..... | 8 |
| Figura 2.5. | Realidad virtual..... | 9 |
| Figura 2.6. | Pista motorizada..... | 10 |
| Figura 2.7. | Fusión de imágenes 3D..... | 10 |
| Figura 2.8. | Método de mallado..... | 11 |
| Figura 2.9. | Sistema de luz estructurada..... | 12 |
| Figura 2.10. | Imagen adquirida por el sistema de navegación en 3D..... | 13 |
| Figura 3.1. | Modo B scan..... | 15 |
| Figura 3.2. | Bloque de captura..... | 16 |
| Figura 3.3. | Bloque de sincronización..... | 16 |
| Figura 3.4. | Bloque de sincronización..... | 17 |
| Figura 3.5. | Carril de referencia..... | 17 |
| Figura 3.6. | Bloque de exploración..... | 18 |
| Figura 3.7. | Exploración o barrido..... | 18 |
| Figura 3.8. | Diagrama a bloques de la captura de imágenes 2D..... | 19 |
| Figura 3.9. | Sistema experimental para capturar imágenes de 3D..... | 20 |
| Figura 3.10. | Estructura metálica..... | 20 |
| Figura 3.11. | Figura de los órganos explorados y forma de exploración del transductor..... | 21 |
| Figura 3.12. | Planos donde se realizaron los cortes de la imagen 3D..... | 22 |
| Figura 3.13. | Formando la imagen 3D..... | 23 |
| Figura 3.14. | Imágenes de los cortes en el plano YZ..... | 23 |
| Figura 3.15. | Imágenes de los cortes en el plano XZ..... | 24 |
| Figura 3.16. | Imagen del hígado y riñón..... | 24 |
| Figura 3.17. | Imágenes de los cortes en el plano YZ..... | 25 |
| Figura 3.18. | Imágenes de los cortes en el plano XZ..... | 26 |
| Figura 3.19. | Imagen del músculo del antebrazo..... | 26 |
| Figura 3.20. | Imágenes de los cortes en el plano YZ..... | 27 |
| Figura 3.21. | Imágenes de los cortes en el plano XZ..... | 28 |
| Figura 4.1. | Subventanas para el filtro mediano multietapa..... | 33 |
| Figura 4.2. | Estructura de los filtros de selección de rango..... | 34 |
| Figura 5.1. | Bosquejo del barrido en 2D y 3D..... | 40 |
| Figura 5.2. | Cubo de barrido 3x3x3..... | 41 |
| Figura 5.3. | Subcubos de 3x3x3 voxels, para los filtros no lineales Multistage Median..... | 42 |
| Figura 5.4. | Gráfica de PSNR de los mejores filtros 3D..... | 48 |
| Figura 5.5. | Gráfica de MAE de los mejores filtros 3D..... | 48 |
| Figura 5.6. | Gráfica de PSNR de los mejores filtros 3D..... | 50 |
| Figura 5.7. | Gráfica de MAE de los mejores filtros 3D..... | 50 |
| Figura 5.8. | Planos donde se realizaron los cortes de la imagen 3D..... | 51 |
| Figura 5.9. | Imagen original e imagen degradada con 30% de ruido impulsivo, corte del plano XY..... | 52 |
| Figura 5.10. | Imagen restaurada con el filtro LUM Smooth y su imagen de error, corte del plano XY..... | 52 |
| Figura 5.11. | Imagen restaurada con el filtro MSM5 y su imagen de error, corte del plano XY.... | 52 |

| | | |
|--------------|--|----|
| Figura 5.12. | Imagen restaurada con el filtro MMKNN con F.I. Hampel y su imagen de error, corte del plano XY..... | 53 |
| Figura 5.13. | Imagen restaurada con el filtro MMKNN con F.I. Cut y su imagen de error, corte del plano XY..... | 53 |
| Figura 5.14. | a) imagen original; b) imagen restaurada con el filtro LUM Smooth; c) imagen restaurada con el filtro MSM5; d) imagen restaurada con el filtro MMKNN con F.I. Hampel; e) imagen restaurada con el filtro MMKNN con F.I. Cut; f) imagen degradada con 30% de ruido impulsivo; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano XZ..... | 54 |
| Figura 5.15. | a) imagen original; b) imagen degradada con 30% de ruido impulsivo; c) imagen restaurada con el filtro LUM Smooth; d) imagen restaurada con el filtro MSM5; e) imagen restaurada con el filtro MMKNN con F.I. Hampel; f) imagen restaurada con el filtro MMKNN con F.I. Cut; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano YZ..... | 55 |
| Figura 5.16. | Imagen original e imagen degradada con 20% de ruido impulsivo, corte del plano XY..... | 56 |
| Figura 5.17. | Imagen restaurada con el filtro LUM Smooth y su imagen de error, corte del plano XY..... | 56 |
| Figura 5.18. | Imagen restaurada con el filtro MSM5 y su imagen de error, corte del plano XY.... | 56 |
| Figura 5.19. | Imagen restaurada con el filtro MMKNN con F.I. Hampel y su imagen de error, corte del plano XY..... | 57 |
| Figura 5.20. | Imagen restaurada con el filtro MMKNN con F.I. Cut y su imagen de error, corte del plano XY..... | 57 |
| Figura 5.21. | Imagen original e imagen degradada con 10% de ruido impulsivo, corte del plano XY..... | 58 |
| Figura 5.22. | Imagen restaurada con el filtro LUM Smooth y su imagen de error, corte del plano XY..... | 58 |
| Figura 5.23. | Imagen restaurada con el filtro MSM5 y su imagen de error, corte del plano XY.... | 58 |
| Figura 5.24. | Imagen restaurada con el filtro MMKNN con F.I. Hampel y su imagen de error, corte del plano XY..... | 59 |
| Figura 5.25. | Imagen restaurada con el filtro MMKNN con F.I. Cut y su imagen de error, corte del plano XY..... | 59 |
| Figura 5.26. | Imagen de las manchas ocasionadas por el transductor coherente del equipo de US..... | 60 |
| Figura 5.27. | Imágenes originales, cortes del plano XY..... | 61 |
| Figura 5.28. | Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XY..... | 61 |
| Figura 5.29. | Imágenes de error, cortes del plano XY..... | 62 |
| Figura 5.30. | Ventana a) y cubos de barrido..... | 63 |
| Figura 5.31. | a) imagen original, b) imagen con 30% de ruido impulsivo, imagen restaurada con el filtro MMKNN con F.I. Cut implementando la c) ventana "a", d) el cubo "b", e) el cubo "c", f) el cubo "d", g) el cubo "e", h) el cubo "f", i) el cubo "g", j) el cubo "h", k) el cubo "i"..... | 65 |
| Figura 5.32. | Imágenes de error del filtro MMKNN con F.I. Cut implementando la a) ventana "a", b) el cubo "b", c) el cubo "c", d) el cubo "d", e) el cubo "e", f) el cubo "f", g) el cubo "g", h) el cubo "h", i) el cubo "i"..... | 66 |
| Figura 6.1. | Diagrama a bloques del procesamiento de imágenes 3D..... | 68 |
| Figura 6.2. | Tarjeta de evaluación del TMS320C6711..... | 71 |
| Figura 6.3. | Tarjetas de desarrollo de FPGA..... | 71 |
| Figura 6.4. | Ventana a) y cubos de barrido..... | 74 |
| Figura A.1. | Diagrama esquemático..... | 82 |

| | | |
|--------------|--|----|
| Figura B.1. | a) imagen original; b) imagen restaurada con el filtro LUM Smooth; c) imagen restaurada con el filtro MSM5; d) imagen restaurada con el filtro MMKNN con F.I. Hampel; e) imagen restaurada con el filtro MMKNN con F.I. Cut; f) imagen degradada con 20% de ruido impulsivo; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano XZ..... | 84 |
| Figura B.2. | a) imagen original; b) imagen degradada con 20% de ruido impulsivo; c) imagen restaurada con el filtro LUM Smooth; d) imagen restaurada con el filtro MSM5; e) imagen restaurada con el filtro MMKNN con F.I. Hampel; f) imagen restaurada con el filtro MMKNN con F.I. Cut; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano YZ..... | 85 |
| Figura B.3. | a) imagen original; b) imagen restaurada con el filtro LUM Smooth; c) imagen restaurada con el filtro MSM5; d) imagen restaurada con el filtro MMKNN con F.I. Hampel; e) imagen restaurada con el filtro MMKNN con F.I. Cut; f) imagen degradada con 10% de ruido impulsivo; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano XZ..... | 86 |
| Figura B.4. | a) imagen original; b) imagen degradada con 10% de ruido impulsivo; c) imagen restaurada con el filtro LUM Smooth; d) imagen restaurada con el filtro MSM5; e) imagen restaurada con el filtro MMKNN con F.I. Hampel; f) imagen restaurada con el filtro MMKNN con F.I. Cut; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano YZ..... | 87 |
| Figura B.5. | Imágenes originales, cortes del plano XZ..... | 88 |
| Figura B.6. | Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XZ..... | 88 |
| Figura B.7. | Imágenes de error, cortes del plano XZ..... | 89 |
| Figura B.8. | Imágenes originales, cortes del plano YZ..... | 89 |
| Figura B.9. | Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano YZ..... | 90 |
| Figura B.10. | Imágenes de error, cortes del plano YZ..... | 90 |
| Figura B.11. | Imágenes originales, cortes del plano XY..... | 91 |
| Figura B.12. | Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XY..... | 91 |
| Figura B.13. | Imágenes de error, cortes del plano XY..... | 92 |
| Figura B.14. | Imágenes originales, cortes del plano XZ..... | 92 |
| Figura B.15. | Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XZ..... | 93 |
| Figura B.16. | Imágenes de error, cortes del plano XZ..... | 93 |
| Figura B.17. | Imágenes originales, cortes del plano YZ..... | 94 |
| Figura B.18. | Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano YZ..... | 94 |
| Figura B.19. | Imágenes de error, cortes del plano YZ..... | 94 |

NOMENCLATURA DE SÍMBOLOS

| | | |
|---|--|----|
| n | Número total de elementos..... | 31 |
| r | Coficiente del filtro Trimmed Mean..... | 31 |
| s | Coficiente del filtro Trimmed Mean..... | 31 |
| $X_{(i)}$ | i th orden estadístico..... | 31 |
| N | Número total de elementos..... | 31 |
| α | Coficiente del filtro Alfa Trimmed Mean..... | 31 |
| MED{} | Mediana de los datos..... | 32 |
| MAX{} | Máximo valor de los datos..... | 32 |
| MEAN{} | Media aritmética de los datos..... | 33 |
| s | Coficiente del filtro LUM Smooth y LUM..... | 34 |
| t | Coficiente del filtro LUM Sharp y LUM..... | 34 |
| t_L | Control de agudeza..... | 34 |
| X^* | Píxel central de la ventana..... | 34 |
| $\psi(X)$ | Función de influencia..... | 36 |
| $\psi_{\text{cut}(r)}(X)$ | Función de influencia Cut..... | 37 |
| $\psi_{\alpha, \beta, r}(X)$ | Función de influencia Hampel..... | 37 |
| $\psi_{\text{sin}(r)}(X)$ | Función de influencia Seno de Andrews..... | 37 |
| $\psi_{\text{bi}(r)}(X)$ | Función de influencia Tukey..... | 37 |
| $\psi_{\text{ber}(r)}(X)$ | Función de influencia Bernoulli..... | 37 |
| MN | Matriz MxN..... | 38 |
| LMN | Matriz LxMxN..... | 41 |
| X^* | Voxel central del cubo..... | 43 |
| $\hat{e}_{\text{MMKNN}}^{(q)}$ | Es la estimación de la iteración q del filtro MMKNN..... | 44 |
| $D_S(x(i, j, k))$ | Detector de picos..... | 44 |
| $\hat{f}_{\text{RMKNN}}^{(w)}(i, j, k)$ | Es la estimación de la iteración w del filtro WMKNN..... | 45 |

| | | |
|---|--|----|
| $\hat{f}_{\text{ABSTMKN}}^{(w)}(i, j, k)$ | Es la estimación de la iteración w del filtro ABSTMKN..... | 46 |
| σ^2 | Varianza..... | 47 |

RESUMEN

La captura y el procesamiento de imágenes 3D en el área de medicina es un tema nuevo y muy amplio. Existen diferentes técnicas de filtrado para restaurar las imágenes médicas de los equipos de ultrasonido, tomografía o resonancia magnética.

En esta tesis se presenta la investigación del filtrado 3D para suprimir ruido de diferente naturaleza y restaurar las imágenes médicas, considerando el tiempo de procesamiento.

Se diseñó y desarrolló el sistema experimental para capturar imágenes de ultrasonido en 3D. Las imágenes 3D de los órganos humanos explorados se pueden observar desde diferentes planos.

Los filtros 2D propuestos en literatura y unos filtros de tipo RMKNN, se adaptaron en 3D y se implementaron para suprimir ruido de diferente naturaleza, obteniendo el desempeño de cada filtro de acuerdo a los criterios MAE y PSNR. Los mejores algoritmos de filtrado se implementaron para restaurar las imágenes 3D.

En el DSP TMS320C6711 se implementó el procesamiento 3D usando los filtros mencionados, obteniendo los tiempos de filtrado. Buscando reducir más los tiempos de procesamiento se propusieron diferentes cubos de barrido obteniendo menores tiempos de procesamiento.

ABSTRACT

The capture and 3D image processing in the medicine are a new and very wide topic. Different filtering techniques exist for restoring the medical images in the ultrasound, tomographic or magnetic resonance equipment. There are different techniques with good restoration ability for medical images but at a large time of processing.

In this thesis, the investigation of the 3D filtering for noise suppressing and restoring the medical images, considering the time needed for processing has been done.

The design and development of the experimental system used for capturing the ultrasound images in 3D is presented. The 3D images of the explored human organs permit to observe it from different planes.

The 2D filters proposed in literature and RMKNN filters have been adapted for 3D and implemented to suppress noise, obtaining the performance of each a filter according to criteria MAE and PSNR. The 3D filters were implemented to restore the images that were blurred, due to the speckle caused by the coherent transducer of the ultrasound equipment.

In DSP TMS320C6711, the 3D filters were implemented obtaining the processing times. Looking for to reduce more the processing times we proposed different sweeping cubes obtaining smaller values of processing time.

CAPITULO I. INTRODUCCIÓN

I.1. Objetivos y metas

Los objetivos de esta investigación son:

- Proponer y diseñar un sistema para la captura de imágenes de ultrasonido en 3D con aplicación en medicina.
- Desarrollar e investigar los filtros no lineales para el procesamiento de imágenes médicas 3D con el propósito de mejorar su ilustración.

Metas:

- Seleccionar y describir algunos filtros no lineales existentes aplicados a imágenes 2D, de acuerdo a sus propiedades y características.
- Describir el procesamiento de los filtros de estadística de orden en 3D.
- Adaptar los filtros 2D de estadística de orden seleccionados a filtros 3D.
- Aplicar los criterios para comparar el desempeño de los filtros 3D.
- Analizar el desempeño de cada filtro 3D, y seleccionar los más eficientes para restaurar las imágenes médicas 3D en presencia de ruido de diferente naturaleza.
- Investigar los diferentes dispositivos orientados al procesamiento de imágenes 3D.
- Implementar el DSP TMS320C6711 en el procesamiento de imágenes 3D.

I.2. Justificación

La captura y la implementación de las imágenes en tres dimensiones es un nuevo campo que actualmente está evolucionando. En los últimos años se han observado los resultados de los nuevos sistemas que implementan las imágenes en tres dimensiones, estos sistemas permiten realizar procesos con mayor seguridad, precisión y rapidez.

La posibilidad de procesar imágenes (campos) en 3D, nos conduce a una nueva aplicación; donde es necesario mejorar la calidad de objetos 3D dentro de la imagen, con supresión de ruido que siempre afecta el proceso de comunicación o adquisición de campos.

Enfocándonos al área de medicina, las imágenes 3D obtenidas por medio de los equipos de ultrasonido, resonancia magnética o tomografía, siempre se encuentran afectadas por ruido,

debido a los transductores y a la forma de captura de la imagen 2D. Por esta razón; es esencial suprimir el ruido, para recuperar los detalles finos, y definir los contornos de la imagen 3D, mejorando la calidad de la imagen del órgano explorado.

Con la imagen 3D restaurada los diagnósticos de los radiólogos o cirujanos son más precisos, la segmentación de la imagen es posible, permitiendo el reconocimiento e interpretación de la imagen y su aplicación en robótica o visión artificial.

La mayoría de los sistemas de captura de imágenes 3D, son realizados por hospitales, centros médicos o laboratorios para aplicaciones específicas. Por este motivo algunos sistemas implementan computadoras personales, perdiendo mucho tiempo en el procesamiento de decenas de imágenes.

La captura y el procesamiento de las imágenes 3D son casos especiales, que dependen de los equipos utilizados y de la aplicación final de la imagen.

Por las razones descritas anteriormente es muy importante realizar una investigación para proponer un sistema de captura y nuevas herramientas para el procesamiento de imágenes 3D, tomando en consideración dos puntos muy importantes: la calidad de restauración de la imagen y el tiempo de procesamiento.

I.3. Estructura de la tesis

Esta tesis presenta tres puntos muy importantes:

- La captura de imágenes 3D.
- El procesamiento 3D para supresión de ruido.
- La implementación del DSP en el procesamiento de imágenes 3D.

Se describieron estos tres puntos de la siguiente manera:

En el capítulo II se describen los sistemas para capturar imágenes 3D y sus aplicaciones, enfocándonos a los sistemas implementados en el área de medicina. También se definen los equipos necesarios para poder realizar este proceso.

En el capítulo III se presenta el bosquejo de la captura de imágenes 3D, la descripción de los bloques que conforman el sistema diseñado y desarrollado para la captura de las imágenes 3D. También se presentan la primera imagen 3D capturada y el segundo grupo de imágenes 3D capturadas.

En el capítulo IV se describen las ventajas de los filtros no lineales comparados con los filtros lineales para suprimir ruido en imágenes 2D, se presentan los filtros seleccionados con sus características para la supresión de ruido y los criterios que se utilizan para obtener el rendimiento del filtro.

En el capítulo V se define el procesamiento de imágenes 3D, la adaptación en 3D de los filtros de estadísticas de orden seleccionados, el desempeño de cada filtro 3D al suprimir ruido Gaussiano e impulsivo en las imágenes capturadas con el sistema experimental, los filtros 3D más eficientes para restaurar las imágenes 3D originales, el desempeño de la ventana de barrido y diferentes cubos de barrido propuestos para la supresión de ruido en imágenes 3D.

En el capítulo VI se presentan las etapas para procesar las imágenes 3D, y los tiempos de procesamiento de los filtros de estadística de orden 3D con diferentes porcentajes de ruido impulsivo, implementados en el DSP TMS320C6711.

CAPITULO II. ESTADO DEL ARTE DE LAS IMÁGENES EN TRES DIMENSIONES

II.1. Introducción

En este capítulo se presenta el estado del arte de las imágenes en 3D, en el cual se describen los diferentes sistemas para capturar las imágenes en 3D y sus aplicaciones.

Como se podrá observar a continuación; la captura, el procesamiento y la aplicación de las imágenes en 3D, es un campo nuevo, que en los últimos años se han empezado a observar las primeras imágenes en 3D y sus aplicaciones.

II.2. Los Rayos X

El tratamiento y la prevención de lesiones deportivas, podría facilitarse con el uso de una máquina de avanzada tecnología que registra imágenes en tres dimensiones del esqueleto humano.

Las imágenes 3D revelan los problemas que padece la persona y permiten que los fisioterapeutas elaboren el tratamiento adecuado.

Cuando el paciente acude al laboratorio para hacerse el escáner, cerca de 40 sensores son colocados en sus hombros, codos, caderas, rodillas y tobillos. Siete cámaras infrarrojas recogen la luz que se refleja en los sensores, mientras la persona desempeña la actividad deportiva pertinente. Esto se reproduce a continuación como un esqueleto de tres dimensiones, suministrando imágenes por computadora que pueden ser rotadas y vistas desde cualquier ángulo.

El sofisticado programa puede superponer la imagen del esqueleto con la que llega a través de las cámaras infrarrojas; entonces se obtiene el análisis del movimiento en 3D [1].

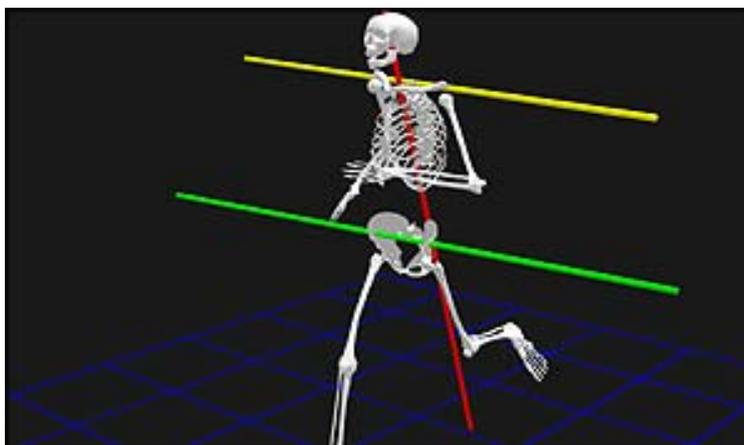


Figura 2.1. Análisis de movimiento en 3D.

II.3. Cirugía Asistida por Ordenador (CAO)

La Cirugía Asistida por Ordenador (CAO) constituye un área tecnológica relativamente reciente que intenta desarrollar y suministrar al cirujano una serie de herramientas que le asistan en la planificación y ejecución de procedimientos quirúrgicos. La idea subyacente en CAO no es la de la sustitución del cirujano en tareas quirúrgicas, hecho imposible de llevar a cabo en la actualidad, sino el desarrollar técnicas y sistemas que permitan ayudar al cirujano en las fases del procedimiento quirúrgico, esto es, diagnóstico, planificación y ejecución.

Si bien en la actualidad, la CAO presenta aplicaciones en prácticamente todas las disciplinas médicas, el presente trabajo se centra en la aplicación de CAO a Neurocirugía.

La CAO en Neurocirugía persigue el desarrollo y puesta a punto de sistemas que permitan al cirujano el poder llevar a cabo en el quirófano las intervenciones que han sido previamente planificadas conociendo en todo momento de forma precisa la posición del instrumental quirúrgico respecto a la anatomía cerebral del paciente. La puesta a punto de tales sistemas, llevan una serie de pasos comunes a todo sistema de CAO que se detallan a continuación:

- Fase de adquisición: En esta fase se adquiere toda la información relevante al objeto de emitir un adecuado diagnóstico. Cualquier patología susceptible de ser abordada mediante Neurocirugía, se apoya siempre en un estudio de TAG, RM y a veces de PET. Dichas imágenes son adquiridas y visualizadas, en la mayoría de las veces sobre placas. Sin embargo, en los centros sanitarios comienzan a adquirirse sistemas de exploración radiológica que presentan salidas digitales de imagen médica en el formato estándar DICOM 3.0. Esto permite mediante la conexión digital directa a los escáneres TAC y RM visualizar sobre un ordenador los estudios realizados. La visualización digital de estudios presenta evidentes ventajas frente a la visualización tradicional ya que permite mejorar notablemente la visión de las imágenes al aplicar técnicas de tratamiento digital de imagen y técnicas de visión 3D. Por ello, el primer paso básico y fundamental de un sistema CAO, consiste en disponer de un sistema de adquisición y visualización digital de la imagen médica.
- Fase de planificación: Durante la fase de planificación y utilizando programas de visualización digital de imagen médica, el cirujano planifica la intervención quirúrgica a efectuar posteriormente. Para ello los sistemas de planificación quirúrgica deben disponer de una serie de herramientas que permitan al clínico definir dicha fase de manera adecuada. Dichas herramientas son entre otras:
 - ✓ Visualización avanzada de las imágenes médicas: es decir visualización de varias modalidades a la vez, visión de planos de corte en el espacio en cualquier orientación, filtros de mejora de imagen, etc.
 - ✓ Segmentación de la imagen médica: reconocer y marcar en cada imagen los distintos tejidos y estructuras anatómicas relevantes (como por ejemplo hueso, tumor, vasos sanguíneos, etc.).
 - ✓ Fusión de la imagen médica: mediante esta herramienta es posible fusionar las distintas modalidades de la imagen médica utilizadas. Esta herramienta resulta imprescindible ya que cada modalidad contiene información complementaria que es conveniente ver superpuesta.
 - ✓ Visualización 3D de la imagen médica: Una vez segmentadas las distintas estructuras, resulta conveniente poder visualizar de forma tridimensional toda la anatomía del paciente para poder definir las vías de abordaje óptimas de intervención quirúrgica. Conviene disponer de herramientas de visualización 3D mediante reconstrucción por superficies y de visualización volumétrica o "volume rendering".

- Fase de intervención: Una vez realizada la fase de planificación se pasa a la fase de intervención la cual se lleva a cabo en quirófano. El sub-sistema más importante de un sistema CAO en Neurocirugía es el navegador quirúrgico también llamado neuronavegador. Dichos sistemas consisten básicamente en un digitalizador tridimensional que proporciona en tiempo real las coordenadas tridimensionales del instrumental quirúrgico utilizado. Dichas coordenadas son visualizadas en una pantalla de ordenador sobre las imágenes médicas previamente tratadas en la fase de planificación. De esta manera el cirujano ve en todo momento la posición espacial del instrumental utilizado, disponiendo de esta forma de un sistema de navegación anatómica que le va guiando durante la intervención, de ahí el nombre de navegador quirúrgico. Existen diversas técnicas de navegación quirúrgica dependiendo de la tecnología del digitalizador utilizado y de la visualización. Mediante dichas técnicas se lleva a cabo un cambio de coordenadas que permite pasar del sistema de coordenadas de imagen médica al de coordenadas del digitalizador 3D, y de este a su vez al de la pantalla del ordenador sobre el cual se visualiza la posición del instrumental quirúrgico.

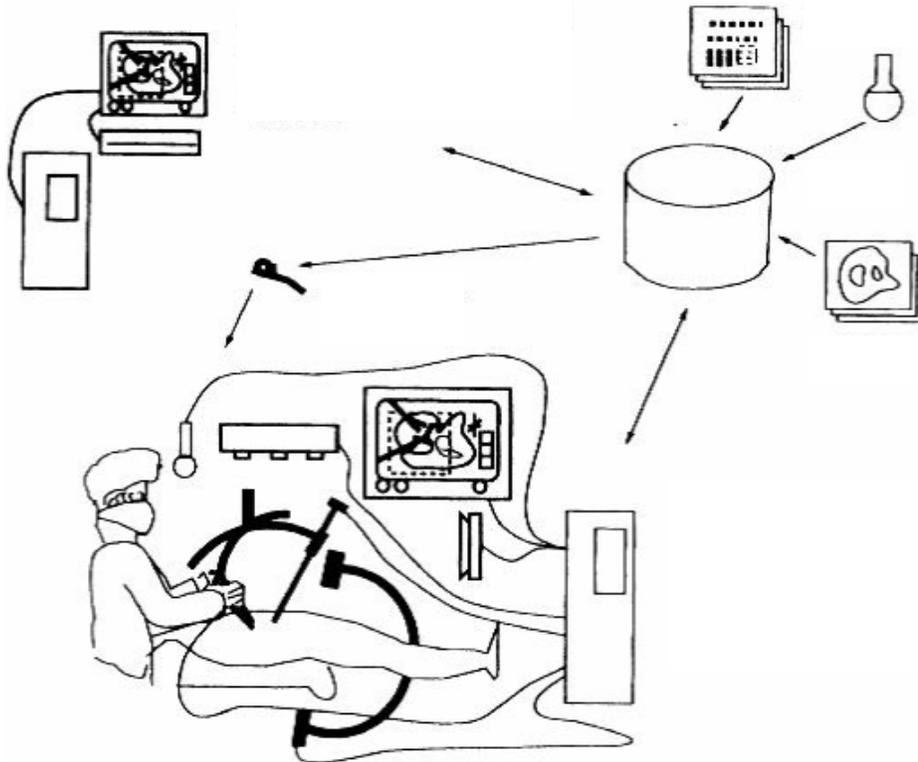


Figura 2.2. Esquema de un sistema CAO.

Este sistema utiliza un digitalizador óptico Flashpoint 5000 (IGT, Boulder, CO, USA) conectado a una estación multiprocesador IBM Intellistation con 4 procesadores Pentium III a 500 Mhz., con 512 Mb RAM sobre la cual se ejecuta el software desarrollado en lenguaje C++, bajo entorno Windows NT. El software desarrollado incluye dos módulos: el módulo de planificación quirúrgica y el módulo de navegación quirúrgica en el quirófano. El módulo de planificación permite la captura de los estudios de imagen, la visualización 3D de la anatomía, la segmentación de estructuras y la planificación de las vías de abordaje y puntos de alcance de la intervención. El módulo de navegación permite al cirujano en el quirófano estar en todo momento guiado viendo la posición del instrumental sobre una pantalla de la estación [2].

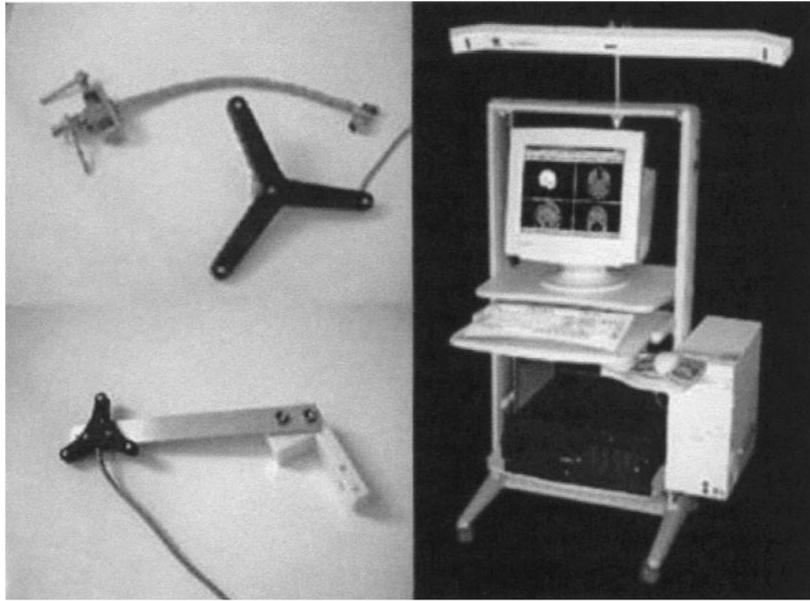


Figura 2.3. Instrumental quirúrgico adaptado y una vista general del primer prototipo del sistema.

II.4. La Tele Radiología

La transmisión de imágenes médicas entre centros quirúrgicos, ha sido desarrollada en los últimos años bajo el término de Tele radiología. En la actualidad las imágenes transmitidas se obtienen a través de procesos de digitalización de imágenes. Este procesamiento se lleva a efecto en sistemas informáticos que contienen dentro de su información un desarrollo volumétrico del objeto estudiado, despliegan, ya sea en placas radiográficas o en monitores de visiones bidimensionales multiplanares. Es por tal efecto una secuencia de fotos que interpretan en forma bidimensional una realidad holográfica, lo cual determina una pérdida de la capacidad interactiva sobre la información de la cual se origina, restringiendo su posterior utilización.

El objetivo del proyecto es desarrollar un sistema médico de transmisión de imágenes neuroquirúrgicas en tres dimensiones que permita la interacción simultánea (sincrónica) entre centros neuroquirúrgicos remotos a un bajo costo (Tele radiología virtual). Para esto se contempla el desarrollo de un sistema médico basado en una plataforma hardware de bajo costo y uso universal, y la creación de un Software Bioinformático que cumpla el rol de interfase entre dos operadores remotos que permita compartir una aplicación médica (imágenes médicas), originada en pacientes reales, con capacidad interactiva en forma cooperativa y su posterior validación clínica en las distintas instancias de interacción de redes, tanto local como a distancia.

El proyecto contempla la utilización del "Software de planificación y manipulación de imágenes médicas tridimensionales NAVIMETRIC-GENSYS", que permite utilizar la información contenida en los exámenes de Tomografía Axial Computarizada y Resonancia Nuclear Magnética, en forma directa, respetando el formato del Software originado (DICOM 3.0) y luego desplegarla tridimensionalmente para su manipulación, bajo conceptos neuroquirúrgicos, en plataformas hardware de bajo costo, en ambiente PC/Windows. Se contemplan varias etapas para llevar a cabo este proyecto.

La primera de estas etapas, considera un período de creación del software para interacción sincrónica de imágenes en tres dimensiones, utilizando como base el programa computacional de procesamiento de imágenes, Gensys-Navimetric. Esta interacción será validada mediante dos estaciones de trabajo unidas en una red local del Servicio de Neurocirugía del Hospital Carlos Van Buren, de Valparaíso. Posteriormente, se realizará la validación clínica del sistema utilizando

imágenes de Tomografía Axial Computarizada (Scanner) y de Resonancia Nuclear Magnética, enviadas mediante una red de tipo Frame Relay, desde los equipos de Scanner y Resonancia Nuclear del Hospital Gustavo Fricke y del Centro Vesalio, respectivamente.

Una vez validado localmente el sistema, se procederá a la segunda etapa del proyecto, que implica establecer una conexión a distancia Inter redes, utilizando una salida de banda ancha, mediante un enlace dedicado, entre la red del Servicio de Neurocirugía del Hospital Carlos Van Buren y la red de la Facultad de Medicina de la Universidad de Valparaíso en Chile, complementando así dicha red con una tercera estación de trabajo remota, que permita el trabajo interactivo y cooperativo, a tiempo real entre las estaciones, sobre un recurso de red preexistente y de uso masivo.

La tercera etapa nace una vez validada en términos clínicos y bioinformáticos dicha aplicación, y utilizando la conexión de Frame Relay, entre el Hospital Gustavo Fricke y el Hospital Carlos Van Buren, para la transmisión de imágenes neuroquirúrgicas de urgencia en 2D y la asistencia remota para el manejo de estos pacientes.

La cuarta etapa comprenderá la apertura de una página web con acceso abierto a un banco de imágenes tridimensionales, ya sea con acceso libre o pagado, dependiendo del nivel de accesibilidad.

El paquete tecnológico a transferir estará compuesto por el software para interacción entre centros remotos, el sitio web con biblioteca virtual de imágenes neuroquirúrgicas, red de transmisión de imágenes entre centros hospitalarios, el almacenamiento digital de imágenes, asistencia docente a distancia, cursos y seminarios de instrucción, servicio post venta y Kits de autoinstrucción. El proyecto logrará un impacto en el sistema de salud nacional, al mejorar el intercambio de información neuroquirúrgica entre hospitales y doctores. Además, permitirá un mejor manejo tanto económico, como de asistencia del paciente neuroquirúrgico de urgencia, una apertura nacional hacia el exterior, con tecnología de punta, con un sistema sin precedentes [3].

II.5. Ultrasonido 3D de Manos Libres

Este sistema consta de una estación de trabajo, un sistema de posición y un equipo de ultrasonido, figura 2.4.

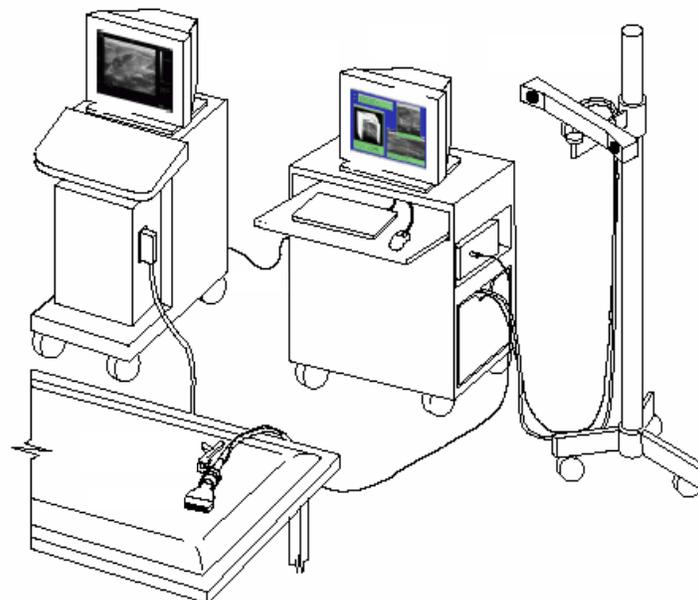


Figura 2.4. Ultrasonido 3D.

El sistema de posición consta de dos partes:

1. El receptor es instalado en el transductor del equipo de ultrasonido y se conecta en la estación de trabajo.
2. El transmisor se conecta en la estación de trabajo.

En conjunto, el transmisor y el receptor dan las coordenadas de cada imagen capturada.

La estación de trabajo recibe la imagen del equipo de ultrasonido y la coordenada de la imagen por medio del sistema de posición, al terminar la exploración (órgano, feto, etc), la estación de trabajo alinea las imágenes 2D y forma la imagen 3D, para poder observar los cortes en los diferentes planos o realizar la reconstrucción de cada imagen 2D para la detección de contornos e implementa el método Marching cubes para observar el órgano en una imagen virtual [4], figura 2.5.

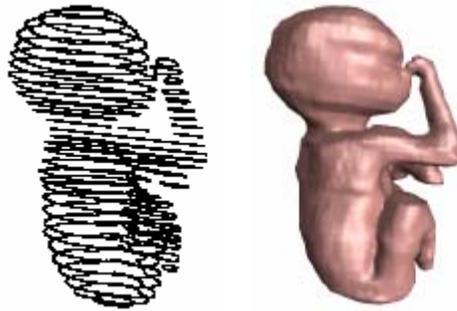


Figura 2.5. Realidad virtual.

II.6. Fusión de Imágenes 3D

Este sistema fusiona los datos de un equipo de ultrasonido 3D y las imágenes obtenidas por un equipo de resonancia magnética (MRI), sin la asistencia de marcas externas o sensores de posición externos. La fusión de estas dos modalidades combina la exploración en tiempo real 3D del ultrasonido que obtiene estructuras de tejido blando, con las grandes estructuras anatómicas obtenidas del MRI. La información complementaria disponible por la modalidad de múltiples imágenes garantiza el desarrollo de fusiones robustas.

La adquisición de datos tridimensionales se estableció para imágenes de resonancia magnética y recientemente para ultrasonido. En una implementación simple, la información de una imagen 3D puede ser obtenida de una serie de secuencias de imágenes 2D topográficas tomadas sobre un volumen.

El potencial usado para unir o fusionar imágenes 3D incluye la información de la posición del tumor del hígado para subsecuente biopsia o terapia; y la fusión de alta resolución del volumen del cerebro por el MRI, hígado o lesiones renales guiados por el ultrasonido para la resección o ablación el tumor.

Las técnicas de fusión generalmente caen dentro de una de las dos categorías: fusión de cuadros o sin cuadros. La fusión de cuadros utiliza cuadros externos, para límites de estructuras óseas o se ponen marcas manualmente. La fusión sin cuadros de tejido suave se usa en superficies de órganos y estructuras vasculares, para alinear imágenes de volúmenes.

La secuencia 3D del MRI, fue obtenida con una unidad GE Signa 1.5-T. Para imágenes del antebrazo se utilizó una exploración de superficie en forma de espiral [con dos milímetros de grosor en el corte y una matriz de 256 x 128 y 10 cm de field-of-view (FOV), produciendo 0.391 mm x 0.782 tamaño del píxel en axial y plano oblicuo axial]. Para imágenes del hígado se utilizó una

exploración del torso en forma de espiral [1.4 mm espacio entre corte, 5.6 mm de grosor del corte, una matriz de 512 x 512, 30cm x 22.5 cm FOV tamaño del píxel 0.586mm].

El volumen del ultrasonido fue obtenido con una pista motorizada que mueve el transductor del ultrasonido a causa del control, con velocidad uniforme, figura 2.6. La velocidad de la pista fue coincidir con la acústica de la tasa de cuadro logrando un espacio entre cada imagen de un milímetro ó 0.5 mm. Los datos del US CINE fueron obtenidos en una orientación sagital, 30 muestras con respecto a un espacio de 0.5 mm, 7 Mhz, modo B-scan, 5 Mhz de color doppler, 4cm de profundidad y orientación oblicua axial (abdomen) [80 muestras con un espacio de un milímetro, 5 Mhz, modo B-scan, 4 Mhz color doppler, 14 cm de profundidad] con un tiempo de adquisición de recorrido de 8 seg. a 20 seg. para 30 cuadros [5].

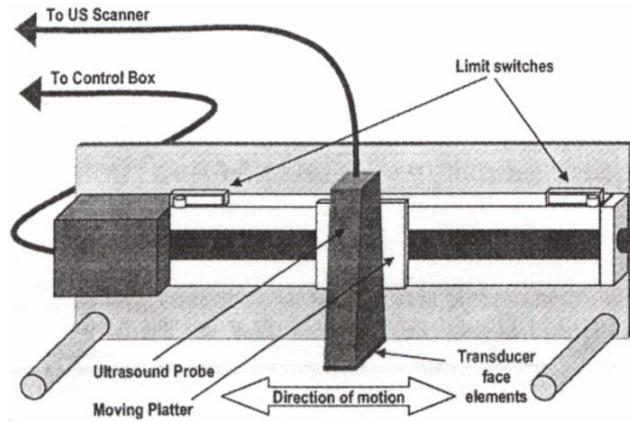


Figura 2.6. Pista motorizada.

La figura 2.7, es una imagen 3D, obtenida al fusionar la imagen 3D obtenida del equipo de ultrasonido y de la imagen 3D del MRI (resonancia magnética). La fusión se realizó con la implementación de la correlación en 3D, sin utilizar marcas en las imágenes 3D.

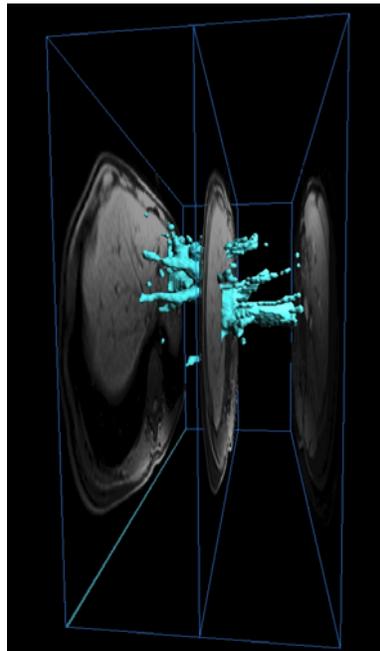


Figura 2.7. Fusión de imágenes 3D.

II.7. Reconstrucción 3D

La reconstrucción 3D es el proceso mediante el cual, objetos reales, son reproducidos en la memoria de una computadora, manteniendo sus características físicas (dimensiones, volumen y forma). Existen dentro de la visión artificial, multitud de técnicas de reconstrucción y métodos de mallado 3D, cuyo objetivo principal es obtener un algoritmo que sea capaz de realizar la conexión del conjunto de puntos representativos del objeto en forma de elementos de superficie, ya sean triángulos, cuadrados o cualquier otra forma geométrica.

Los algoritmos desarrollados hasta el momento, se debaten entre el coste computacional y la calidad del mallado obtenido. Los algoritmos que trabajan con nubes de puntos, tratan de obtener la denominada matriz de conexiones. Esta matriz, almacena que puntos del conjunto inicial deben estar conectados entre sí. Si empleamos triángulos (método bastante común), esta matriz tiene la forma de $3 \times n$ (siendo el número total de triángulos que contiene la pieza), es decir que cada fila de la matriz representa un triángulo en el plano o en el espacio, figura 2.8.

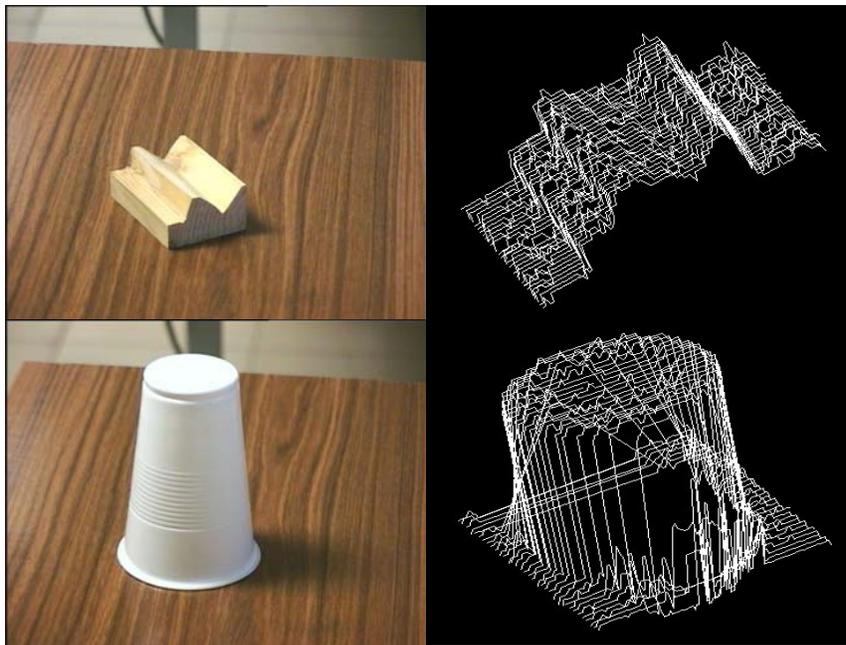


Figura 2.8. Método de mallado.

La eficiencia del algoritmo es la que define la calidad final del mallado. Si suponemos un conjunto de puntos mal representado, existirán puntos definidos que no cumplan las condiciones óptimas para el mallado. Los puntos que se encuentran muy cercanos entre sí, los puntos ruidosos y los puntos redundantes, no ofrecen ninguna información para la reconstrucción.

La reconstrucción tridimensional, se realiza partiendo de un conjunto de puntos demasiado grande, en el cual, una gran parte de los puntos no ofrecen ninguna información y deben ser eliminados.

La adquisición de los puntos, se ha realizado mediante un sistema de Luz Estructurada (Range Image), figura 2.9, el cual se encarga de realizar una serie de cortes a la pieza, u objeto que se desea reconstruir, mediante un plano láser. Un sensor CCD o cámara, se encarga de ir recogiendo las imágenes que se producen cuando el plano láser incide sobre la pieza. Después, estas imágenes son debidamente procesadas aplicando una serie de algoritmos cuyo resultado final es la obtención de la nube de puntos del objeto.

La calibración del sistema se realiza con un algoritmo bastante común, método de Tsai, el cual proporciona los parámetros intrínsecos de la cámara, necesarios para realizar el cambio, de coordenadas de la cámara 2D a coordenadas del mundo 3D [6].

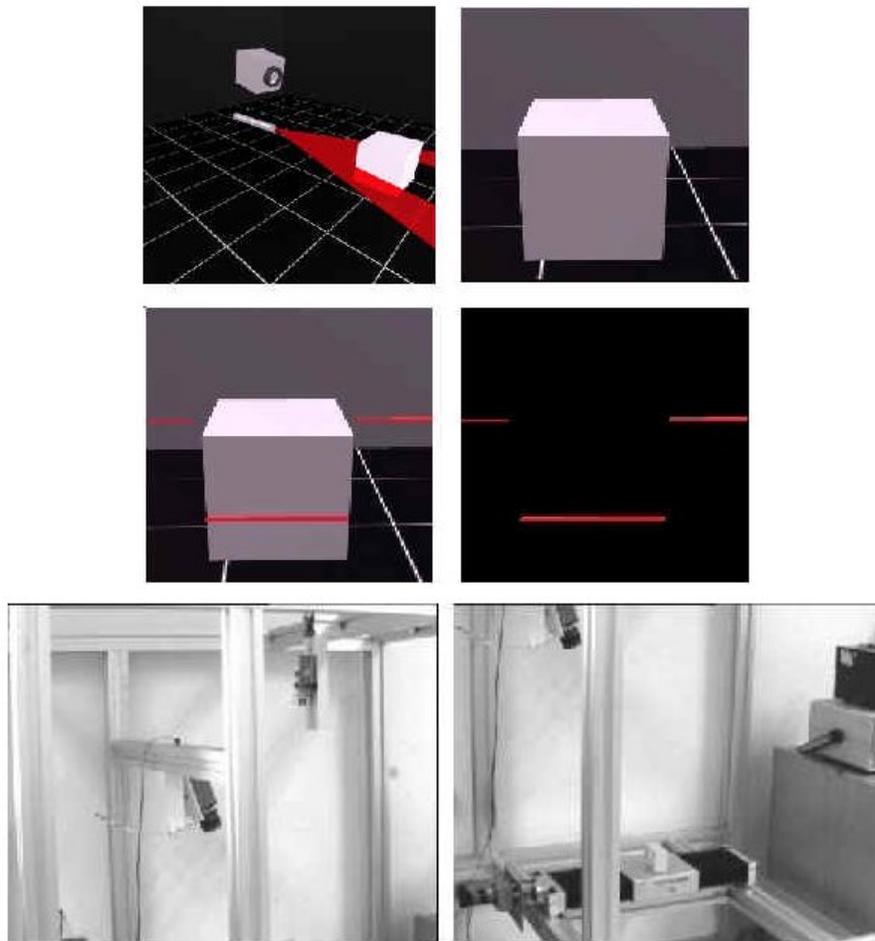


Figura 2.9. Sistema de luz estructurada.

II.8. Sistema de Navegación en 3D

En la actualidad, la información visual que ofrecen la mayoría de los sistemas de guiado, se presenta en dos dimensiones, a través de pantallas de color o monocromáticas. Algunos fabricantes de coches (como Nissan) y proveedores de hardware y equipos (Siemens VDO o Pioneer) van un paso más allá y cuentan con navegadores cuyas imágenes se ven en tres dimensiones, aunque bastante simplificadas, figura 2.10. Por ejemplo, no se ofrecen perspectivas de los edificios y sólo aparecen representadas las carreteras.

Las últimas evoluciones en este campo todavía se encuentran en fase de estudio, pero ofrecen un grado de realismo muy cercano a lo que el conductor vería con sus propios ojos. Con un navegador de este tipo, las instrucciones serán tan claras e inequívocas como, "después de la iglesia, gire a la izquierda".

En cualquier caso, la verdadera revolución se producirá a partir del 2008. En ese año, se espera que esté completamente desplegada la constelación de 30 satélites que formarán Galileo, el programa de navegación desarrollado por la Unión Europea y la Agencia Espacial Europea. A través de receptores de tipo dual (los actuales no servirán), el conductor podrá sumar a la información que en estos momentos recibe del GPS estadounidense, la de los satélites de Galileo.

De esta forma, el guiado será más preciso (con un error de apenas unos centímetros), y se dispondrá de una alternativa por si falla uno de los dos sistemas. Además, el europeo no tiene ninguna limitación de tipo militar [7].



Figura 2.10. Imagen adquirida por el sistema de navegación en 3D.

II.9. Conclusiones

Las imágenes en 3D nos permiten realizar tareas y procesos con mayor seguridad, precisión y rapidez; las cuales no se pueden realizar con secuencias de imágenes de 2D, ni mucho menos con una imagen de 2D.

Los problemas a los que nos enfrentamos para poder introducirnos al nuevo campo de aplicación de las imágenes en 3D son:

- 1) Los sistemas y equipos para adquirir imágenes en 3D son muy complejos y costosos. Esto se debe al tipo de sensores y equipos que se utilizan.
- 2) La mayoría de los sistemas para adquirir imágenes en 3D realizan alguna tarea específica; estos se deben a que son desarrollados por universidades, clínicas o laboratorios, para fines particulares; por lo cual, no son sistemas comerciales.
- 3) El tiempo de captura y el procesamiento de las imágenes en 3D, es mucho mayor que el de las imágenes en 2D, esto se debe:
 - Utilizan varios sensores sincronizados.
 - Necesitan n imágenes en 2D, para formar una imagen en 3D.
 - Los mejores sistemas para adquirir imágenes en 3D, fusionan imágenes de 3D de diferentes equipos.
 - Sincronización de los sensores y equipos para adquirir las imágenes en 3D.
 - Supresión de ruido en las imágenes en 3D.
 - Umbralización.
 - Detección de líneas y contornos.
 - Reconstrucción de las imágenes en 3D.
 - Rotación de las imágenes en 3D.
 - Convolución, correlación, etc.

Tenemos que tener en cuenta un punto muy importante: la captura de la imagen en 3D.

El sistema que se utilice para realizar la captura de la imagen en tres dimensiones, nos va a señalar:

- Las herramientas adecuadas para la construcción de la imagen en 3D, y una mejor implementación de las técnicas de segmentación de imágenes.
- Los límites de nuestro sistema, que implementa esta imagen en 3D para alguna aplicación específica.

Una imagen en 3D nos permite observarla desde cualquier ángulo (imagen virtual) o realizar cortes de la imagen, en los planos **xy**, **yz** o **xz**, para observar su interior; lo que nos permite tener nuevas herramientas.

CAPITULO III. SISTEMA DESARROLLADO PARA CAPTURAR IMÁGENES EN 3D

III.1. Introducción

En este capítulo se presenta el sistema experimental que se diseñó y desarrolló para capturar imágenes 3D. La idea principal para obtener una imagen de 3D es capturar una secuencia de n imágenes 2D, teniendo entre cada imagen un punto de referencia y una distancia constante [5].

El sistema experimental descrito a continuación utiliza un equipo de ultrasonido y un carril de referencia, donde la función es capturar la secuencia de las n imágenes 2D, teniendo una distancia y altura constante entre cada una de ellas. Estos parámetros de referencia se consideran al momento de unir las n imágenes 2D y construir una imagen de 3D con las dimensiones reales del órgano explorado.

El sistema experimental está constituido por 3 bloques:

- 1) Bloque de captura.
- 2) Bloque de sincronización.
- 3) Bloque de exploración.

A continuación se describe cada bloque, y posteriormente se muestran las imágenes de los resultados obtenidos del sistema experimental.

III.2. Bloque de Captura

Este bloque consta de un equipo de ultrasonido GE, modelo LOGIQ α 200, configurado en el modo B scan [8], figura 3.1, una tarjeta de captura PINNACLE SYSTEMS [9], y una computadora. Este bloque digitaliza las imágenes del equipo de ultrasonido por medio de la tarjeta de captura y las almacena en la computadora, figura 3.2.

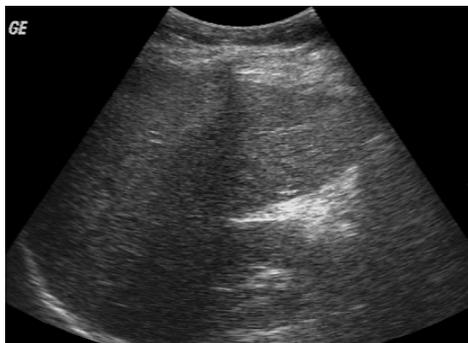


Figura 3.1. Modo B scan.

El equipo de ultrasonido tiene una salida de video compuesto, que se conectó en la entrada del video compuesto de la tarjeta de captura. La tarjeta de captura se instaló internamente en la computadora. Este bloque captura video (VCD, SVCD y DVD) e imágenes en formato BMP en escala de grises.

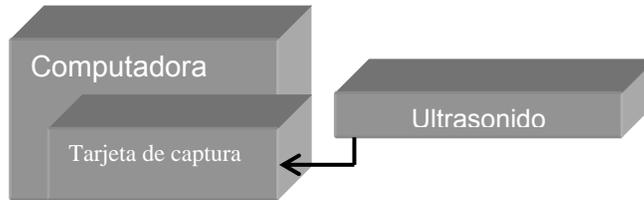


Figura 3.2. Bloque de captura.

III.3. Bloque de Sincronización

Este bloque sincroniza la captura de las imágenes del ultrasonido y el movimiento del motor paso a paso, manteniendo una distancia constante entre cada imagen y evitando capturar mas de una imagen en un mismo punto, figuras 3.3 y 3.4.

Este bloque está compuesto por tres módulos (anexo A):

- 1) Módulo de control:
 - Microcontrolador PIC16F84 [10].
 - Dos push buttons y tres leds que forman el menú de selección.
 - Sistema para activar la cámara digital de la tarjeta de captura [9].
 - Etapa de optocopladores que protegen al microcontrolador de la etapa de potencia [11].
- 2) Módulo de acondicionamiento de señal:
 - Cuatro amplificadores operacionales en configuración de multiplicador no inversor de ganancia constante [12-13].
- 3) Módulo de potencia:
 - Cuatro transistores de potencia y un disipador [14].

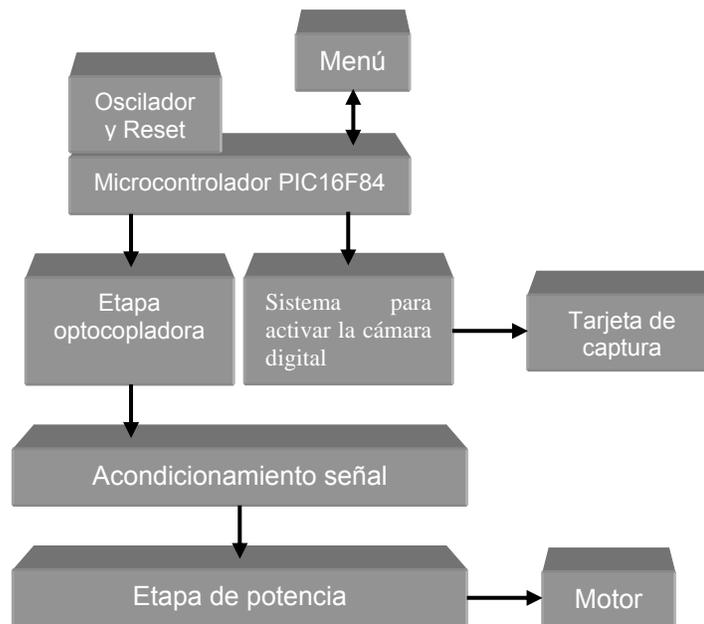


Figura 3.3. Bloque de sincronización.

El microcontrolador se programó para manejar dos modos de exploración o de barrido:

- Modo 1: $d = 0.63 \text{ mm}$.
- Modo 2: $d = 0.3 \text{ mm}$.

Los modos 1 y 2 son las opciones de distancia entre cada imagen. Todas las exploraciones se realizaron en el modo 1, debido al tiempo de barrido y sincronización. Estas distancias se tomaron de referencia del artículo "Three-Dimensional Registration and Fusion Of Ultrasound and MRI Using Major Vessels as Fiducial Markers", ellos utilizan una distancia de 1 mm ó 0.5 mm entre cada imagen [5].



Figura 3.4. Bloque de sincronización.

III.4. Bloque de Exploración

Este bloque está constituido por el carril de referencia y el transductor del equipo de ultrasonido a 3.5 Mhz. [8], figuras 3.5 y 3.6.

El carril de referencia es una estructura de plástico rígido de 41.5 cm de largo y 11 cm de ancho. En esta estructura se instalaron dos carriles de metal, un motor paso a paso, una banda con dientes, una polea y una base de plástico.

Los carriles se instalaron a lo largo de la estructura de plástico a una distancia de 5.5 cm entre ellas (en paralelo). En los carriles se instaló la base de plástico que corre libremente. En esta base se fijó de un lado la banda con dientes y por el otro lado una abrazadera donde se instaló el transductor del equipo de ultrasonido. De un extremo de la estructura de plástico se instaló la polea para ejercer una tensión sobre la banda y en el otro extremo el motor paso a paso [15].

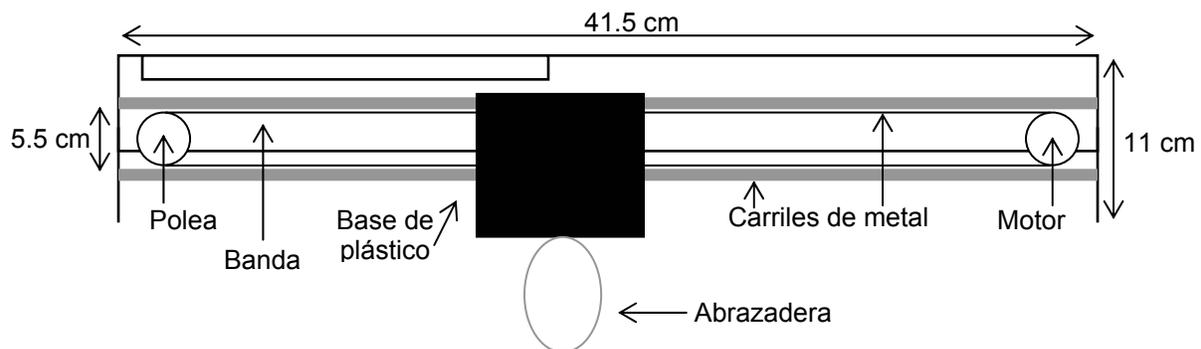


Figura 3.5. Carril de referencia.

El carril mantiene la altura constante del transductor del equipo de ultrasonido. El carril corresponde al eje **Z**, con el eje **X** y **Y** constantes.



Figura 3.6. Bloque de exploración.

III.5. Proceso para Capturar Imágenes en 3D

La idea principal del proceso del sistema experimental consiste en desplazar el transductor del ultrasonido en un carril y adquirir una imagen 2D en cada punto, las cuales se almacenan en una computadora para después unir las n imágenes bidimensionales y formar la imagen de tres dimensiones [5], figura 3.7.

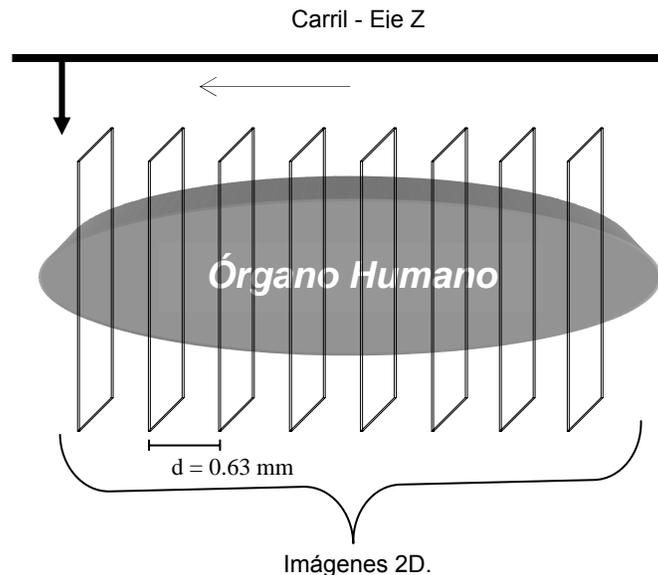


Figura 3.7. Exploración o barrido.

El proceso del sistema es el siguiente: antes de iniciar el barrido se selecciona el modo deseado (modo 1 ó 2), inmediatamente la tarjeta captura y almacena la imagen del equipo de ultrasonido que está en ese punto, el microcontrolador entra en un estado de espera para dar tiempo a la captura y almacenamiento de la imagen, secuencialmente el motor da un paso y se repite el proceso hasta terminar la exploración planeada, figura 3.8.

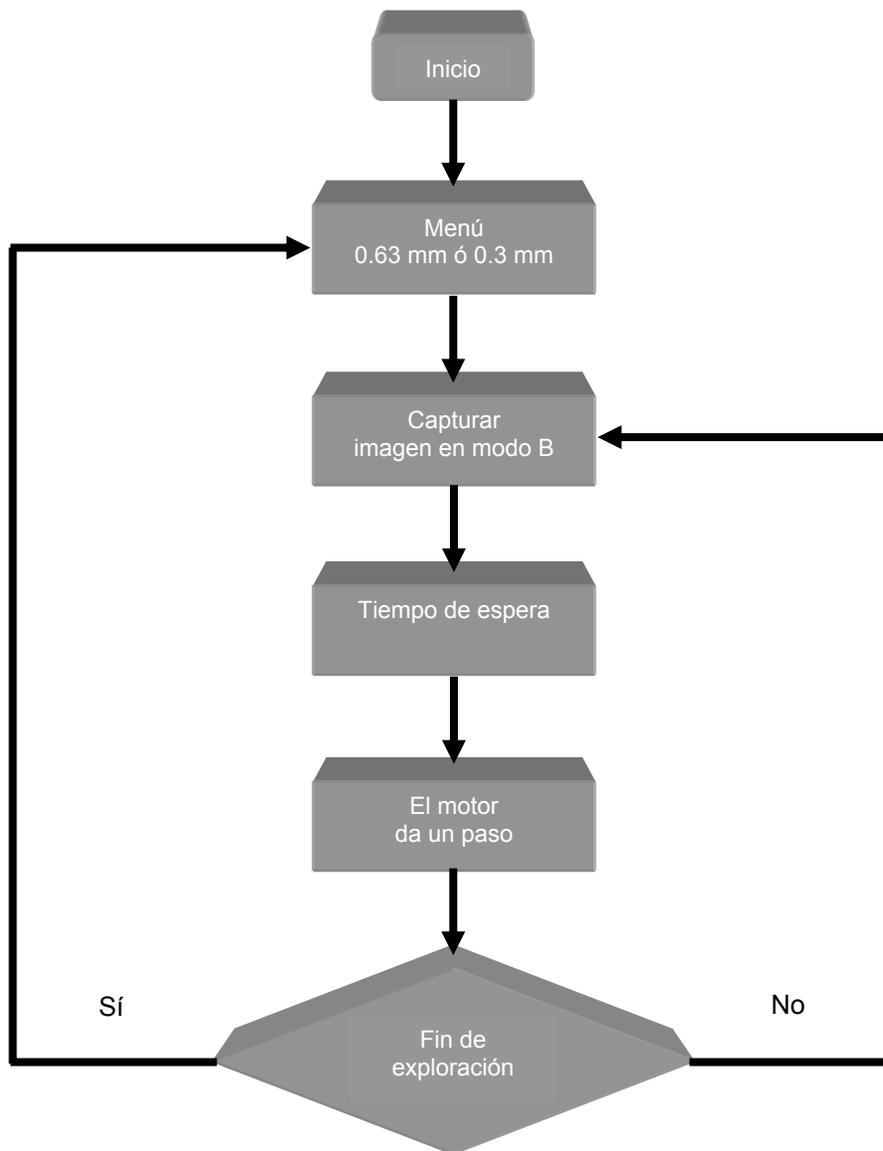


Figura 3.8. Diagrama a bloques de la captura de imágenes 2D.

El sistema consta de un grado de libertad del control del movimiento del transductor. Se diseñó para realizar exploraciones en línea recta (cortes transversales y longitudinales).

Antes de iniciar cada exploración con el sistema experimental (figura 3.9), el radiólogo localizaba el inicio del órgano y la trayectoria adecuada para realizar una exploración longitudinal o transversal de este.

Al tener la imagen de tres dimensiones se realizaron cortes en los planos **YZ** y **XZ**, para observar el interior del órgano humano explorado.



Figura 3.9. Sistema experimental para capturar imágenes de 3D.

Cabe mencionar que se diseñó y se instaló una estructura de metal, que se probó en el primer grupo de pruebas, para sostener y mantener en una altura fija, sin movimiento alguno, el carril de referencia. Esta estructura no funcionó debido a que no tenía una movilidad suave, causando muchas complicaciones al momento de iniciar la exploración, figura 3.10.



Figura 3.10. Estructura metálica.

Se realizaron varias pruebas donde los órganos explorados fueron: los riñones, el hígado, la vesícula biliar y el músculo del antebrazo derecho. En la figura 3.11 se observan los órganos explorados y la dirección de los cortes en el momento de realizar los barridos [16].

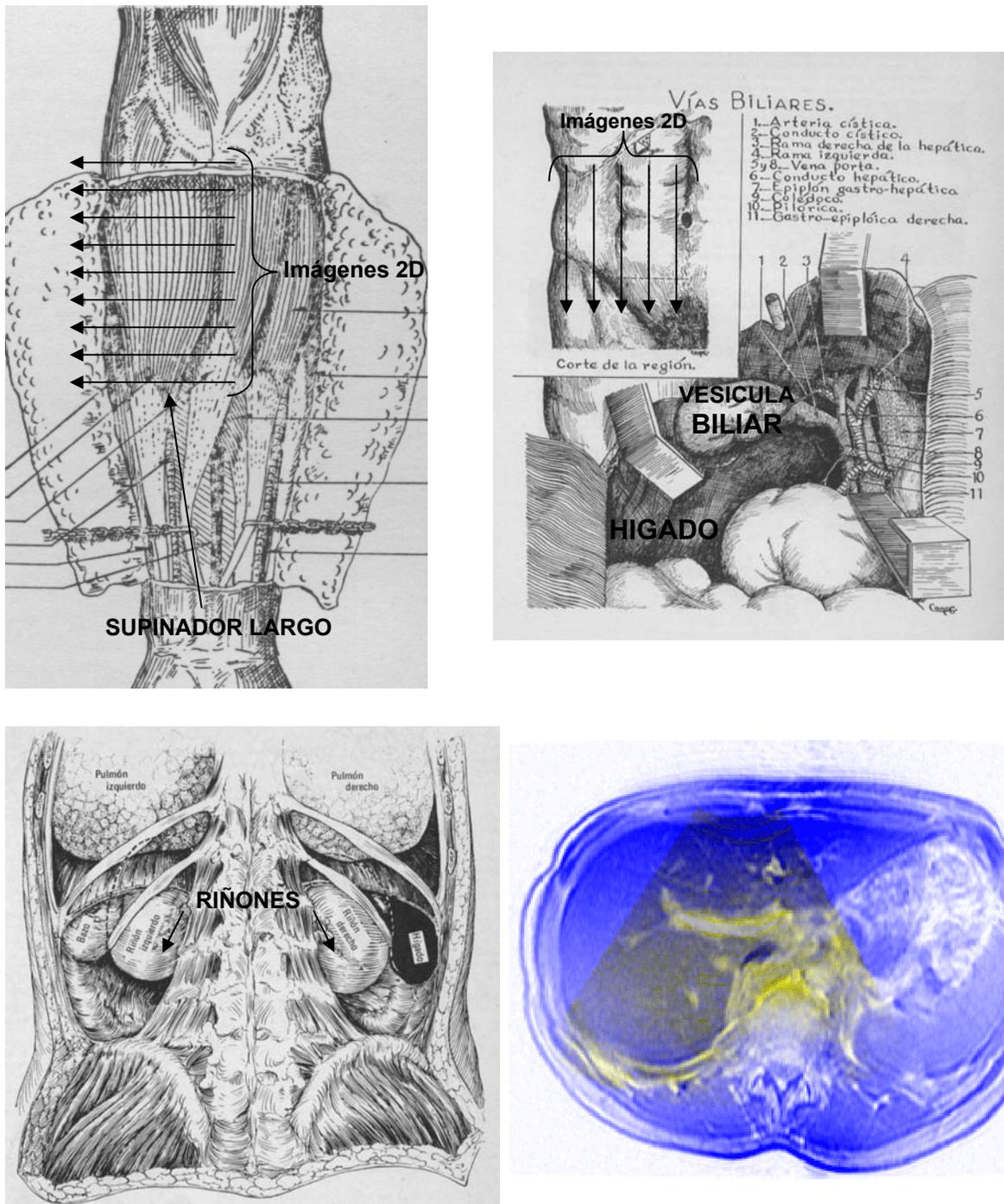


Figura 3.11. Figura de los órganos explorados y forma de exploración del transductor.

En el primer grupo de pruebas se obtuvieron mas de 700 imágenes 2D, donde únicamente 12 imágenes cumplían con los requisitos para formar una imagen 3D (secuencia correcta del órgano).

En el segundo grupo de pruebas, se realizó con dos pacientes del sexo femenino, obteniendo:

- I. Con el primer paciente se obtuvieron 12 barridos (12 imágenes 3D):
 1. Barrido 1: 117 imágenes.
 2. Barrido 2: 20 imágenes.
 3. Barrido 3: 21 imágenes.
 4. Barrido 4: 9 imágenes.
 5. Barrido 5: 13 imágenes.
 6. Barrido 6: 91 imágenes.
 7. Barrido 7: 14 imágenes.
 8. Barrido 8: 18 imágenes.
 9. Barrido 9: 20 imágenes.
 10. Barrido 10: 22 imágenes.
 11. Barrido 11: 14 imágenes.
 12. Barrido 12: 99 imágenes.

- II. Con el segundo paciente se obtuvieron 15 barridos (15 imágenes 3D):
 1. Barrido 1: 34 imágenes.
 2. Barrido 2: 26 imágenes.
 3. Barrido 3: 25 imágenes.
 4. Barrido 4: 25 imágenes.
 5. Barrido 5: 28 imágenes.
 6. Barrido 6: 102 imágenes.
 7. Barrido 7: 47 imágenes.
 8. Barrido 8: 20 imágenes.
 9. Barrido 9: 71 imágenes.
 10. Barrido 10: 54 imágenes.
 11. Barrido 11: 114 imágenes.
 12. Barrido 12: 47 imágenes.
 13. Barrido 13: 20 imágenes.
 14. Barrido 14: 28 imágenes.
 15. Barrido 15: 12 imágenes.

A continuación; se presenta la primera imagen 3D (12 imágenes) obtenida con el sistema experimental, y 2 imágenes 3D del segundo grupo, con sus respectivos cortes en los planos **XZ** y **YZ**, figura 3.12.

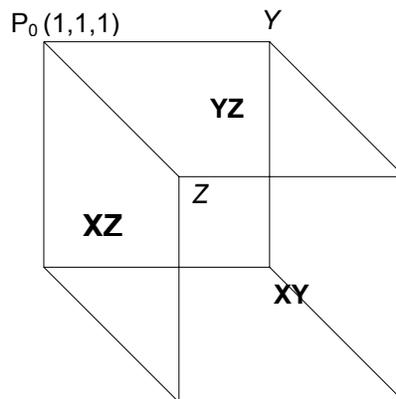


Figura 3.12. Planos donde se realizaron los cortes de la imagen 3D.

III.6. Imágenes 3D Obtenidas del Sistema Experimental

Se presenta la primera secuencia obtenida que consta de 12 imágenes 2D. Para poder observar los cortes en los diferentes planos, cada imagen se repitió 5 veces y se separó cada grupo con otras 5 imágenes de color negro. Las imágenes son de la exploración longitudinal de una sección del hígado y del riñón derecho, figura 3.13.

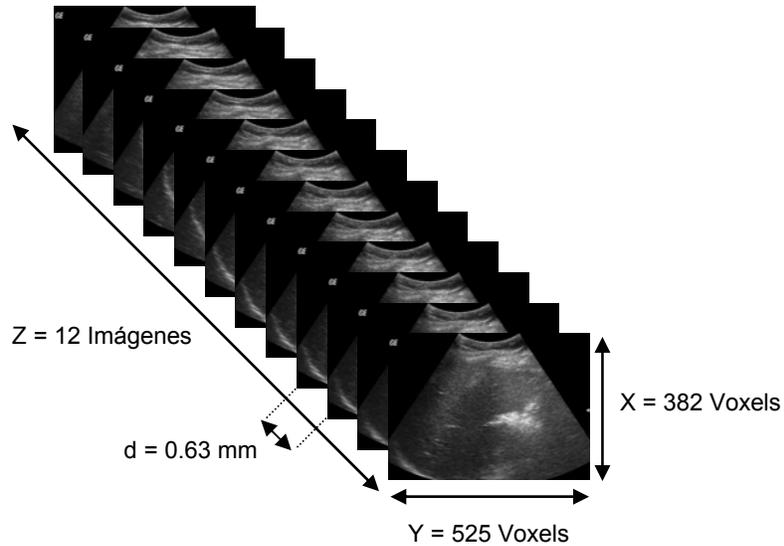


Figura 3.13. Formando la imagen 3D.

Las siguientes imágenes son de los diferentes cortes realizados en los planos **YZ** y **XZ** de la imagen de 3D que se formó de las 12 imágenes anteriores.

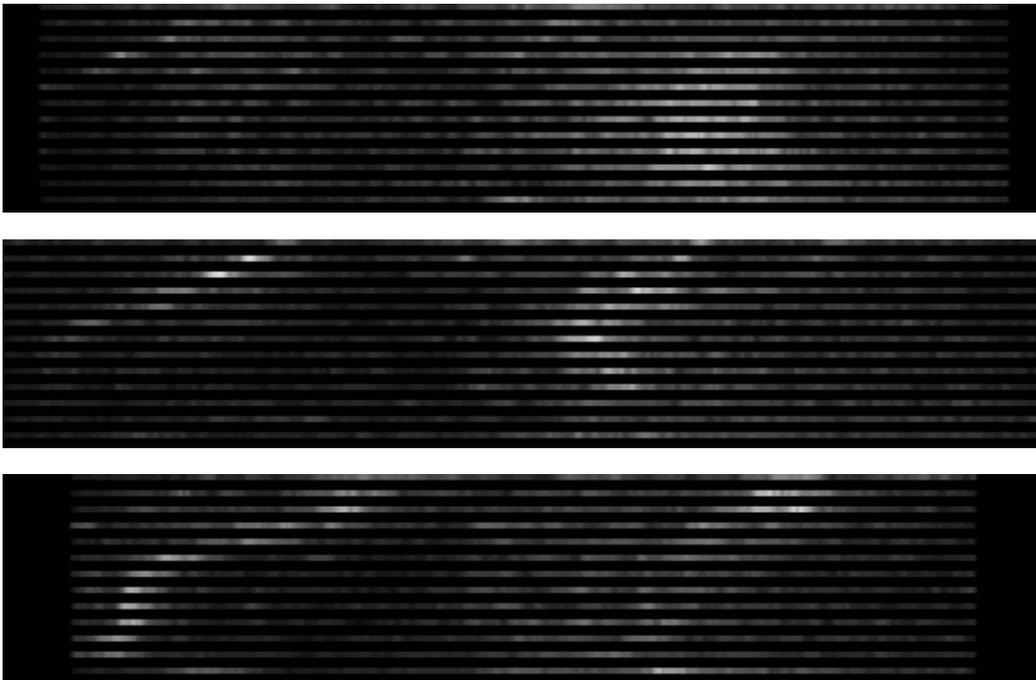


Figura 3.14. Imágenes de los cortes en el plano YZ.

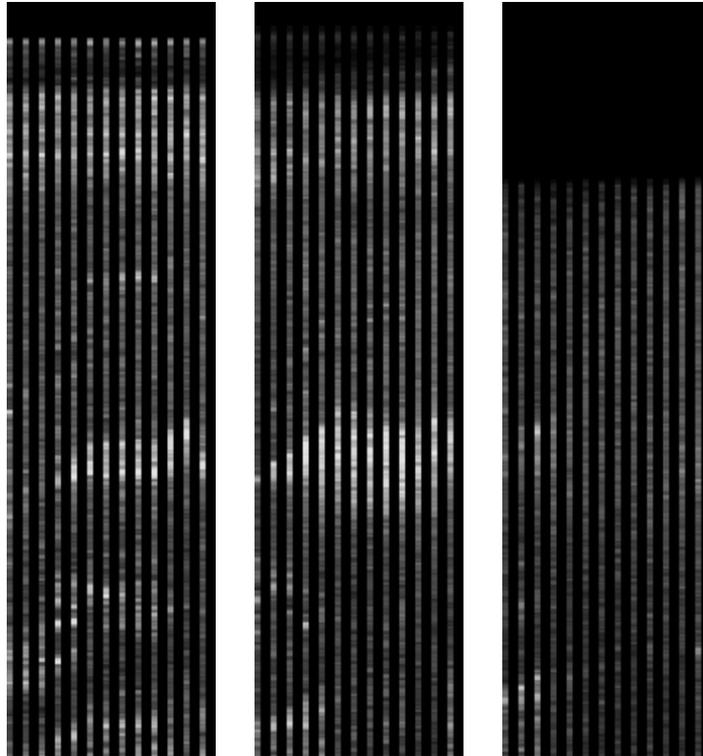


Figura 3.15. Imágenes de los cortes en el plano XZ.

La siguiente imagen pertenece a la imagen 3D de 640x480x91 voxels. Este barrido se realizó al hígado y riñón, figura 3.16.



Figura 3.16. Imagen del hígado y riñón.

Las siguientes imágenes son de los cortes en los diferentes planos de la imagen 3D.

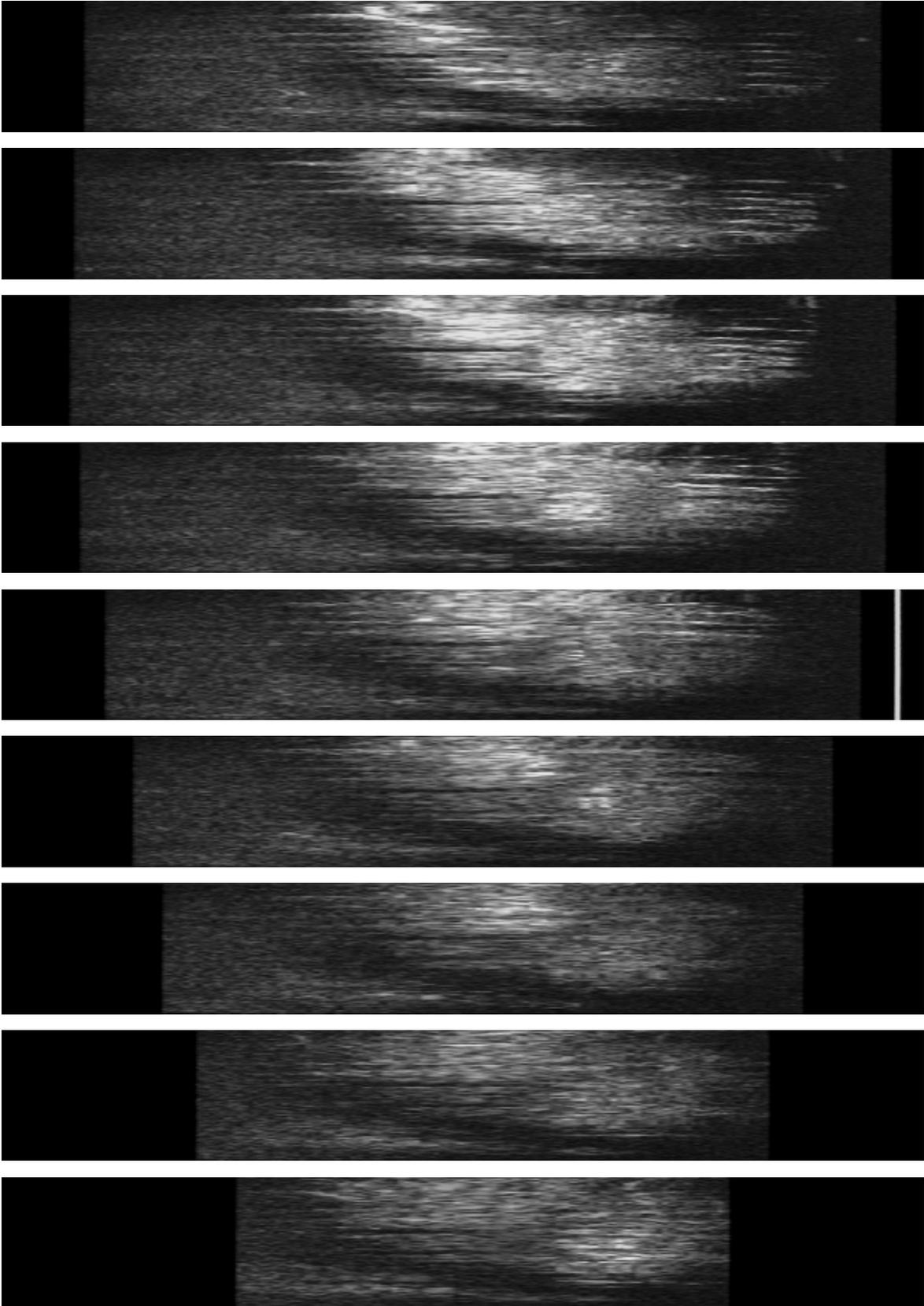


Figura 3.17. Imágenes de los cortes en el plano YZ.

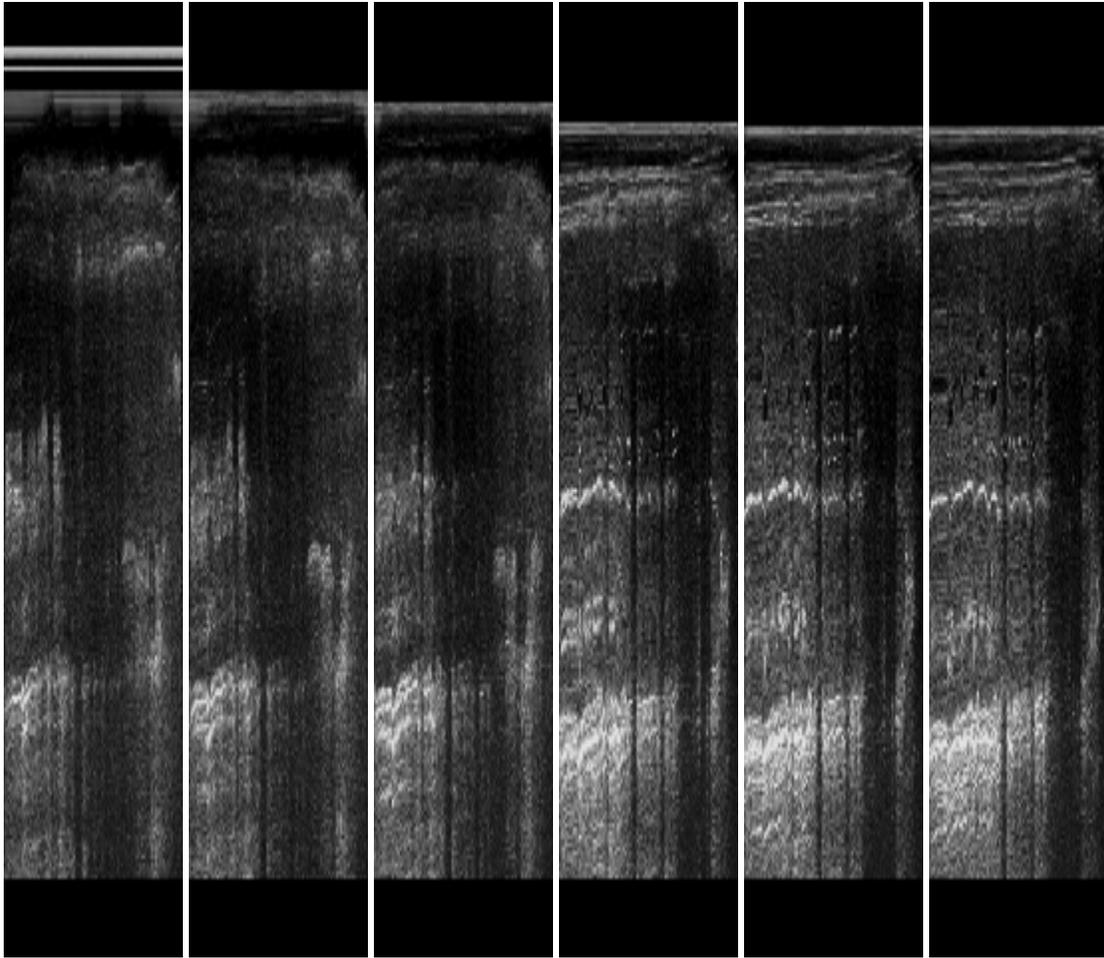


Figura 3.18. Imágenes de los cortes en el plano XZ.

La siguiente imagen pertenece a la imagen 3D de 640x480x99 voxels. Este barrido se realizó al músculo del antebrazo derecho, figura 3.19.



Figura 3.19. Imagen del músculo del antebrazo.

Las siguientes imágenes son de los cortes en los diferentes planos de la imagen 3D.

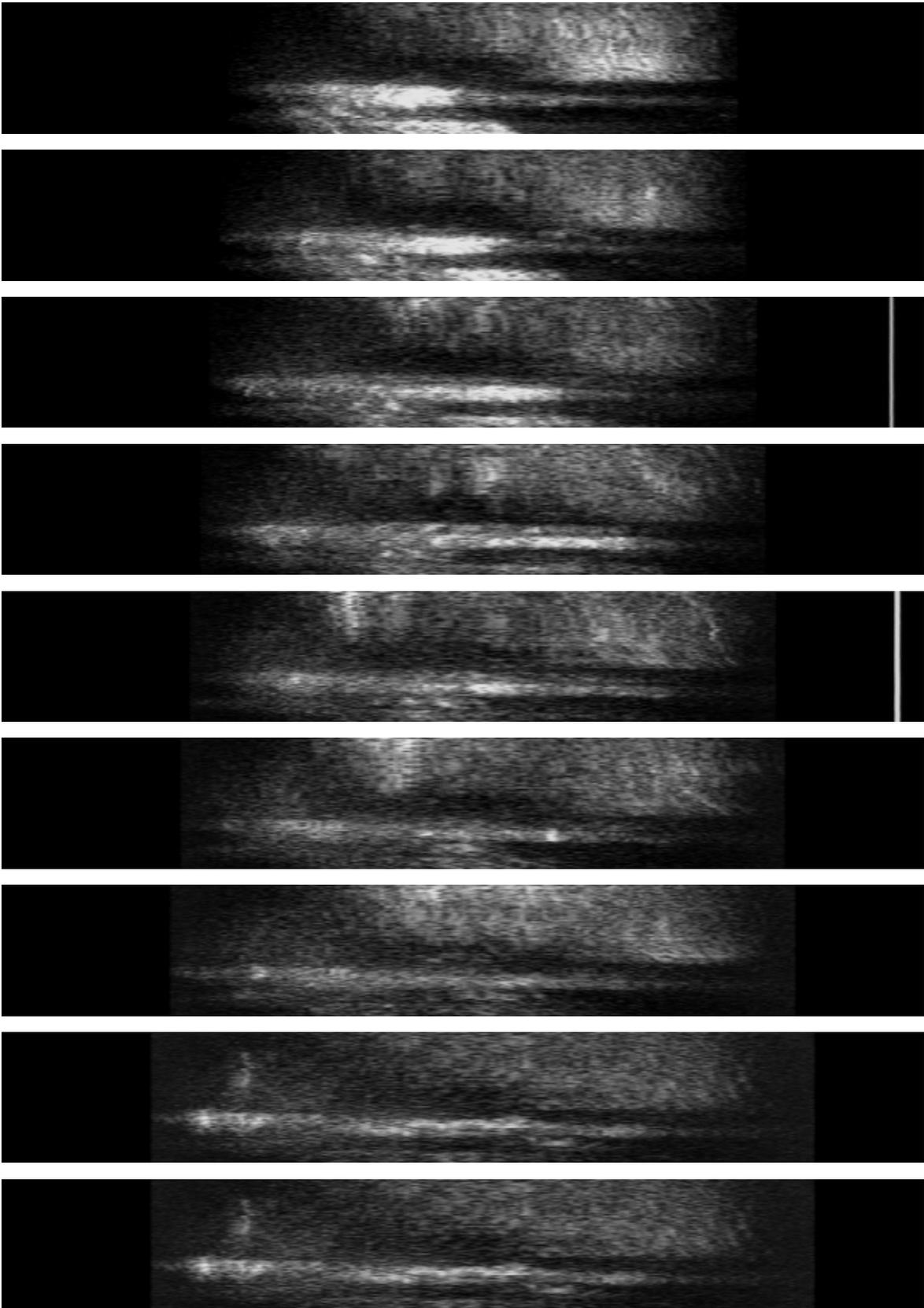


Figura 3.20. Imágenes de los cortes en el plano YZ.

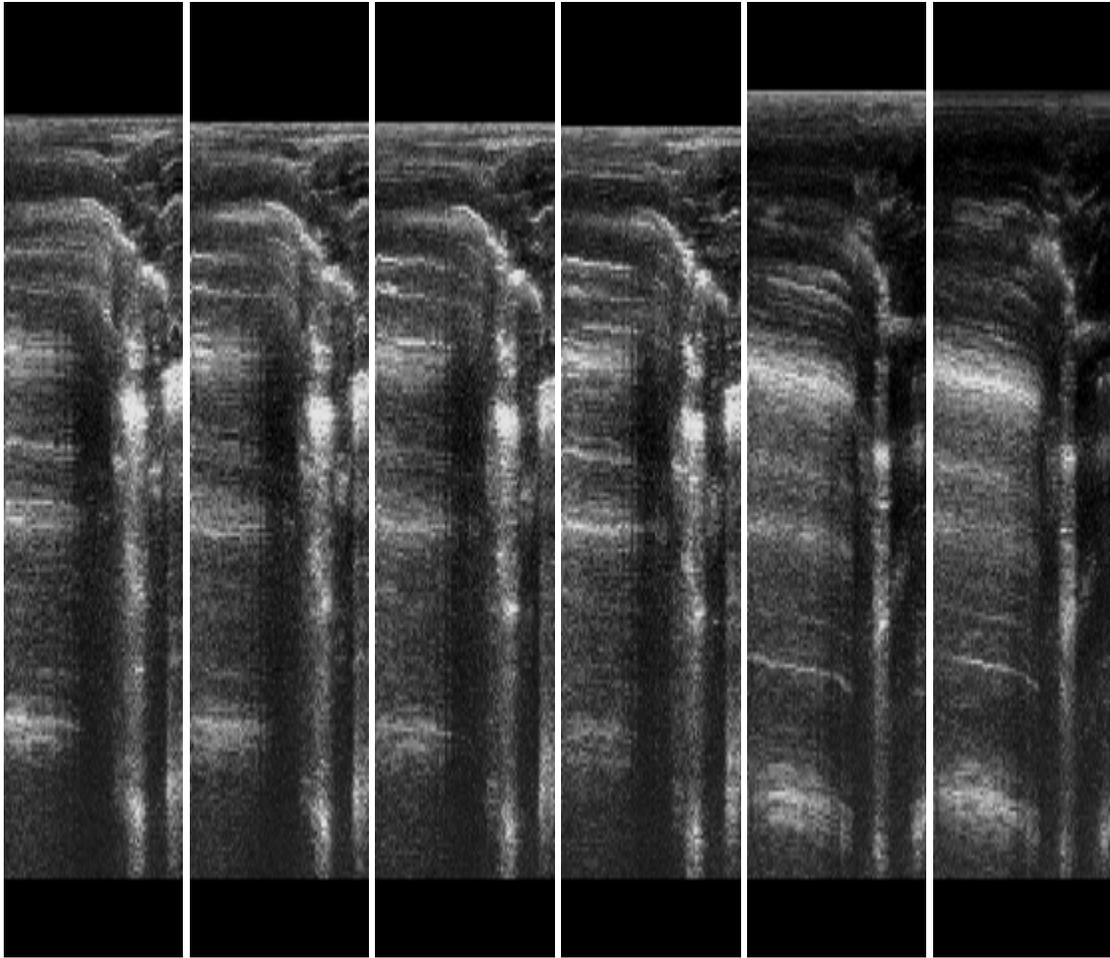


Figura 3.21. Imágenes de los cortes en el plano XZ.

III.7. Conclusiones

Se diseñó y presentó el sistema experimental que nos permitió introducirnos al nuevo campo de captura de las imágenes 3D, por primera vez aplicado en México.

La captura de imágenes 3D con el equipo de ultrasonido, es más difícil, que con los equipos de tomografía o resonancia magnética. La dificultad está en el movimiento y calibración del transductor o sensores. En los equipos de tomografía o resonancia magnética los sensores ya están instalados en una estructura fija y calibrados para cualquier tipo de exploración. En el equipo de ultrasonido el transductor se tiene que calibrar por cada exploración, para poder planear una trayectoria adecuada para la captura de la imagen 3D.

En las figuras 3.14 y 3.15, se observan las imágenes de los cortes en los planos YZ y XZ de la primera imagen 3D capturada con el sistema experimental, estas imágenes carecen de nitidez, debido a problemas del sistema. Estos problemas se resolvieron y se realizó un segundo grupo de pruebas, donde se obtuvieron los mejores resultados, con barridos mayores a 90 imágenes, obteniendo mayor nitidez.

Hay dos opciones para visualizar una imagen 3D:

1. Realizando los cortes en los diferentes planos.

2. Imagen 3D virtual; que se obtiene utilizando el método de Delaunay. Este método, primero detecta las líneas y contornos de cada imagen 2D, y con estas imágenes se forma una matriz 3D. A la matriz 3D; se le distribuyen puntos (aleatoriamente, o con alguna función de distribución). Con los puntos distribuidos se forman miles o millones de tetraedros, formando una imagen 3D virtual, ya que sus dimensiones, volumen y forma (frecuencias altas) se conservan, pero se pierden los detalles finos (frecuencias bajas).

Como se observó en las imágenes anteriores, se optó por la opción 1. Este es el método de visualización más usado por los radiólogos ya que se pueden observar los detalles finos de la imagen.

CAPITULO IV. FILTROS NO LINEALES EXISTENTES APLICADOS A IMÁGENES 2D

IV.1. Introducción

En este capítulo se presentan los filtros no lineales seleccionados en la literatura, de acuerdo a sus características. Se describen las características de cada filtro, resaltando sus ventajas y desventajas al implementarse al suprimir ruido de diferente naturaleza en imágenes 2D.

IV.2. Filtros Lineales Versus Filtros No Lineales

Los filtros lineales fueron las primeras herramientas que se utilizaron en el procesamiento digital de señales, debido a su fácil diseño e implementación. Los filtros lineales ofrecen un desempeño satisfactorio cuando las imágenes se procesan en el dominio de la frecuencia, como por ejemplo: Watermarking, estenografía, o cuando se desea suprimir el ruido de una imagen que se encuentra en una determinada frecuencia o ancho de banda [17, 18].

Los filtros lineales tienen un pobre desempeño en presencia de ruido que no es aditivo, tal como en problemas donde existen sistemas con no linealidades o en estadísticas no Gaussianas. Si se emplean los filtros lineales para suprimir el ruido en imágenes, suavizan o emborronan la imagen suprimiendo las líneas y contornos, sin suprimir el ruido. Es también conocido que, aunque las características exactas de nuestro sistema visual no son bien entendidas, resultados experimentales indican que los primeros niveles de procesamiento de nuestro sistema visual poseen características no lineales [19].

Por estas razones, las técnicas de filtrado no lineal para el procesamiento de señales/imágenes fueron consideradas desde 1958 [19]. El filtrado no lineal ha tenido un desarrollo dinámico desde entonces. Este es un indicativo por la cantidad de investigaciones publicadas y por la popularidad que han tenido los filtros no lineales en una gran variedad de aplicaciones, notablemente en telecomunicaciones, procesamiento de imágenes en el área de procesamiento de señales geofísicas y en medicina [19]. Si consideramos que los filtros no lineales tienen un mejor desempeño que los filtros lineales, es debido a que sus ventajas han sido demostradas en problemas de filtrado donde [20]

- los filtros puedan remover el ruido, especialmente el ruido impulsivo;
- la señal y el ruido ocupan la misma banda de frecuencia.

El filtrado no lineal ha sido exitoso en resolver este tipo de problemas. Este filtrado también tiene un muy buen desempeño en análisis de imágenes. Esto es debido a que el filtrado no lineal minimiza los requerimientos de linealidad, por lo que diferentes clases de filtros no lineales nunca serán peores que los filtros lineales.

A continuación se presentan los diferentes filtros no lineales seleccionados de acuerdo a sus características.

IV.3. Filtro α - Trimmed Mean

El filtro de mediana y el filtro de promedio tienen propiedades deseables e indeseables. Por ejemplo, el filtro de mediana suprime bien el ruido impulsivo, pero en el caso del ruido Gaussiano es menos eficiente. Por lo contrario, el filtro de promedio se desempeña mejor para suprimir el ruido Gaussiano, pero deteriora gramáticamente la imagen al tratar de suprimir el ruido impulsivo.

Por consiguiente los filtros Trimmed Mean utilizan las características de la mediana para suprimir ruido impulsivo y del promedio o media aritmética para suprimir ruido Gaussiano [19-21].

La primera forma del filtro Trimmed Mean, es el (r, s) – fold trimmed mean. En este filtro se ordenan los datos de la ventana de filtrado y se omiten todas las muestras con subíndices $r + s$ para $X_{(1)}, X_{(2)}, \dots, X_{(r)}$ y $X_{(n-s+1)}, X_{(n-s+2)}, \dots, X_{(n)}$. Entonces la salida es generada por el promedio de las muestras escogidas para el cálculo [19-21]:

$$TrMean = \frac{1}{n - r - s} \sum_{i=r+1}^{n-s} X_{(i)} \quad (4.1)$$

Si el número de elementos recortados de ambos extremos son iguales, $r = s$, la suma de los elementos recortados es a menudo especificada por la proporción de elementos recortados. Esta proporción es típicamente denotada por $\alpha = j / N$, donde $0 \leq j \leq N / 2$ es un entero. El número de αN muestras son el recorte de cada extremo. Ese filtro es llamado α -trimmed mean [19-21].

Los filtros α -trimmed mean tienen un compromiso dado entre los filtros medianos y de promedio. Estos filtros no logran preservar los contornos y detalles finos, pero a diferencia del filtro de promedio no emborronan las imágenes [19-21].

$$\alpha TM = \frac{1}{N - 2\alpha N} \sum_{i=\alpha N+1}^{N-\alpha N} X_{(i)} \quad (4.2)$$

IV.4. Filtro Modified Trimmed Mean

En el filtro α -trimmed mean, al colocar las $k - \alpha N$ muestras cercanas a la mediana $X_{(k+1)}$, se selecciona dos conjuntos de k muestras por ambos lados de la mediana. La salida se obtiene por el promedio de estas muestras y la mediana. En otras palabras, la salida es obtenida por el promedio de las muestras en la ventana, en el rango $[X_{(k+1)} - q_1, X_{(k+1)} + q_2]$, donde q_1 y q_2 dependen del dato. El filtro modified trimmed mean, se basa en la idea de cambiar q_1 y q_2 para ser independientes del dato. En este filtro, se estima el valor q y se fija, entonces todas las muestras en el rango $[X_{(k+1)} - q, X_{(k+1)} + q]$ son promediadas, y el promedio es la salida. Por lo tanto la idea del filtro modified trimmed mean es excluir todas las muestras de la ventana que se encuentran muy lejos del valor de la mediana. El parámetro q es usado para clasificar las muestras en dos grupos, los datos demasiado lejos de la mediana y los datos cercanos a la mediana [19-21].

$$MTM = \frac{\sum_{i=1}^N a_i X_i}{\sum_{i=1}^N a_i} \quad (4.3)$$

$$\text{donde } a_i = \begin{cases} 1, & \text{si } |X_i - X_{(k+1)}| \leq q \\ 0, & \text{otro caso} \end{cases}$$

El filtro modified trimmed mean es eficiente para suprimir el ruido Gaussiano. En el caso de ruido impulsivo, suaviza la imagen borrando los contornos y líneas (frecuencias altas) y no suprime el ruido.

IV.5. Filtro de Orden de Rango

El filtro de orden de rango es una modificación del filtro de mediana. La salida del filtro del r th del orden de rango se obtiene [19, 20]:

$$RO(X_1, X_2, \dots, X_n; r) = X_{(r)} \quad . \quad (4.4)$$

El filtro de orden de rango puede ser usado en situaciones donde la distribución del ruido no es simétrica, por ejemplo cuando hay más impulsos positivos que negativos. En este caso el uso de rangos bajos, pueden hacer un filtro más robusto que el filtro de mediana [19, 20].

IV.6. Filtro Mediano de Multietapa

Estos filtros se diseñaron por el problema que existe al calcular la mediana en ventanas muy grandes. Muchas veces en el caso de procesamiento de imágenes, cuando la ventana de filtrado es mayor a 9×9 . El filtro de mediano de multietapa usa varias etapas de filtros medianos, esto es, se calcula la mediana de medianas. Otra razón para usar el filtro mediano de multietapa es que toma en cuenta la información de orden temporal o espacial mucho mejor, en comparación con el filtro mediano. La tercera razón, relacionada con la segunda y primera, es que tiene mayor flexibilidad que el filtro mediano [19, 20, 22].

El filtro mediano separable (separable two-dimensional median filter) es una modificación del filtro mediano, el cual conserva las propiedades de los filtros medianos, pero tiene la ventaja de que su algoritmo es mucho más rápido. Este filtro tiene dos etapas, en la primera etapa se calculan las medianas en dirección horizontal y en la segunda etapa en dirección vertical. Esta idea puede ser fácilmente generalizada en varias etapas y diferentes subventanas de filtrado. Las subventanas de filtrado para los filtros medianos multietapa se muestran en la figura 4.1. Entonces los filtros medianos multietapa son calculados de la siguiente forma [19, 20, 22]:

$$MSM_1 = MED\{X_{m+1,m+1}, h - med, v - med\} \quad , \quad (4.5a)$$

$$MSM_2 = MED\{X_{m+1,m+1}, d45 - med, d135 - med\} \quad , \quad (4.5b)$$

$$MSM_3 = MED\{h - med, v - med, d45 - med, d135 - med\} \quad , \quad (4.5c)$$

$$MSM_4 = MED\{X_{m+1,m+1}, h - med, v - med, d45 - med, d135 - med\} \quad , \quad (4.5d)$$

$$MSM_5 = MED\left\{ \begin{array}{l} X_{m+1,m+1}, MED\{X_{m+1,m+1}, h - med, v - med\}, \\ MED\{X_{m+1,m+1}, d45 - med, d135 - med\} \end{array} \right\} \quad , \quad (4.5e)$$

$$MSM_6 = MED\{X_{m+1,m+1}, c - med, x - med\} \quad , \quad (4.5f)$$

donde el subíndice m es encontrado de $z = 2m + 1$ y $z \times z = n$.

Finalmente, se calcula la mediana de las medianas de las ecuaciones 4.5. Por otro lado, los filtros medianos multietapa tienen relación con los filtros máximo/mediano (max/median filters), en donde la salida esta definida como [19, 20, 22, 23]:

$$MaxMed = MAX\{h - med, v - med, d45 - med, d135 - med\} \quad , \quad (4.6)$$

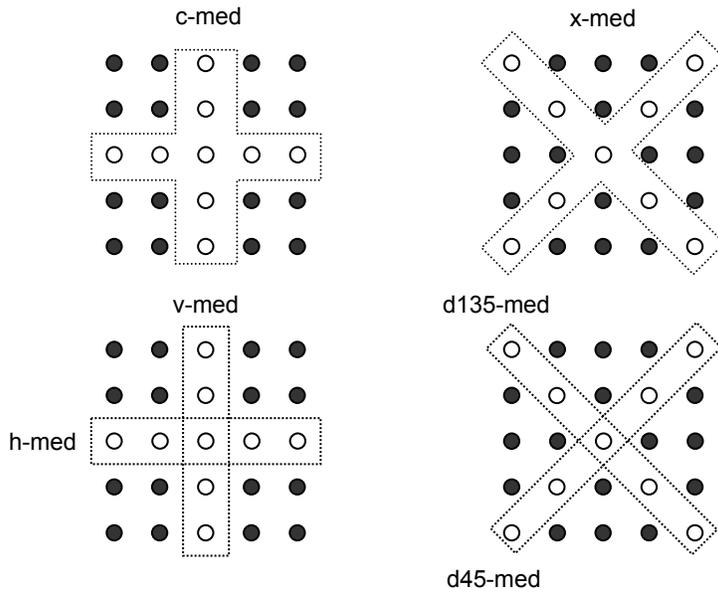


Figura 4.1. Subventanas para el filtro mediano multietapa.

IV.7. Filtro de Comparación y Selección

Este filtro se basa en la observación de que en la salida del filtro de mediana es pequeña que la salida del filtro de promedio después del resalte de los bordes; y la salida del filtro de mediana es mayor que la salida del filtro de promedio antes del resalte de los bordes. Así, el método para mejorar los bordes borrosos [19, 20] es:

$$CS = \begin{cases} X_{(k+1-j)} & \text{MEAN}\{X_1, X_2, \dots, X_n\} \geq \text{MED}\{X_1, X_2, \dots, X_n\} \\ X_{(k+1+j)} & \text{cualquier otro caso} \end{cases} \quad (4.7)$$

(4.7)

donde $1 \leq j \leq k$, $n = 2k + 1$ y el valor de j es propuesto.

El filtro de comparación y selección, es un filtro para resaltar los bordes borrosos, a expensa de suprimir el ruido [19, 20].

IV.8. Filtro de Selección de Promedio y Filtro de Selección de Mediana

La idea en estos dos filtros, es utilizar dos predictores; un predictor hacia atrás y un predictor hacia adelante. Los predictores son la media aritmética de k muestras previas y la media aritmética de las k muestras siguientes, para el filtro de selección de promedio [19, 20].

$$\text{SelAve}(X_1, X_2, \dots, X_N; j) = \begin{cases} \text{Media}_b, & \text{si } |\text{media}_b - X_{k+1}| \leq |\text{media}_f - X_{k+1}| \\ \text{Media}_f, & \text{Otro caso} \end{cases} \quad (4.8)$$

$$\text{SelMed}(X_1, X_2, \dots, X_N; j) = \begin{cases} \text{Mediana}_b, & \text{si } |\text{mediana}_b - X_{k+1}| \leq |\text{mediana}_f - X_{k+1}| \\ \text{Mediana}_f, & \text{Otro caso} \end{cases} \quad (4.9)$$

Mediana_f, Otro caso

donde $media_b = \text{MEAN}\{X_1, X_2, \dots, X_k\}$, $media_f = \text{MEAN}\{X_{k+2}, X_{k+3}, \dots, X_N\}$, $mediana_b = \text{MED}\{X_1, X_2, \dots, X_k\}$, $mediana_f = \text{MED}\{X_{k+2}, X_{k+3}, \dots, X_N\}$.

Este filtro puede simultáneamente mejorar los bordes borrosos y suprimir el ruido eficientemente. En computo, es más eficiente que el filtro Hachimura – Kuwahara [19, 20].

IV.9. Filtros de Selección de Rango

Los filtro de selección de rango en general cae en la estructura de los filtros de selección. El diagrama a bloques de la figura 4.2, muestra la estructura de los filtros de selección de rango. Este consiste en tres bloques diferentes. Uno de los bloques extrae alguna información de las muestras $\mathbf{x} = (X_{(1)}, X_{(2)}, \dots, X_{(n)})$. Esta información es guardada en el vector \mathbf{z} . El segundo bloque produce el vector ordenado \mathbf{x}_0 . Finalmente el tercer bloque representa el selector de rangos, el cual escoge el rango más apropiado de las muestras del vector $\mathbf{x}_0 = (X_{(1)}, X_{(2)}, \dots, X_{(n)})$ para ser la salida del filtro. Esta selección esta basada en las características del vector \mathbf{z} y en la regla de selección de rango $\mathbf{S}(\cdot)$ [19, 20, 24].

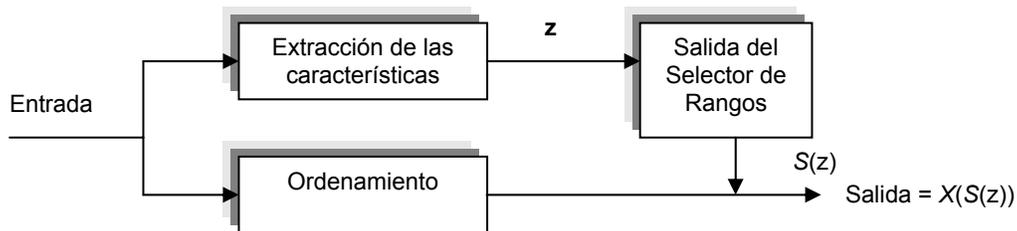


Figura 4.2. Estructura de los filtros de selección de rango.

IV.9.1. Filtro “Lower Upper Middle Smoother”

$$\text{LUMSmooth}(X_1, X_2, \dots, X_N; s) = \text{MED}\{X_{(s)}, X^*, X_{(N-s+1)}\} \quad (4.10)$$

El valor del parámetro s activa el tono entre la preservación de detalles y la supresión de ruido. Si $s = 1$ el filtro LUM smoother es un filtro de identidad; si $s = k + 1$, es el filtro de mediana [19, 20, 24].

IV.9.2. Filtro “Lower Upper Middle Sharpener”

El filtro LUM Sharp mueve las muestras lejos de la mediana, a los extremos del orden estadístico para agudizar y resaltar los bordes [19, 20, 24].

$$\text{LUMSharp}(X_1, X_2, \dots, X_N; t) = \begin{cases} X_{(t)}, & \text{Si } X_{(t)} < X^* \leq t_l \\ X_{(N-t+1)}, & \text{Si } t_l < X^* \leq X_{(N-t+1)} \\ X^*, & \text{Otro caso} \end{cases} \quad (4.11)$$

donde el punto medio del orden estadístico entre $X_{(t)}$ (lower) y $X_{(N-t+1)}$ (upper) es $t_L = \frac{X_{(t)} + X_{(N-t+1)}}{2}$, y t es el que controla la agudeza de los contornos [19, 20, 24].

IV.9.3. Filtros “Lower Upper Middle”

Cuando la filosofías del filtro LUM Smoother y LUM Sharpener son combinadas, se obtiene el filtro LUM. La salida del filtro LUM con los parámetros s y t , $1 \leq s \leq t \leq k + 1$, se obtiene por [19, 20]:

$$\text{LUM}(X_1, X_2, \dots, X_N; s, t) = \begin{cases} X_{(s)} & \text{si } X^* < X_{(s)} \\ X_{(t)} & \text{si } X_{(t)} < X^* \leq t_L \\ X_{(n-t+1)} & \text{si } t_L < X^* \leq X_{(n-t+1)} \\ X_{(n-s+1)} & \text{si } X_{(n-s+1)} < X^* \\ X^* & \text{cualquier otro caso} \end{cases}, \quad (4.12)$$

$$\text{donde } t_L = \frac{X_{(t)} + X_{(n-t+1)}}{2}.$$

IV.10. Filtro Wilcoxon

Los filtros R son basados en los estimadores R. Los estimadores R son conocidos por ser estimadores robustos. El primer filtro R introducido es el filtro Wilcoxon [19, 20, 25]:

$$\theta_{\text{Wil}} = \text{MED} \left\{ \frac{1}{2} (X_{(i)} + X_{(j)}), 1 \leq i \leq j \leq n \right\}, \quad (4.13)$$

Así, el filtro Wilcoxon consiste de una operación lineal, llamada promedio de Walsh $W_{(i,j)} = (X_i + X_j)/2$ u orden de promedio de Walsh $W_{(i,j)} = (X_{(i)} + X_{(j)})/2$, y de una operación no lineal, la mediana. El filtro Wilcoxon, de hecho es una versión de los estimadores Hodges-Lehmann, aplicados a los filtros. El filtro Wilcoxon actualmente es un caso del filtro híbrido de mediana. Este filtro es la mejor estimación cuando la información de la distribución de los datos tiene forma simétrica [19, 20, 25].

Al filtro Wilcoxon se le puede aplicar la operación de winsorización o trimming. Esta operación consiste en que los promedios o averages de Walsh deben ser computados solo para las muestras cuyos rangos están colocados a una distancia máxima entre ellos [19, 20, 25]:

$$\theta_{\text{RW-Wil}} = \text{MED} \left\{ \frac{1}{2} (X_{(i)} + X_{(j)}), j-i < r, 1 \leq i \leq j \leq n \right\}, \quad (4.14)$$

mediante la asignación del rango de r varía de 1 hasta n , el filtro (4.14) llamado filtro Wilcoxon tipo rango con winsorización (Rank Winsorized Wilcoxon Filter) varía del filtro mediano al Wilcoxon. Este filtro ofrece un compromiso entre el filtro mediano y el filtro Wilcoxon, por lo que pueden remover el ruido impulsivo completamente [19, 20, 25].

La principal desventaja del filtro Wilcoxon es la complejidad computacional. El filtro Wilcoxon computa la mediana de $N(N+1)/2$ promedios de Walsh [19, 20, 25].

IV.11. Estimadores M

Huber propuso los estimadores M como una generalización de los estimadores de máximo de verosimilitud (MLE) [19, 20, 26, 27]. Los filtros M son simplemente estimadores M de los parámetros locales necesarios en aplicaciones de filtrado. Su definición es dada por una función ρ

$\{\rho(X) = \ln(F(X))\}$ conectada con la función de densidad de probabilidad $F(X)$ de una muestra de datos $X_i, i = 1, \dots, N$ [19, 20, 26, 27]:

$$\hat{\theta} = \operatorname{argmin}_{\theta \in \Theta} \sum_{i=1}^N \rho(X_i - \theta) \quad , \quad (4.15)$$

La estimación del parámetro local θ puede ser encontrada mediante el cálculo de la derivada parcial de ψ con respecto a θ [19, 20, 26, 27]

$$\sum_{i=1}^N \psi(X_i - \theta) = 0 \quad , \quad (4.16)$$

donde θ es un parámetro local.

La solución del estimador robusto M para θ es determinado imponiendo ciertas restricciones en la función de influencia $\psi(X)$ o en las muestras $X_i - \theta$, llamada censorización o "trimming" [19, 20, 26, 27]. La técnica estándar para el estimador M asume el uso del método iterativo de Newton [19, 20, 26, 27]

$$\theta^{(q+1)} = \theta^{(q)} + S^{(q)} \frac{\sum \psi \left[\frac{X_i - \theta^{(q)}}{S^{(q)}} \right]}{\sum \psi' \left[\frac{X_i - \theta^{(q)}}{S^{(q)}} \right]} \quad , \quad (4.17)$$

donde $\theta^{(q)}$ es la estimación- M de la muestra del parámetro θ y $S^{(q)}$ es una estimación de escala, ambas en el paso o iteración q . Usualmente, $\theta^{(0)}$ es la mediana de los datos primarios y $S^{(q)}$ esta definido como [19, 20, 26, 27]

$$S^{(q)} = G^{-1} \left(\frac{3}{4} \right) \operatorname{med} \left\{ X_i - \theta^{(q)} \right\} \approx 1.483 \cdot \operatorname{med} \left\{ X_i - \theta^{(q)} \right\} \quad , \quad (4.18)$$

donde G es la función estándar de la distribución Gaussiana. El denominador de la ecuación 4.17 no es crítico y este puede ser reemplazado por una constante. Si $0 \leq \psi' \leq 1$ cualquier constante en el denominador $> \frac{1}{2}$ deberá de converger [19, 20, 26, 27].

La limitación de la desviación de resultados de $\psi(X)$ es una función $\tilde{\psi}(X)$ de la forma:

$$\tilde{\psi}_b(X) = [\psi(X)]_b^a = \begin{cases} -b, & \text{si } \psi(X) - a < -b \\ \psi(X), & \text{si } -b \leq \psi(X) - a < b \\ b, & \text{si } \psi(X) - a > b \end{cases} \quad , \quad (4.19)$$

donde $a = 0$ para una distribución simétrica de X_i . La forma más simple de la ec. (4.19) sobre el rango de $\psi(X)$ es el límite del estimador M de Huber para una distribución normal teniendo limitaciones [19, 20, 26, 27]

$$\tilde{\varphi}_b(X) = \text{MIN}(b, \text{MAX}(X, -b)) = \text{MIN}\left(1, \frac{b}{|X|}\right) = [X]_{-b}^b = \begin{cases} X & |X| < b \\ b \cdot \text{sgn}(X) & |X| \geq b \end{cases} \quad (4.20)$$

Otra manera para encontrar la función $\tilde{\varphi}(X)$ es la de eliminar los valores lejanos de una muestra de datos. Este método es llamado estimaciones M mínimas. Hampel demostró que la función de corte mediana (skipped median) es la más reducida estimación robusta M [19, 20, 26, 27]:

$$\psi_{\text{med}(r)}(X) = \text{sgn}(X) \cdot 1_{[-r,r]}(X) = \begin{cases} \text{sgn}(X), & |X| \leq r \\ 0, & X > r \end{cases} \quad (4.21)$$

este estimador es el B -robusto (donde B comienza con bias o sesgo). Este estimador es llamado “skipped median” porque sus observaciones desaparecen al llegar al valor de r . La palabra “skipped” se refiere al “Huber-type skipped mean” y no a los “Tukey’s skipping procedures” [19, 20, 26, 27].

Para reducir la complejidad de la función de influencia, se propuso inicialmente el uso de la función simple de corte (simplest cut) para derivar los estimadores robustos M aplicables al procesamiento de imágenes [19, 20, 26, 27]

$$\psi_{\text{cut}(r)}(X) = X \cdot 1_{[-r,r]}(X) = \begin{cases} X, & |X| \leq r \\ 0, & |X| > r \end{cases} \quad (4.22)$$

También existen diferentes funciones de influencia propuestas en la literatura [19, 20, 26, 27]:

la función de Hampel (Hampel’s three part redescending function)

$$\psi_{\alpha,\beta,r}(X) = \begin{cases} X, & 0 \leq |X| \leq \alpha \\ \alpha \cdot \text{sgn}(X), & \alpha \leq |X| \leq \beta \\ \alpha \frac{r - |X|}{r - \beta}, & \beta \leq |X| \leq r \\ 0, & r \leq |X| \end{cases} \quad (4.23)$$

la función del seno de Andrews (Andrews sine function)

$$\psi_{\text{sin}(r)}(X) = \begin{cases} \sin(X/r), & |X| \leq r\pi \\ 0, & |X| > r\pi \end{cases} \quad (4.24)$$

la función de Tukey (Tukey biweight function)

$$\psi_{\text{bi}(r)}(X) = \begin{cases} X^2(r^2 - X^2), & |X| \leq r \\ 0, & |X| > r \end{cases} \quad (4.25)$$

y finalmente la función de Bernoulli (Bernoulli function)

$$\psi_{\text{ber}(r)}(X) = X^2 \sqrt{r^2 - X^2} \cdot 1_{[-r,r]}(X) \quad , \quad (4.26)$$

Las funciones de influencia (4.21) a (4.26) fueron utilizadas para derivar los estimadores robustos M aplicables para procesamiento de imágenes. Estas funciones proveen buena supresión de ruido impulsivo y ruido multiplicativo [19, 20, 26, 27].

IV.12. Criterios Usados para Obtener el Rendimiento del Filtro Propuesto

El criterio usado para comparar el rendimiento de supresión de ruido de diferentes filtros es la relación pico de señal a ruido (PSNR) [19, 20]

$$\text{PSNR} = 10 \cdot \log \left[\frac{(255)^2}{\text{MSE}} \right] \text{ dB} \quad , \quad (4.27)$$

donde $\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x(i, j) - \hat{e}(i, j)]^2$ es el error cuadrático medio, el cual representa una medida objetiva de la desviación del promedio cuadrado para encontrar la estimación del valor verdadero (es la medida objetiva más común para comparar la calidad del filtrado entre la imagen original y la filtrada); $x(i, j)$ es la imagen original; $\hat{e}(i, j)$ es la imagen restaurada o filtrada; y M, N es el tamaño de la imagen [19, 20].

Y para evaluar la preservación de contornos y detalles finos el criterio usado es el error absoluto medio (MAE) [19, 20].

$$\text{MAE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |x(i, j) - \hat{e}(i, j)| \quad (4.28)$$

IV.13. Conclusiones

En este capítulo, se presentaron los filtros Trimmed Mean, que tiene la característica de eliminar ambos extremos de los datos ordenados, para no incluir en el cálculo los datos más lejanos a la mediana y obtener una mejor estimación del píxel. Los filtros Mediano Multietapa, son más rápidos en calcular la mediana en ventanas de procesamiento muy grandes, y tiene mayor flexibilidad que el filtro de mediano. Los filtros de selección y comparación mejoran los bordes borrosos y suprimen el ruido eficientemente. Los filtros LUM preservan las líneas y contornos. El filtro Wilcoxon suprime bien el ruido impulsivo.

Las funciones de influencia presentadas se utilizan para derivar los estimadores robustos M , aplicados para el procesamiento de imágenes. Estas funciones proveen buena supresión de ruido impulsivo y ruido multiplicativo.

Las características de estos filtros son importantes para el diseño de nuevos filtros, y tener un

equilibrio entre la supresión de ruido y el tiempo de procesamiento.

Los filtros presentados en este capítulo, se modificaron para implementarse en la supresión de ruido, restauración de los detalles finos y preservar los contornos en imágenes médicas 3D, que se describen en el capítulo V.

CAPITULO V. IMPLEMENTACIÓN DE LOS FILTROS DE ESTADÍSTICA DE ORDEN EN IMÁGENES MÉDICAS 3D

V.1. Introducción

Las manchas en las imágenes del ultrasonido resultan directamente del uso del transductor coherente y ocurre cuando la estructura en un objeto es de una escala demasiado pequeña para ser resuelta por el sistema de imágenes (Burckhardt, 1978) [28]. Las manchas son una propiedad indeseada ya que enmascaran pequeñas características significantes para el diagnóstico [28-30].

En este capítulo se presentan los resultados obtenidos al implementar filtros 3D para la supresión de ruido Gaussiano e impulsivo en imágenes 3D capturadas con el sistema experimental descrito en el capítulo III.

Para poder suprimir el ruido en las imágenes médicas 3D, se modificaron los filtros de estadística de orden implementados en imágenes de 2D a filtros 3D para su implementación en imágenes médicas de 3D. De los filtros implementados se eligieron los más óptimos para restaurar las imágenes 3D originales, ya que esta contiene manchas debido al transductor coherente.

También se incluyen los resultados del desempeño de la ventana y diferentes formas de cubos de barrido para el filtrado 3D.

V.2. Ventana y Cubo

Para la implementación de los filtros de estadística de orden en imágenes de dos dimensiones se implementa una ventana de $n \times n$, la cual recorre toda la imagen pixel por pixel, figura 5.1. Para implementar los filtros de estadística de orden en imágenes de tres dimensiones, se implementa un cubo de $n \times n \times n = n^3$ voxels, el cual recorre toda la imagen de tres dimensiones voxel por voxel, figura 5.1.

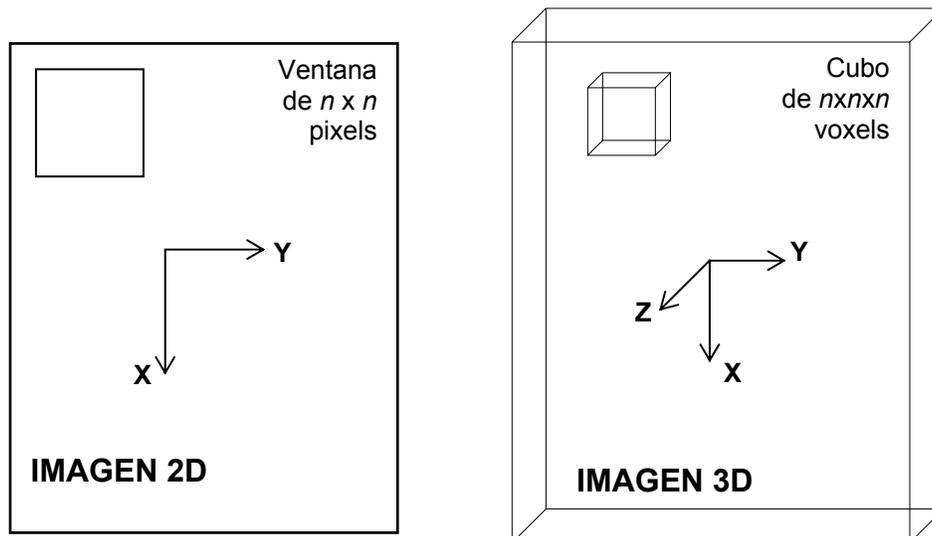


Figura 5.1. Bosquejo del barrido en 2D y 3D.

V.3. Criterios usados para obtener el rendimiento de los filtros 3D

El criterio usado para comparar el rendimiento de supresión de ruido de los diferentes filtros 3D, fue la relación pico de señal a ruido (PSNR) [20-35]:

$$\text{PSNR} = 10 \cdot \log \left[\frac{(255)^2}{\text{MSE}} \right], \text{dB} \quad , \quad (5.1)$$

y para evaluar la preservación de contornos y detalles finos el criterio usado fue el error absoluto medio (MAE) [20-35]:

$$\text{MAE} = \frac{1}{LMN} \sum_{h=0}^{L-1} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |x(h, i, j) - \hat{e}(h, i, j)| \quad , \quad (5.2)$$

donde $\text{MSE} = \frac{1}{LMN} \sum_{h=0}^{L-1} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x(h, i, j) - \hat{e}(h, i, j)]^2$ es el error cuadrático medio, el cual representa una medida objetiva de la desviación del promedio cuadrado para encontrar la estimación del valor verdadero (es la medida objetiva más común para comparar la calidad del filtrado entre la imagen original y la filtrada); $x(h, i, j)$ es la imagen original; $\hat{e}(h, i, j)$ es la imagen restaurada o filtrada; y L, M, N es el tamaño de la imagen [20-35].

V.4. Filtros de Estadística de Orden en 3D

Se seleccionaron varios filtros de estadística de orden implementados en imágenes 2D, en base a investigaciones y pruebas realizadas [20-35], y se adaptaron en forma 3D implementando un cubo de barrido de 3x3x3, figura 5.2. A continuación se muestran los filtros en su forma 3D.



Figura 5.2. Cubo de barrido 3x3x3.

V.4.1. Filtro α - Trimmed Mean

$$\text{TrMean}(X_1, X_2, \dots, X_N; \alpha) = \frac{1}{N - 2\alpha N} \sum_{i=\alpha N+1}^{N-\alpha N} X_{(i)} \quad , \quad (5.3)$$

donde $\alpha = j / N = 8 / 27$, donde $0 \leq j \leq N / 2$, j es un entero.

V.4.2. Filtro Modified Trimmed Mean

$$MTM (X_1, X_2, \dots, X_N; q) = \frac{\sum_{i=1}^N a_i X_i}{\sum_{i=1}^N a_i}, \quad (5.4)$$

donde: $a_i = \begin{cases} 1, & \text{si } |X_i - X_{(k+1)}| \leq q \\ 0, & \text{otro caso.} \end{cases}$

V.4.3. Filtro Ranked – Order

$$RO(X_1, X_2, \dots, X_N; r) = X_{(r)} \quad (5.5)$$

donde: $r = 14$.

V.4.4. Filtros Multistage Median

$$MSM1 = MED\{X_{i+1,l+1,l+1}, h - med, v - med\} \quad (5.6a)$$

$$MSM2 = MED\{X_{i+1,l+1,l+1}, d45 - med, d135 - med\} \quad (5.6b)$$

$$MSM3 = MED\{h - med, v - med, d45 - med, d135 - med\} \quad (5.6c)$$

$$MSM4 = MED\{X_{i+1,l+1,l+1}, h - med, v - med, d45 - med, d135 - med\} \quad (5.6d)$$

$$MSM5 = MED\{X_{i+1,l+1,l+1}, MSM1, MSM2\} \quad (5.6e)$$

$$MSM6 = MED\{X_{i+1,l+1,l+1}, c - med, x - med\} \quad (5.6f)$$

$$MaxMed = MAX\{h - med, v - med, d45 - med, d135 - med\} \quad (5.6g)$$

donde: $z = (2 \times l) + 1, z \times z \times z = N = 27, l = 1$.

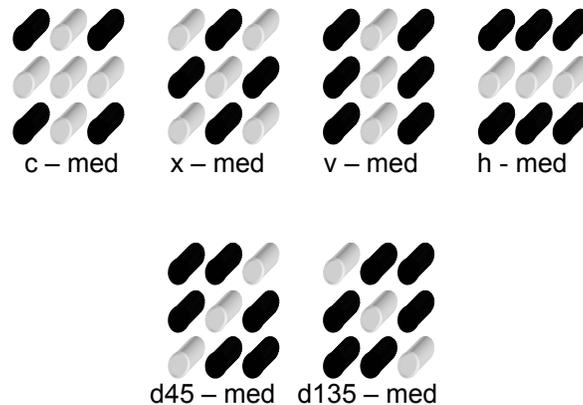


Figura 5.3. Subcubos de 3x3x3 voxels, para los filtros no lineales Multistage Median.

En los subcubos de barrido de la figura 5.3, los paquetes de tres datos (cilindros de color blanco), son los que se implementan en el momento de hallar la mediana. Los diferentes filtros Multistage Median combinan estos subcubos de barrido.

V.4.5. Filtro Comparison y Selection

$$CS(X_1, X_2, \dots, X_N; j) = \begin{cases} X_{(k+1-j)}, & \text{MEAN}\{X_1, X_2, \dots, X_N\} \geq \text{MED}\{X_1, X_2, \dots, X_N\} \\ X_{(k+1+j)}, & \text{Otro caso.} \end{cases}, \quad (5.7)$$

donde: $j = 8$.

V.4.6. Filtro Selective Average

$$\text{SelAve}(X_1, X_2, \dots, X_N) = \begin{cases} \text{Media}_b, & \text{si } |\text{media}_b - X_{k+1}| \leq |\text{media}_f - X_{k+1}| \\ \text{Media}_f, & \text{Otro caso.} \end{cases}, \quad (5.8)$$

donde $\text{Media}_b = \text{MEAN}\{X_1, X_2, \dots, X_k\}$, $\text{Media}_f = \text{MEAN}\{X_{k+2}, X_{k+3}, \dots, X_N\}$.

V.4.7. Filtro Selective Median

$$\text{SelMed}(X_1, X_2, \dots, X_N) = \begin{cases} \text{Mediana}_b, & \text{si } |\text{mediana}_b - X_{k+1}| \leq |\text{mediana}_f - X_{k+1}| \\ \text{Mediana}_f, & \text{Otro caso} \end{cases}, \quad (5.9)$$

donde $\text{Mediana}_b = \text{MED}\{X_1, X_2, \dots, X_k\}$, $\text{Mediana}_f = \text{MED}\{X_{k+2}, X_{k+3}, \dots, X_N\}$.

V.4.8. Filtro LUM Smooth

$$\text{LUMSmooth}(X_1, X_2, \dots, X_N; s) = \text{MED}\{X_{(s)}, X^*, X_{(N-s+1)}\}, \quad (5.10)$$

donde: $s = 7$.

V.4.9. Filtro LUM Sharpener

$$\text{LUMSharp}(X_1, X_2, \dots, X_N; t) = \begin{cases} X_{(t)}, & \text{Si } X_{(t)} < X^* \leq t_i \\ X_{(N-t+1)}, & \text{Si } t_i < X^* \leq X_{(N-t+1)} \\ X^*, & \text{Otro caso} \end{cases}, \quad (5.11)$$

donde: $t = 7$ y $t_i = (X_{(t)} + X_{(N-t+1)}) / 2$.

V.4.10. Filtro LUM

$$\text{LUM}(X_1, X_2, \dots, X_N; s, t) = \begin{cases} X_{(s)}, & \text{Si } X^* < X_{(s)} \\ X_{(t)}, & \text{Si } X_{(t)} < X^* \leq t_i \\ X_{(N-t+1)}, & \text{Si } t_i < X^* \leq X_{(N-t+1)} \\ X_{(N-s+1)}, & \text{Si } X_{(N-s+1)} < X^* \\ X^*, & \text{Otro caso} \end{cases}, \quad (5.12)$$

donde: $s = 4$, $t = 7$ y $t_i = (X_{(t)} + X_{(N-t+1)}) / 2$.

V.4.11. Filtro WILCOXON

$$WIL(X_1, X_2, \dots, X_N) = \text{MED} \left\{ \frac{1}{2} (X_i + X_j), 1 \leq i \leq j \leq N \right\}, \quad (5.13)$$

V.4.12. Filtro MM-KNN

El filtro MM-KNN 3D es determinado de la siguiente manera [31, 32]:

$$\hat{e}_{\text{MMKNN}}^{(q)}(i, j, k) = \text{med} \left\{ g^{(q)}(i+l, j+m, k+n) \right\}, \quad (5.14)$$

donde $g^{(q)}(i+l, j+m, k+n)$ es un conjunto de K_{close} voxels con peso de acuerdo a la función $\psi(X)$ usada y encerrados por el valor del cubo del filtro con la estimación del paso previo $\hat{e}_{\text{MMKNN}}^{(q-1)}(i, j, k)$; la estimación inicial es $\hat{e}_{\text{MMKNN}}^{(0)}(i, j, k) = x(i, j, k)$; $x(i, j, k)$ es la imagen 3D original degradada por ruido impulsivo o es el voxel central de la ventana de filtrado; $(2L+1)^2$ es el tamaño del cubo del filtro con $l, m, n = -L, \dots, L$; $\hat{e}_{\text{MMKNN}}^{(q)}$ denota la estimación en la iteración q ; y q es el índice de la iteración actual. Las iteraciones finalizan cuando el estimador actual $\hat{e}_{\text{MMKNN}}^{(q)}$ es igual al estimador previo $\hat{e}_{\text{MMKNN}}^{(q-1)}(i, j, k)$.

$K_{\text{close}}(i, j, k)$ es el número actual de voxels vecinos cercanos [31, 32]:

$$K_{\text{close}}(i, j, k) = \left[K_{\min} + a \cdot D_s(x(i, j, k)) \right] \leq K_{\max}, \quad (5.15)$$

donde a controla la sensibilidad del filtro para una buena detección de detalles; K_{\min} es el número mínimo de vecinos para remover el ruido; K_{\max} es el número máximo de vecinos que restringe los contornos y los detalles suaves; $D_s(x(i, j, k))$ es el detector de impulsos y MAD es la mediana de las desviaciones absolutas de la mediana [20-32]:

$$D_s(x(i, j, k)) = \frac{\text{med} \left\{ |x(i, j, k) - x(i+l, j+m, k+n)| \right\}}{\text{MAD} \{x(i, j, k)\}} + 0.5 \frac{\text{MAD} \{x(i, j, k)\}}{\text{med} \{x(i+e, j+f, k+g)\}} \quad (5.16)$$

$$\text{MAD} \{x(i, j, k)\} = \text{med} \left\{ |x(i+l, j+m, k+n) - \text{med} \{x(i+e, j+f, k+g)\}| \right\} \quad (5.17)$$

Las funciones de influencia implementadas son [20-32]:

Simplest Cut

$$\psi_{\text{cut}(r)}(X) = X \cdot 1_{[-r, r]}(X) = \begin{cases} X, & |X| \leq r \\ 0, & |X| > r \end{cases} \quad (5.18)$$

Hampel (Hampel's three part redescending function)

$$\psi_{\alpha,\beta,r}(X) = \begin{cases} X, & 0 \leq |X| \leq \alpha \\ \alpha \cdot \text{sgn}(X), & \alpha \leq |X| \leq \beta \\ \alpha \frac{r - |X|}{r - \beta}, & \beta \leq |X| \leq r \\ 0, & r \leq |X| \end{cases} \quad (5.19)$$

Seno de Andrews (Andrews sine function)

$$\psi_{\sin(r)}(X) = \begin{cases} \sin(X/r), & |X| \leq r\pi \\ 0, & |X| > r\pi \end{cases} \quad (5.20)$$

Tukey (Tukey biweight function)

$$\psi_{\text{bi}(r)}(X) = \begin{cases} X^2(r^2 - X^2), & |X| \leq r \\ 0, & |X| > r \end{cases} \quad (5.21)$$

Bernoulli (Bernoulli function)

$$\psi_{\text{ber}(r)}(X) = X^2 \sqrt{r^2 - X^2} \cdot 1_{[-r,r]}(X) \quad (5.22)$$

V.4.12.1. Filtro WM-KNN

El filtro WM-KNN es una combinación de los estimadores robustos R y M , y se define de la siguiente manera [31, 32]:

$$\hat{f}_{\text{WMKNN}}^{(w)}(i, j, k) = \text{med} \left\{ \frac{\mathbf{h}^{(w)}(i+l, j+m, k+n) + \mathbf{h}^{(w)}(i+l_1, j+m_1, k+n_1)}{2} \right\} \quad (5.23)$$

donde $\mathbf{h}^{(w)}(i+l, j+m, k+n)$ y $\mathbf{h}^{(w)}(i+l_1, j+m_1, k+n_1)$ es un conjunto de K_{close} voxels con peso de acuerdo a la función $\psi(X)$ usada y encerrados por el valor del cubo del filtro con la estimación del paso previo $\hat{f}_{\text{RMKNN}}^{(w-1)}(i, j, k)$; la estimación inicial es $\hat{f}_{\text{RMKNN}}^{(0)}(i, j, k) = x(i, j, k)$ y $\hat{f}_{\text{RMKNN}}^{(w)}(i, j, k)$ denota la estimación de la iteración w . El algoritmo termina cuando $\hat{f}_{\text{RMKNN}}^{(w)}(i, j, k) = \hat{f}_{\text{RMKNN}}^{(w-1)}(i, j, k)$.

El filtro WM-KNN cuenta con el detector de impulsos $D_s(i, j, k)$ y la mediana de las desviaciones absolutas de la mediana MAD, definidas anteriormente en las ecuaciones (5.16) y (5.17).

V.4.12.2. Filtro ABSTM-KNN (Ansari-Bradley-Siegel-Tukey)

El filtro ABSTM-KNN, también es una combinación de los estimadores robustos R y M , y se define de la siguiente manera [31, 32]:

$$\hat{f}_{\text{ABSTMKNN}}^{(w)}(i, j, k) = \text{med} \begin{cases} h^{(w)}(i+l, j+m, k+n) & i \leq \frac{N}{2} \\ \frac{h^{(w)}(i+l, j+m, k+n) + h^{(w)}(i+l, j+m_1, k+n_1)}{2} & i > \frac{N}{2} \end{cases} \quad (5.24)$$

donde $h^{(w)}(i+l, j+m, k+n)$ y $h^{(w)}(i+l_1, j+m_1, k+n_1)$ es un conjunto de K_{close} , de los voxels $\leq N/2$, con peso de acuerdo a la función $\psi(X)$ usada y encerrados por el valor del cubo del filtro con la estimación del paso previo $\hat{f}_{\text{ABSTMKNN}}^{(w-1)}(i, j, k)$ y $h^{(w)}(i+l, j+m, k+n)$ es un conjunto de K_{close} , de los voxels $\geq N/2$, con peso de acuerdo a la función $\psi(X)$ usada y encerrados por el valor del cubo del filtro con la estimación del paso previo $\hat{f}_{\text{ABSTMKNN}}^{(w-1)}(i, j, k)$. Donde la estimación inicial es $\hat{f}_{\text{ABSTMKNN}}^{(0)}(i, j, k) = x(i, j, k)$ y $\hat{f}_{\text{ABSTMKNN}}^{(w)}$ denota la estimación de la iteración w . El algoritmo termina cuando $\hat{f}_{\text{ABSTMKNN}}^{(w)}(i, j, k) = \hat{f}_{\text{ABSTMKNN}}^{(w-1)}(i, j, k)$.

El filtro ABSTM-KNN cuenta con el detector de impulsos $D_s(i, j, k)$ y la mediana de las desviaciones absolutas de la mediana MAD, definidas anteriormente en las ecuaciones (5.16) y (5.17).

En la tabla 5.1 se definen los parámetros para los filtros MM-KNN, WM-KNN, ABSTM-KNN y las Funciones de Influencia. Estos parámetros se obtuvieron al realizar varias pruebas con diferentes imágenes 3D.

Tabla 5.1. Parámetros para los filtros KNN y las Funciones de Influencia.

| FUNCION | a | r | ALFA | BETA | Kmin | Kmax |
|-----------------|----------|----------|-------------|-------------|-------------|-------------|
| CUT | 8 | 255 | | | 5 | 24 |
| HAMPEL | 8 | 256 | 200 | 230 | 5 | 24 |
| SENO | 10 | 255 | | | 5 | 24 |
| TUKEY | 10 | 255 | | | 5 | 24 |
| BERNULLI | 10 | 255 | | | 5 | 24 |

V.5. Resultados de las Simulaciones

Las simulaciones constan de los siguientes puntos:

1. Introducir ruido a la imagen 3D.
2. Implementar el filtrado en 3D para las imágenes degradadas.
3. Caracterizar el comportamiento de los filtros 3D, usando:
 - La imagen 3D error.
 - MAE.
 - PSNR.

La imagen 3D implementada para las simulaciones es de 525x382x12 voxels. Se utilizó esta imagen debido a que fue la primera imagen capturada con el sistema experimental y por el tiempo de procesamiento.

Se realizaron dos grupos de pruebas, utilizando en el primer grupo el ruido Gaussiano y en el segundo grupo el ruido impulsivo.

La imagen 3D de error se obtuvo de la siguiente manera [20-32]:

$$\text{error}(h, i, j) = |x(h, i, j) - \hat{e}(h, i, j)| \quad (5.25)$$

donde $x(h, i, j)$ es la imagen original; $\hat{e}(h, i, j)$ es la imagen restaurada o filtrada.

Los resultados de las simulaciones son los siguientes:

Tabla 5.2. Resultados obtenidos de las pruebas realizadas con ruido Gaussiano.

| Filtros No Lineales | Ruido Gaussiano | | | | | | | | | | | |
|---------------------------|-------------------|---------------|------------------|---------------|-------------------|---------------|-------------------|---------------|-------------------|---------------|-------------------|---------------|
| | $\sigma^2 = 0.05$ | | $\sigma^2 = 0.1$ | | $\sigma^2 = 0.15$ | | $\sigma^2 = 0.20$ | | $\sigma^2 = 0.25$ | | $\sigma^2 = 0.30$ | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| Alfa Trimmed Mean | 14.610 | 22.005 | 17.677 | 20.575 | 19.955 | 19.667 | 21.760 | 18.997 | 23.244 | 18.468 | 24.718 | 17.984 |
| Modified Trimmed Mean | 15.124 | 20.418 | 18.663 | 19.095 | 21.257 | 18.245 | 23.327 | 17.621 | 25.109 | 17.111 | 26.572 | 16.702 |
| Ranked Order | 14.520 | 21.587 | 18.179 | 19.802 | 20.832 | 18.737 | 23.020 | 17.957 | 24.689 | 17.411 | 26.361 | 16.901 |
| MSM1 | 17.624 | 20.568 | 23.684 | 18.061 | 28.104 | 16.592 | 31.521 | 15.589 | 34.624 | 14.774 | 37.316 | 14.117 |
| MSM2 | 17.789 | 20.484 | 23.725 | 18.038 | 28.152 | 16.574 | 31.619 | 15.562 | 34.640 | 14.767 | 37.418 | 14.098 |
| MSM3 | 14.206 | 22.421 | 18.456 | 20.261 | 21.704 | 18.890 | 24.255 | 17.932 | 26.524 | 17.161 | 28.643 | 16.504 |
| MSM4 | 15.401 | 21.697 | 20.351 | 19.348 | 24.100 | 17.911 | 27.076 | 16.906 | 29.793 | 16.087 | 32.107 | 15.440 |
| MSM5 | 20.207 | 19.554 | 27.444 | 16.964 | 32.608 | 15.478 | 36.801 | 14.431 | 40.354 | 13.625 | 43.546 | 12.951 |
| MSM6 | 14.688 | 22.083 | 19.374 | 19.744 | 23.025 | 18.287 | 25.979 | 17.256 | 28.415 | 16.477 | 30.752 | 15.807 |
| MaxMed | 24.206 | 18.562 | 32.913 | 15.919 | 39.757 | 14.288 | 45.492 | 13.119 | 50.686 | 12.192 | 55.294 | 11.444 |
| CS | 32.875 | 15.435 | 39.778 | 13.843 | 43.220 | 13.082 | 45.416 | 12.587 | 47.048 | 12.209 | 48.338 | 11.907 |
| SelAve | 17.647 | 21.182 | 22.814 | 19.192 | 26.865 | 17.875 | 30.038 | 16.955 | 32.856 | 16.213 | 35.181 | 15.644 |
| SelMed | 15.750 | 20.836 | 20.094 | 19.013 | 23.300 | 17.870 | 25.969 | 17.029 | 28.284 | 16.345 | 30.206 | 15.804 |
| LUMSmooth | 25.142 | 17.915 | 33.823 | 15.440 | 39.991 | 14.001 | 44.771 | 13.010 | 48.893 | 12.227 | 52.450 | 11.593 |
| LUMSharp | 30.927 | 15.625 | 36.425 | 14.444 | 39.383 | 13.819 | 41.228 | 13.430 | 42.431 | 13.173 | 43.327 | 12.977 |
| LUM | 31.427 | 15.518 | 36.748 | 14.379 | 39.559 | 13.784 | 41.324 | 13.418 | 42.576 | 13.149 | 43.330 | 12.988 |
| Wilcoxon | 15.191 | 22.614 | 19.247 | 20.712 | 22.731 | 19.333 | 26.409 | 18.077 | 30.860 | 16.713 | 32.400 | 16.382 |
| MMKNN CUT | 15.199 | 21.554 | 20.995 | 18.949 | 25.110 | 17.457 | 27.888 | 16.546 | 30.419 | 15.768 | 32.440 | 15.191 |
| MMKNN HAMPEL | 15.169 | 21.572 | 20.798 | 19.040 | 24.571 | 17.671 | 26.809 | 16.934 | 28.740 | 16.341 | 30.191 | 15.927 |
| MMKNN SINE | 14.614 | 21.399 | 20.226 | 18.640 | 24.310 | 17.101 | 27.474 | 16.048 | 30.552 | 15.145 | 32.900 | 14.521 |
| MMKNN BERNOULLI | 13.309 | 22.658 | 17.819 | 20.075 | 21.460 | 18.253 | 24.419 | 16.884 | 27.233 | 15.837 | 30.008 | 14.896 |
| MMKNN TUKEY | 13.446 | 22.499 | 18.125 | 19.855 | 21.543 | 18.302 | 23.890 | 17.357 | 26.242 | 16.487 | 28.358 | 15.744 |

En la tabla 5.2 se observa que el filtro Alfa Trimmed Mean es el más eficiente para suprimir el ruido Gaussiano, esto se debe al recorte de los datos que se encuentran en los extremos, que no se consideran al momento de realizar el promedio. El filtro Ranked Order suprime bien el ruido Gaussiano con varianzas de 0.20, 0.25 y 0.30.

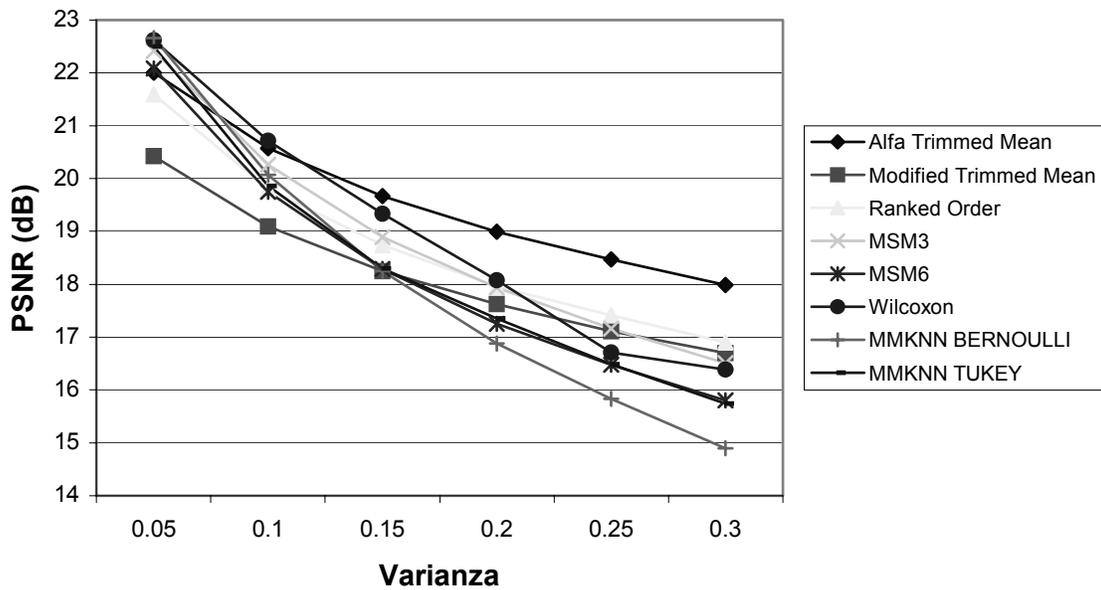


Figura 5.4. Gráfica de PSNR de los mejores filtros 3D.

En la figura 5.4 se observan los mejores filtros para suprimir el ruido Gaussiano. El filtro Alfa Trimmed Mean es el más eficiente para suprimir el ruido Gaussiano y el filtro Ranked Order suprime bien el ruido con varianzas de 0.20, 0.25 y 0.30, como se comentó anteriormente. El filtro Wilcoxon suprime bien el ruido con varianzas menores, pero no preserva los contornos y detalles finos de la imagen. El filtro Modified Trimmed Mean suprime de manera aceptable valores altos de varianza, debido a que únicamente incluye en el promedio los datos más cercanos a la mediana del cubo de barrido.

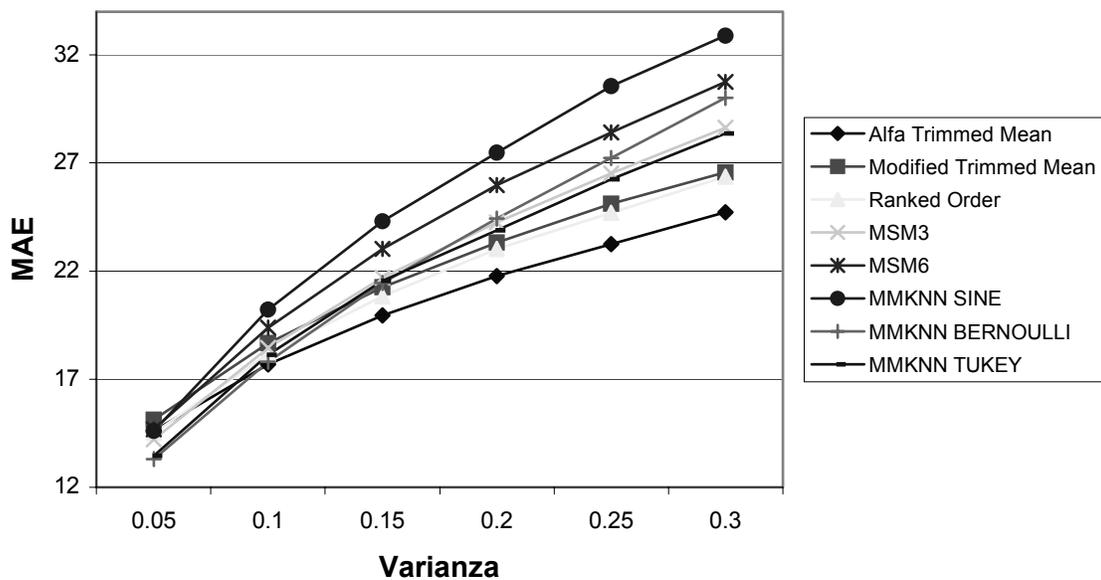


Figura 5.5. Gráfica de MAE de los mejores filtros 3D.

En la figura 5.5 se observa que el filtro Alfa Trimmed Mean preserva mejor los contornos y detalles finos de la imagen. Los filtros Ranked Order y Modified Trimmed Mean obtuvieron buenos resultados para la preservación de contornos y detalles finos.

El siguiente grupo de resultados es muy importante en nuestra investigación, ya que se implementaron los filtros 3D para suprimir diferentes porcentajes de ruido impulsivo.

Tabla 5.3. Resultados obtenidos de las pruebas realizadas con ruido impulsivo.

| Filtros No Lineales | Ruido Impulsivo | | | | | | | | | | | |
|---------------------------|-----------------|---------------|--------------|---------------|--------------|---------------|--------------|---------------|--------------|---------------|--------------|---------------|
| | 5% | | 10% | | 15% | | 20% | | 25% | | 30% | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| Alfa Trimmed Mean | 7.286 | 26.146 | 7.411 | 26.035 | 7.559 | 25.901 | 7.764 | 25.745 | 8.085 | 25.531 | 8.586 | 25.215 |
| Modified Trimmed Mean | 7.049 | 24.903 | 7.104 | 24.912 | 7.206 | 24.832 | 7.348 | 24.715 | 7.535 | 24.522 | 7.795 | 24.275 |
| Ranked Order | 6.672 | 26.502 | 6.745 | 26.449 | 6.834 | 26.376 | 6.942 | 26.279 | 7.084 | 26.136 | 7.268 | 25.946 |
| MSM1 | 4.250 | 28.923 | 4.454 | 28.515 | 4.796 | 27.740 | 5.302 | 26.685 | 6.047 | 25.331 | 7.069 | 23.898 |
| MSM2 | 5.062 | 28.135 | 5.219 | 27.806 | 5.518 | 27.161 | 6.004 | 26.187 | 6.729 | 24.962 | 7.734 | 23.621 |
| MSM3 | 5.948 | 27.461 | 6.051 | 27.357 | 6.210 | 27.186 | 6.461 | 26.880 | 6.821 | 26.435 | 7.355 | 25.784 |
| MSM4 | 5.331 | 27.942 | 5.403 | 27.828 | 5.557 | 27.570 | 5.809 | 27.134 | 6.207 | 26.467 | 6.830 | 25.505 |
| MSM5 | 3.770 | 29.449 | 3.997 | 28.790 | 4.427 | 27.612 | 5.168 | 25.985 | 6.238 | 24.230 | 7.643 | 22.564 |
| MSM6 | 5.069 | 28.299 | 5.129 | 28.205 | 5.254 | 27.993 | 5.445 | 27.675 | 5.770 | 27.099 | 6.261 | 26.269 |
| MaxMed | 6.240 | 27.104 | 6.833 | 26.227 | 7.774 | 24.902 | 9.204 | 23.248 | 11.190 | 21.532 | 13.771 | 19.871 |
| CS | 8.848 | 23.946 | 9.352 | 23.520 | 9.848 | 23.099 | 10.340 | 22.707 | 10.809 | 22.334 | 11.292 | 21.948 |
| SelAve | 6.978 | 26.839 | 9.195 | 25.268 | 11.701 | 23.744 | 14.329 | 22.344 | 16.951 | 21.141 | 19.619 | 20.062 |
| SelMed | 5.636 | 27.434 | 5.871 | 27.028 | 6.109 | 26.675 | 6.369 | 26.314 | 6.682 | 25.906 | 7.086 | 25.389 |
| LUMSmooth | 2.756 | 29.943 | 3.128 | 28.936 | 3.772 | 27.347 | 4.879 | 25.261 | 6.573 | 23.027 | 8.811 | 21.000 |
| LUMSharp | 17.282 | 17.368 | 18.861 | 16.503 | 19.894 | 16.015 | 20.167 | 15.904 | 19.815 | 16.077 | 19.053 | 16.461 |
| LUM | 15.518 | 18.538 | 16.202 | 18.235 | 17.081 | 17.850 | 17.684 | 17.599 | 17.866 | 17.517 | 17.715 | 17.581 |
| Wilcoxon | 7.055 | 26.615 | 7.268 | 26.441 | 7.648 | 26.095 | 8.346 | 25.455 | 9.774 | 24.291 | 11.797 | 22.887 |
| MMKNN CUT | 4.277 | 28.836 | 4.562 | 28.519 | 4.847 | 28.225 | 5.136 | 27.919 | 5.457 | 27.590 | 5.819 | 27.222 |
| MMKNN HAMPEL | 4.314 | 28.790 | 4.596 | 28.480 | 4.878 | 28.193 | 5.164 | 27.890 | 5.484 | 27.567 | 5.843 | 27.212 |
| MMKNN SINE | 4.660 | 28.367 | 4.940 | 28.082 | 5.221 | 27.803 | 5.516 | 27.473 | 5.838 | 27.131 | 6.246 | 26.630 |
| MMKNN BERNOULLI | 4.641 | 28.443 | 4.915 | 28.171 | 5.187 | 27.921 | 5.465 | 27.651 | 5.773 | 27.350 | 6.141 | 26.986 |
| MMKNN TUKEY | 4.643 | 28.433 | 4.919 | 28.158 | 5.192 | 27.904 | 5.474 | 27.619 | 5.785 | 27.311 | 6.161 | 26.906 |
| WMKNN CUT | 10.873 | 22.801 | 11.014 | 22.616 | 11.232 | 22.375 | 11.539 | 22.044 | 11.908 | 21.720 | 12.456 | 21.270 |
| WMKNN HAMPEL | 10.919 | 22.759 | 11.059 | 22.574 | 11.279 | 22.328 | 11.590 | 21.986 | 11.964 | 21.656 | 12.513 | 21.210 |
| ABSTM CUT | 5.160 | 27.525 | 5.426 | 27.288 | 5.693 | 27.060 | 5.957 | 26.823 | 6.241 | 26.596 | 6.568 | 26.313 |
| ABSTM HAMPEL | 5.210 | 27.458 | 5.473 | 27.737 | 5.737 | 27.007 | 5.997 | 26.774 | 6.280 | 26.550 | 6.604 | 26.275 |

En la Tabla 5.3 se observa que el filtro MMKNN con funciones de influencia Cut y Hampel, son los más eficientes para suprimir mayores porcentajes de ruido impulsivo, preservando los contornos y detalles finos de la imagen 3D. El filtro MMKNN con funciones de influencia Seno, Bernoulli y Tukey obtuvieron buenos resultados para suprimir el ruido impulsivo. Los filtros LUM Smooth y MSM5 preservan mejor los contornos y detalles finos de la imagen 3D a menores porcentajes de ruido impulsivo, estos dos filtros no se desempeñan bien para suprimir mayores porcentajes de ruido impulsivo.

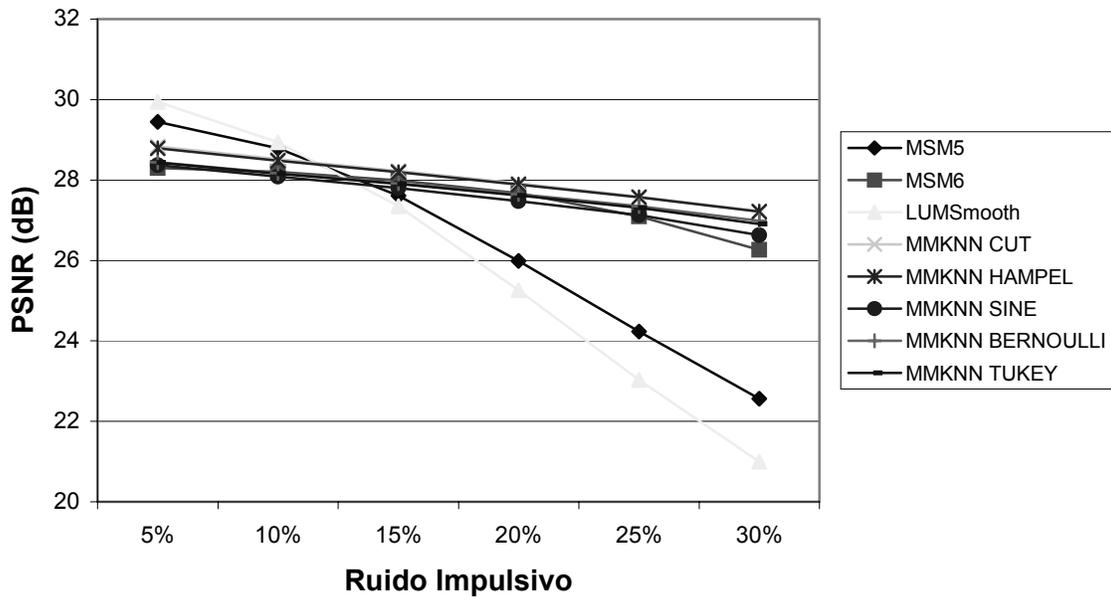


Figura 5.6. Gráfica de PSNR de los mejores filtros 3D.

En la figura 5.6 se graficaron los mejores filtros para suprimir el ruido impulsivo. Como se mencionó anteriormente el filtro MMKNN es el mejor para suprimir mayores porcentajes de ruido, esto se debe a que implementa un detector de impulsos.

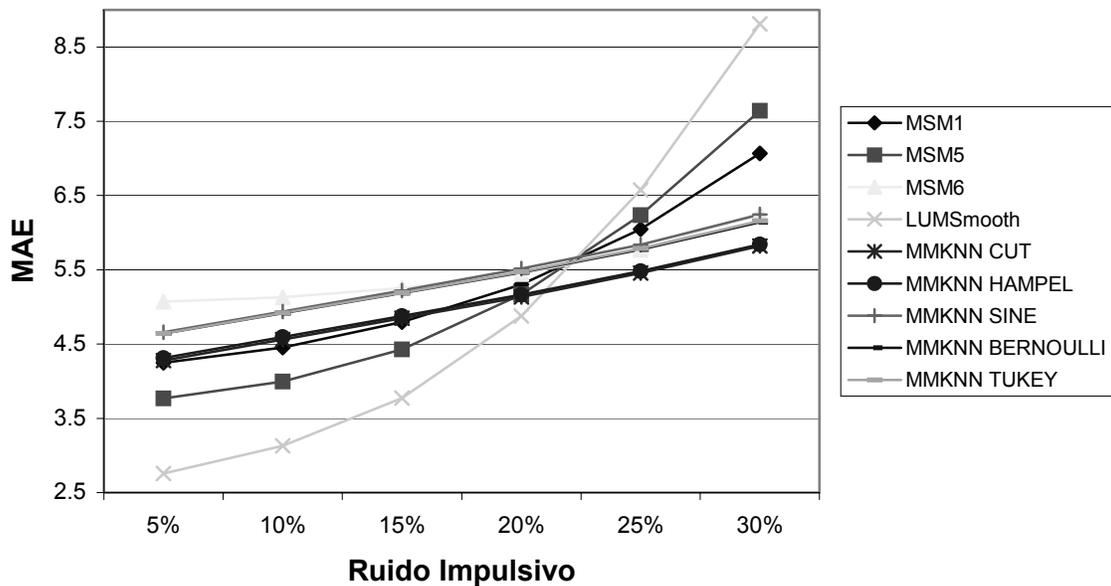


Figura 5.7. Gráfica de MAE de los mejores filtros 3D.

En la figura 5.7 se observan los mejores filtros para preservar los contornos y detalles finos de la imagen 3D. El filtro MSM5 y LUM Smooth preservan de manera excelente los contornos y detalles finos de la imagen 3D, pero no son eficientes para suprimir el ruido impulsivo. El filtro MMKNN preserva los contornos y detalles finos de la imagen 3D con porcentajes grandes de ruido.

La imagen 3D que se implementó para las simulaciones es de 525x382x12 voxels. Es una imagen 3D muy pequeña, en la cual no se aprecian las imágenes de los cortes en los diferentes planos, figura 5.8. Por este motivo se eligió la mejor imagen 3D del segundo grupo de pruebas capturadas con el sistema experimental y se implementaron los mejores filtros para la visualización de los resultados. La imagen 3D implementada es de 640x480x90 y es del músculo del antebrazo derecho. A continuación se muestran los resultados de los filtros implementados en esta imagen 3D.

Tabla 5.4. Resultados obtenidos de las pruebas realizadas con la imagen de 640x480x90.

| Mejores Filtros 3D | Ruido Impulsivo | | | | | |
|--------------------|-----------------|-------|-------|-------|-------|-------|
| | 10% | | 20% | | 30% | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| MMKNN Cut | 3.046 | 29.34 | 3.41 | 28.83 | 3.921 | 28.04 |
| MMKNN Hampel | 3.146 | 29 | 3.504 | 28.55 | 4.006 | 27.83 |
| MSM5 | 1.942 | 32.27 | 3.618 | 26.04 | 8.643 | 20.4 |
| LUM Smooth | 1.542 | 31.85 | 4.873 | 23.56 | 13.93 | 17.51 |

En la tabla 5.4 se obtuvo la misma relación que en la tabla 5.3, con respecto a los mejores filtros. El filtro MMKNN con funciones de influencia Cut y Hampel, son superiores al suprimir mayor porcentaje de ruido impulsivo, y preservar los detalles finos y contornos. Los filtros MSM5 y LUM Smooth se comportan mejor con poco porcentaje de ruido y preservan mejor los detalles finos y los contornos.

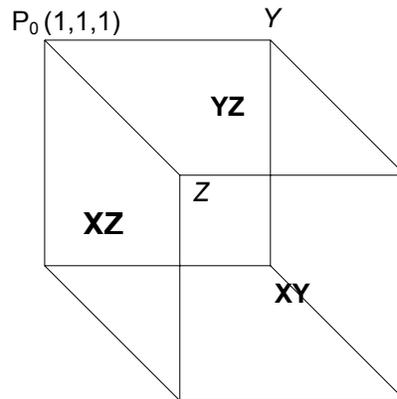


Figura 5.8. Planos donde se realizaron los cortes de la imagen 3D.

A continuación se presentan las imágenes de los cortes en los planos XY, YZ y XZ de los resultados de los filtros MMKNN con funciones de influencia Cut y Hampel, el filtro MSM5 y el filtro LUM Smooth. Las imágenes se dividieron en tres grupos de acuerdo al porcentaje de ruido impulsivo que se utilizó para degradar las imágenes 3D (10%, 20% y 30%). En cada grupo se hace un análisis de las imágenes restauradas y de error, de los cortes en los planos XY, YZ y XZ.

V.5.1. Imágenes Resultantes al Suprimir 30% de Ruido Impulsivo

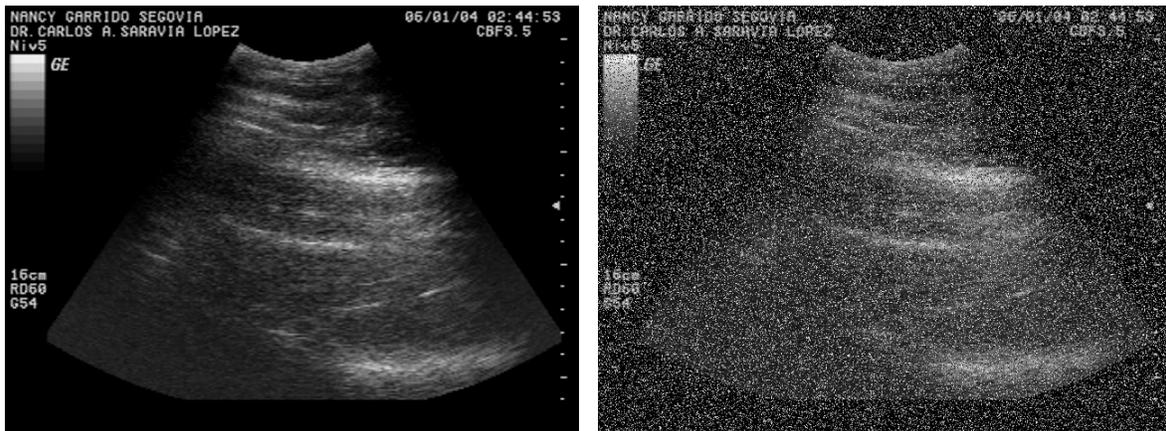


Figura 5.9. Imagen original e imagen degradada con 30% de ruido impulsivo, corte del plano XY.

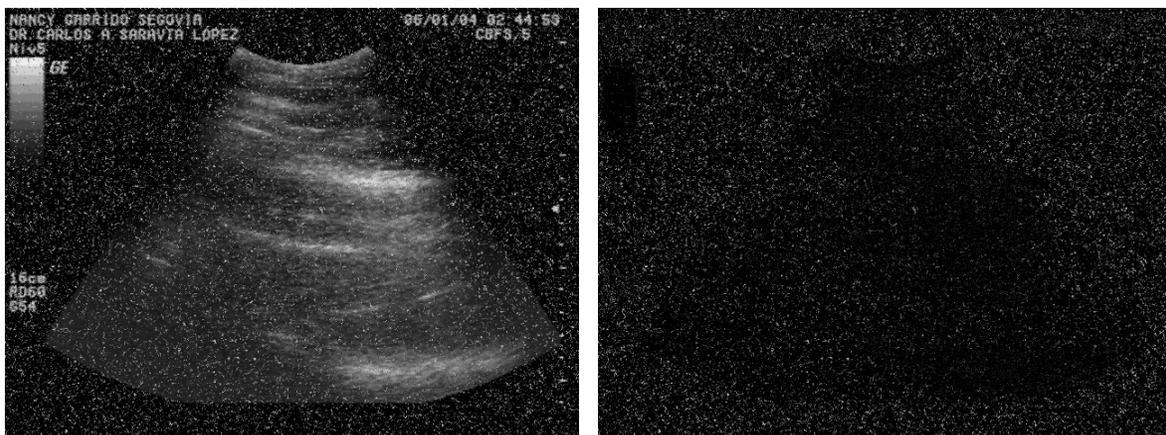


Figura 5.10. Imagen restaurada con el filtro LUM Smooth y su imagen de error, corte del plano XY.

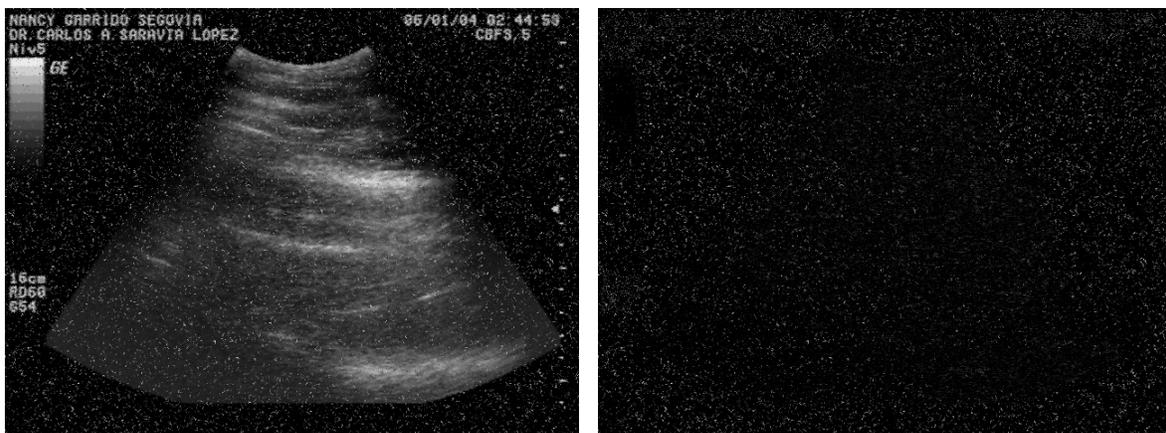


Figura 5.11. Imagen restaurada con el filtro MSM5 y su imagen de error, corte del plano XY.

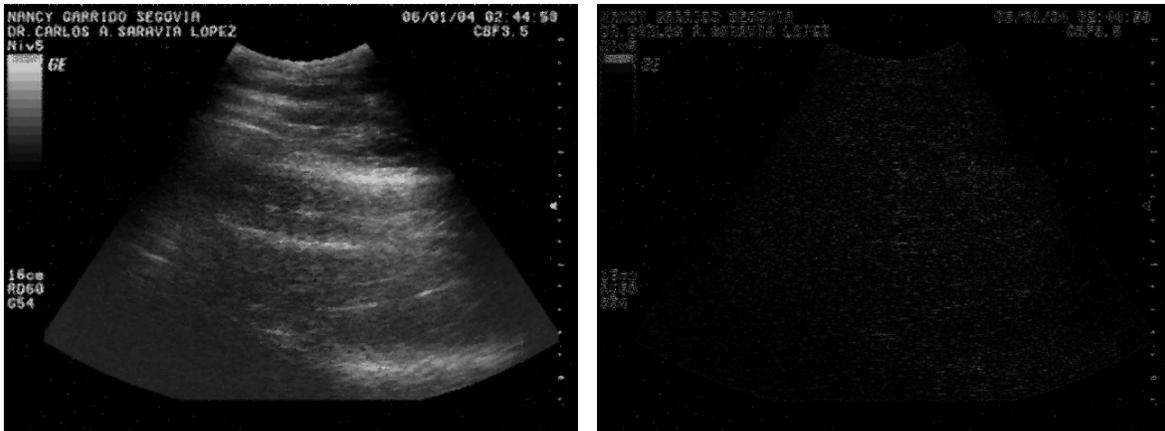


Figura 5.12. Imagen restaurada con el filtro MMKNN con F.I. Hampel y su imagen de error, corte del plano XY.

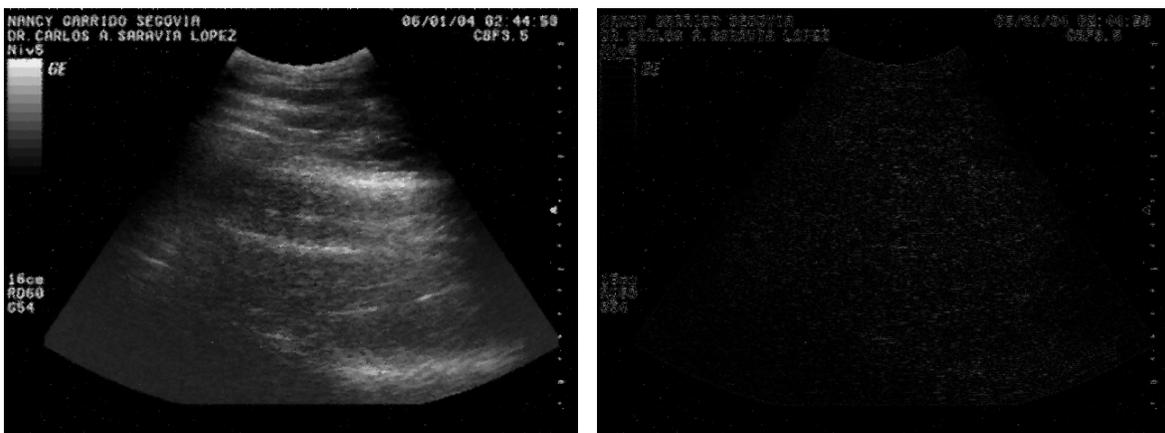


Figura 5.13. Imagen restaurada con el filtro MMKNN con F.I. Cut y su imagen de error, corte del plano XY.

En la figura 5.10 se muestra la imagen restaurada y de error del filtro LUM Smooth. Se observa en la imagen restaurada que el filtro LUM Smooth no suprimió las manchas de ruido, por este motivo no preservó los detalles finos de la imagen, como se aprecia en la imagen de error. En la figura 5.11 se observan los resultados con el filtro MSM5. En la imagen restaurada hay menos manchas de ruido en comparación con el filtro LUM Smooth, y por tal motivo existe un poco de mejora al preservar los detalles finos. En la figura 5.12, la imagen restaurada con el filtro MMKNN con F.I. Hampel, suprimió casi por completo las manchas, y definió los contornos de la imagen. En la imagen de error del filtro MMKNN con F.I. Hampel, se observa que el filtro suprimió los detalles finos de la imagen. La imagen restaurada con el filtro MMKNN con F.I. Cut que se muestra en la figura 5.13, suprimió casi por completo las manchas y también definió los contornos, y en la imagen de error se observa que también suprimió los detalles finos de la imagen original.

Las manchas que no pudieron suprimir los filtros LUM Smooth y MSM5, se debe a que más de la mitad o todos los datos del cubo de barrido sean ruido. Por este motivo los filtros LUM Smooth y MSM5, no logran llegar a una estimación correcta del voxel, preservando las manchas de ruido. El filtro MMKNN con funciones de influencia Cut y Hampel, suprimen casi por completo las manchas, a pesar de que más de la mitad o todos los datos del cubo de barrido sean ruido, esto se debe a la función de influencia y al detector de impulsos implementados en el filtro, entre otras características. Los detalles finos que se observan en las imágenes de error del filtro MMKNN con funciones de influencia Cut y Hampel, son las manchas ocasionadas por el transductor coherente del equipo de ultrasonido [28].

El mismo análisis descrito en el párrafo anterior, se observa en las figuras 5.14 y 5.15, de los planos XZ y YZ, respectivamente. Confirmando el análisis desde otras perspectivas visuales.

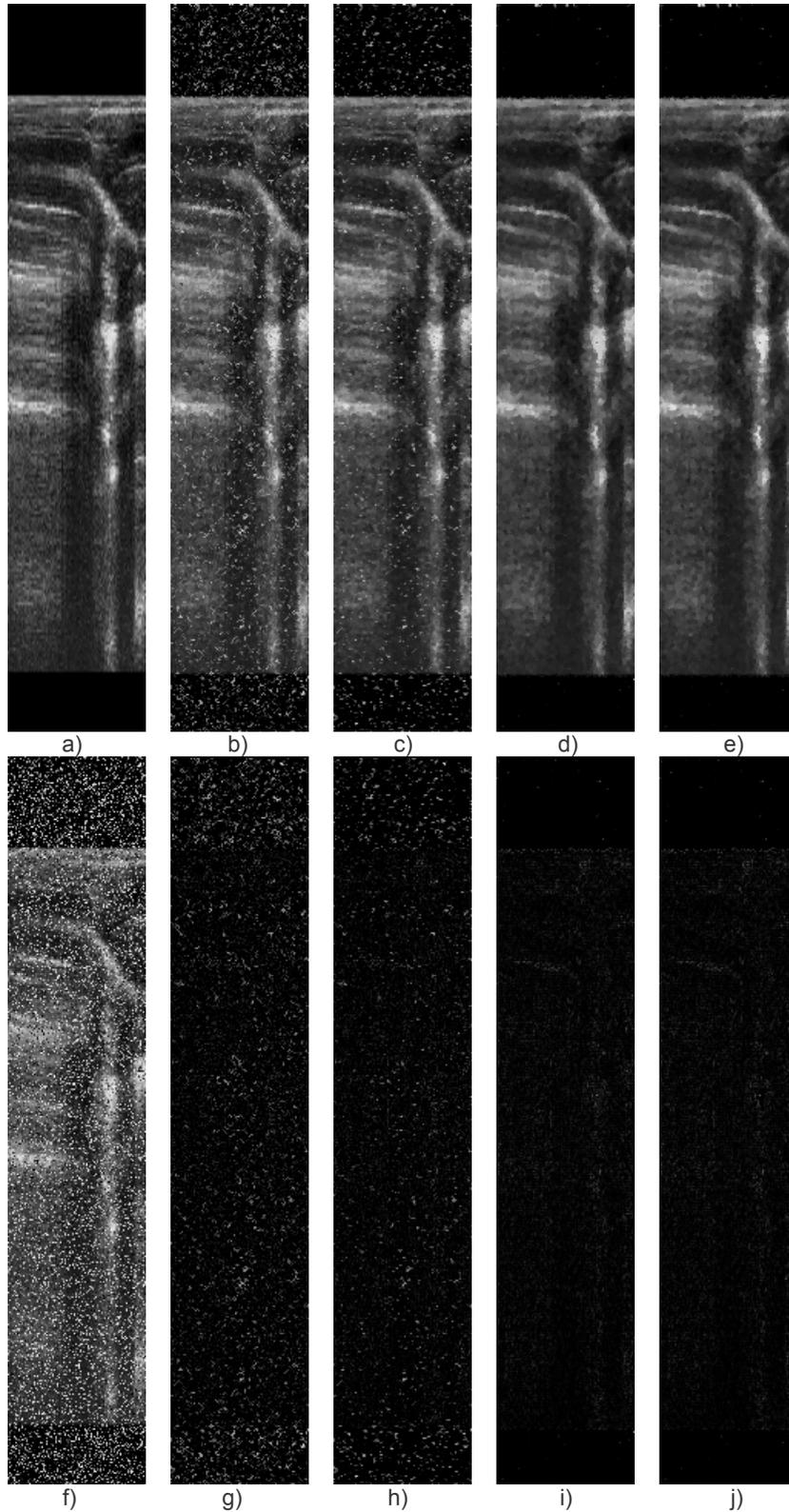


Figura 5.14. a) imagen original; b) imagen restaurada con el filtro LUM Smooth; c) imagen restaurada con el filtro MSM5; d) imagen restaurada con el filtro MMKNN con F.I. Hampel; e) imagen restaurada con el filtro MMKNN con F.I. Cut; f) imagen degradada con 30% de ruido impulsivo; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano XZ.

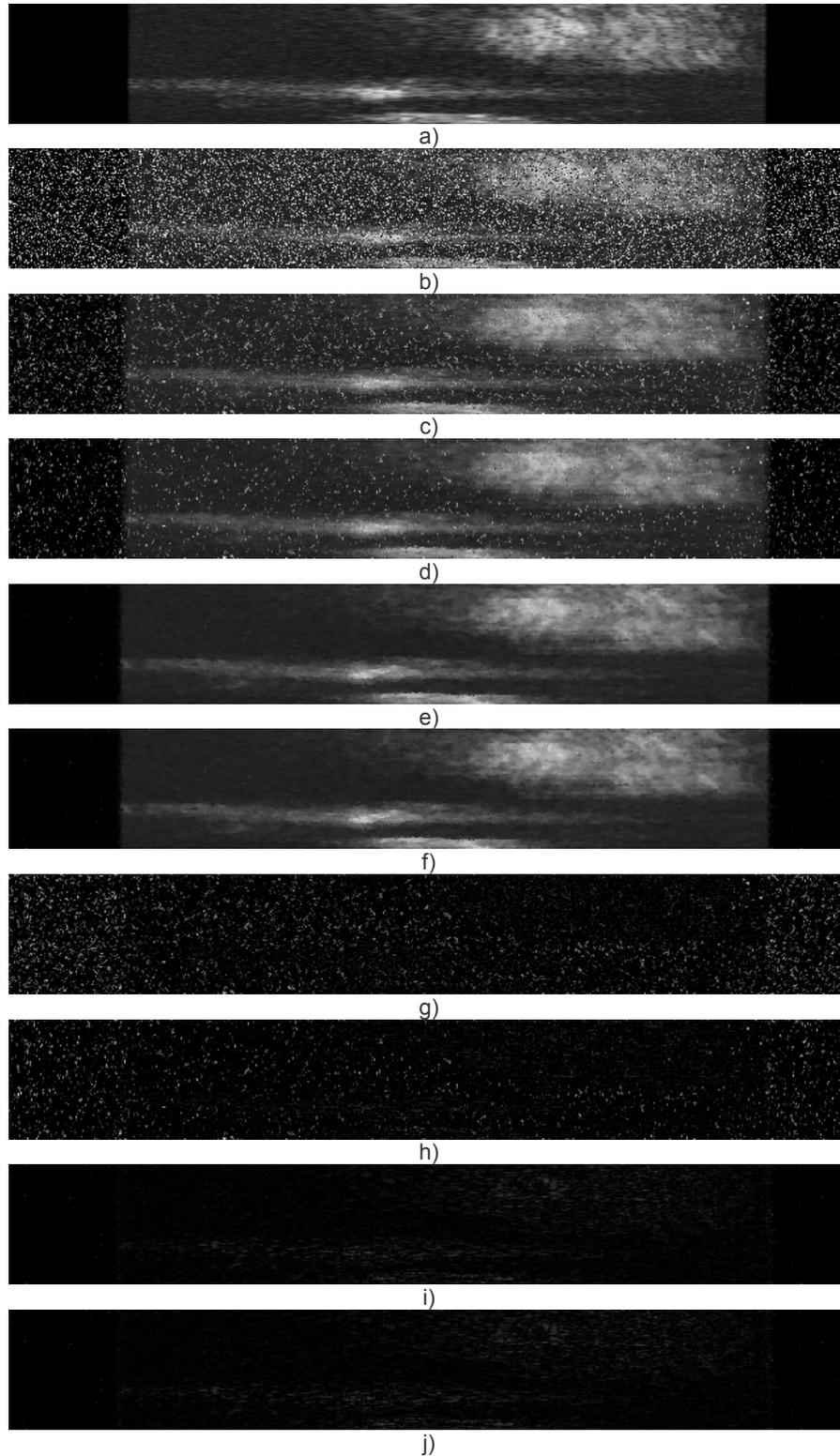


Figura 5.15. a) imagen original; b) imagen degradada con 30% de ruido impulsivo; c) imagen restaurada con el filtro LUM Smooth; d) imagen restaurada con el filtro MSM5; e) imagen restaurada con el filtro MMKNN con F.I. Hampel; f) imagen restaurada con el filtro MMKNN con F.I. Cut; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano YZ.

V.5.2. Imágenes Resultantes al Suprimir 20% de Ruido Impulsivo

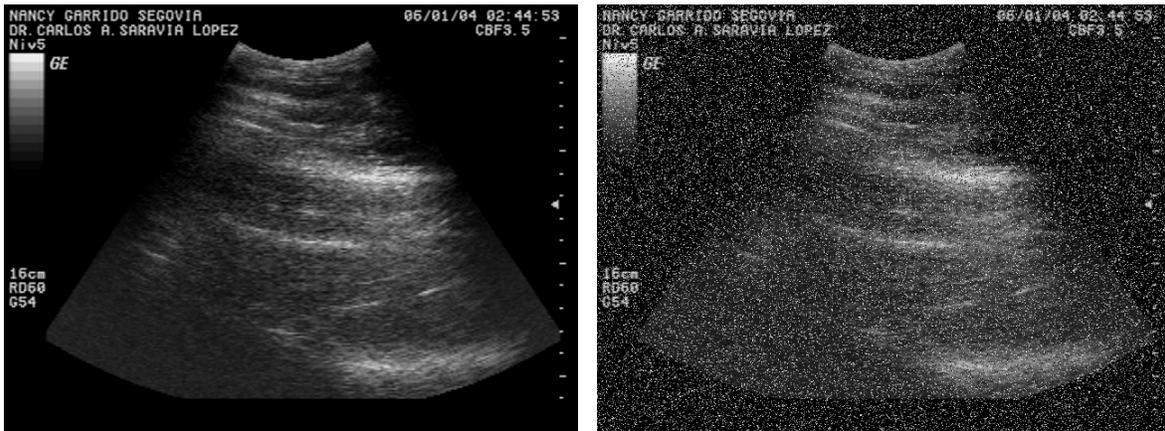


Figura 5.16. Imagen original e imagen degradada con 20% de ruido impulsivo, corte del plano XY.

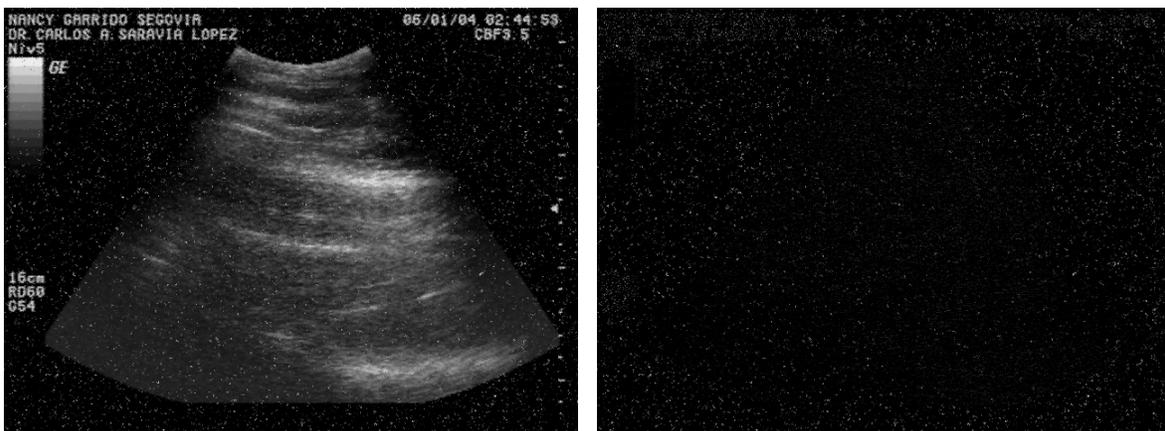


Figura 5.17. Imagen restaurada con el filtro LUM Smooth y su imagen de error, corte del plano XY.

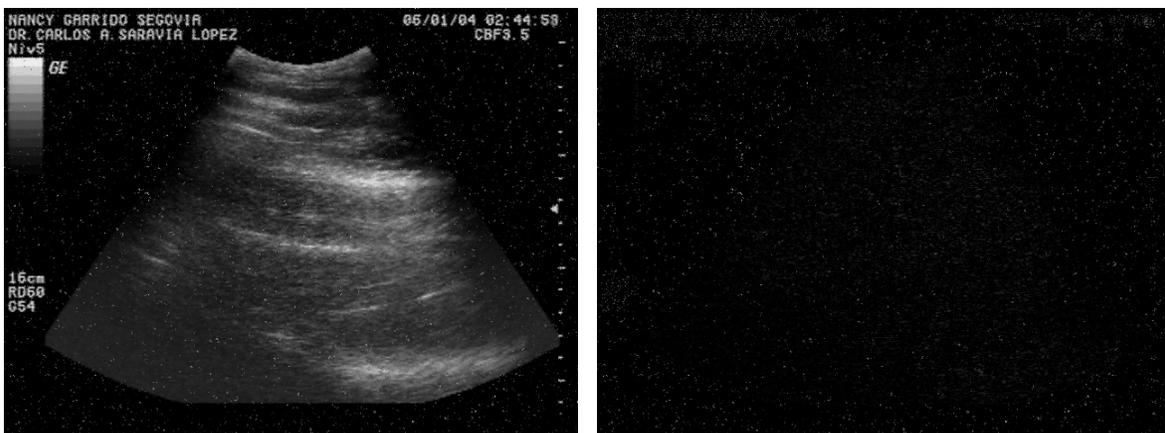


Figura 5.18. Imagen restaurada con el filtro MSM5 y su imagen de error, corte del plano XY.

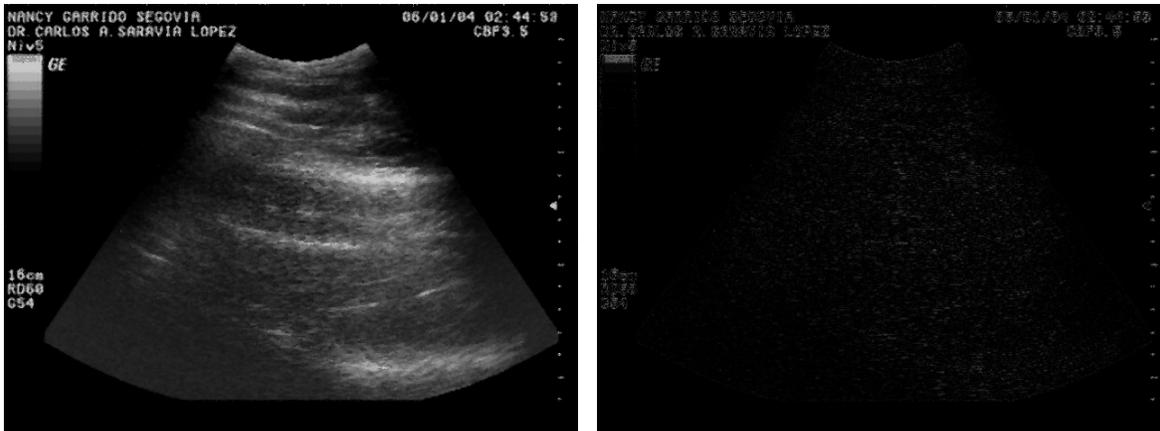


Figura 5.19. Imagen restaurada con el filtro MMKNN con F.I. Hampel y su imagen de error, corte del plano XY.

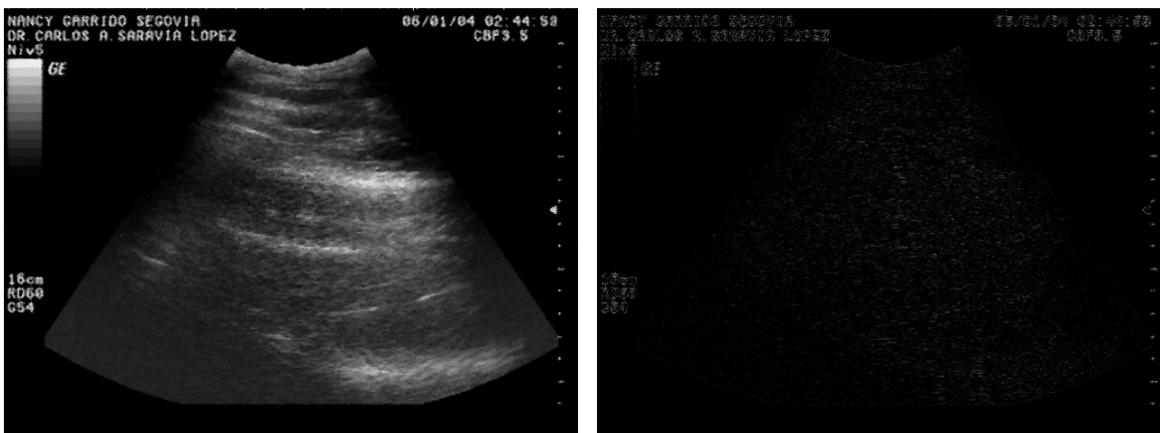


Figura 5.20. Imagen restaurada con el filtro MMKNN con F.I. Cut y su imagen de error, corte del plano XY.

En la figura 5.17 se muestra la imagen restaurada y de error del filtro LUM Smooth. Se observa en la imagen restaurada que el filtro LUM Smooth no suprime las manchas de ruido por completo, y en la imagen de error, hay mejor preservación en los detalles finos, ya que se definen las manchas de ruido y pocos detalles finos de la imagen. En la figura 5.18 se muestran los resultados del filtro MSM5. En la imagen restaurada hay menos manchas de ruido en comparación con el filtro LUM Smooth, y por tal motivo hay mayor preservación en los detalles finos de la imagen. En la figura 5.19, la imagen restaurada con el filtro MMKNN con F.I. Hampel, suprimió por completo las manchas, y definió mucho mejor los contornos de la imagen. En la imagen de error del filtro MMKNN con F.I. Hampel, se observa que el filtro suprimió los detalles finos de la imagen. La imagen restaurada con el filtro MMKNN con F.I. Cut que se muestra en la figura 5.20, suprimió por completo las manchas y también definió mucho mejor los contornos de la imagen, y en la imagen de error se observa que también suprimió los detalles finos de la imagen.

Como se describió anteriormente, las manchas y los detalles finos (manchas ocasionadas por el transductor coherente) que no pudieron suprimir los filtros LUM Smooth y MSM5, se debe a que más de la mitad o todos los datos del cubo de barrido sean ruido. Por este motivo los filtros LUM Smooth y MSM5, no logran llegar a una estimación correcta del voxel, preservando las manchas de ruido y las manchas del transductor coherente del equipo de ultrasonido. En cambio el filtro MMKNN, con el detector de impulsos y las funciones de influencia Cut y Hampel, suprimen por completo las manchas y definen mejor los contornos en la imagen 3D.

En el anexo B se observan las imágenes resultantes de los planos XZ y YZ.

V.5.3. Imágenes Resultantes al Suprimir 10% de Ruido Impulsivo

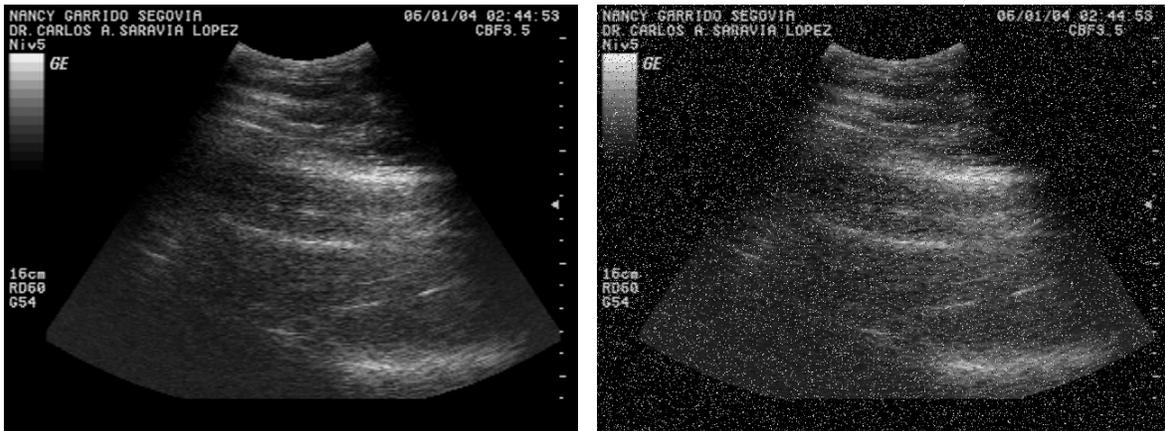


Figura 5.21. Imagen original e imagen degradada con 10% de ruido impulsivo, corte del plano XY.

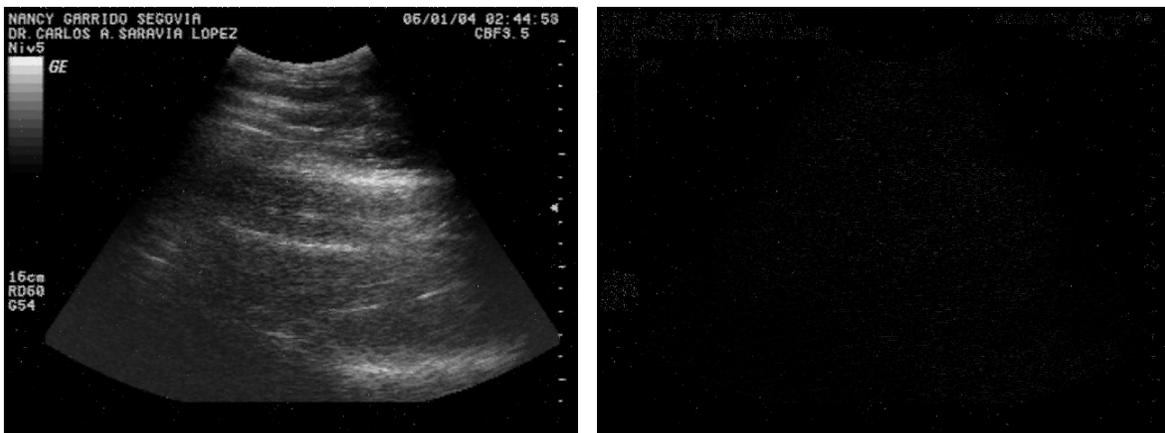


Figura 5.22. Imagen restaurada con el filtro LUM Smooth y su imagen de error, corte del plano XY.

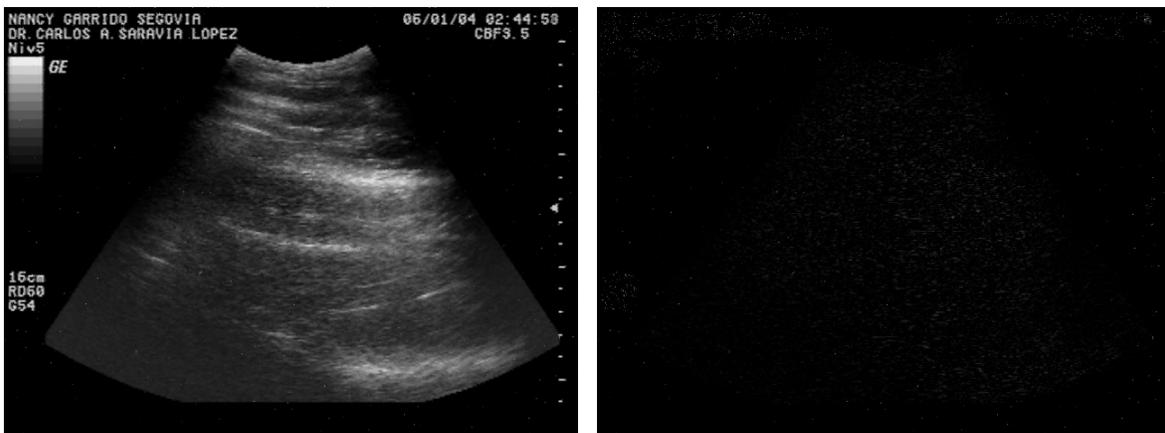


Figura 5.23. Imagen restaurada con el filtro MSM5 y su imagen de error, corte del plano XY.

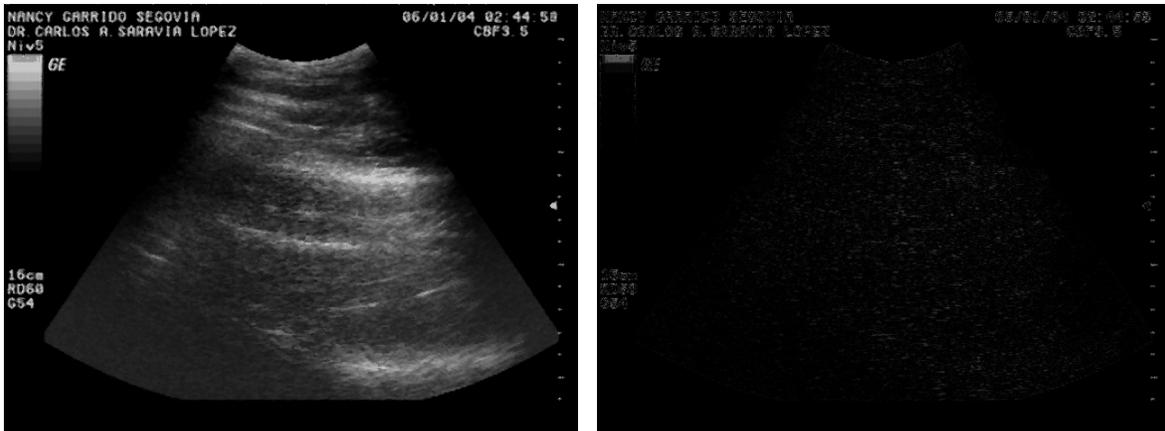


Figura 5.24. Imagen restaurada con el filtro MMKNN con F.I. Hampel y su imagen de error, corte del plano XY.

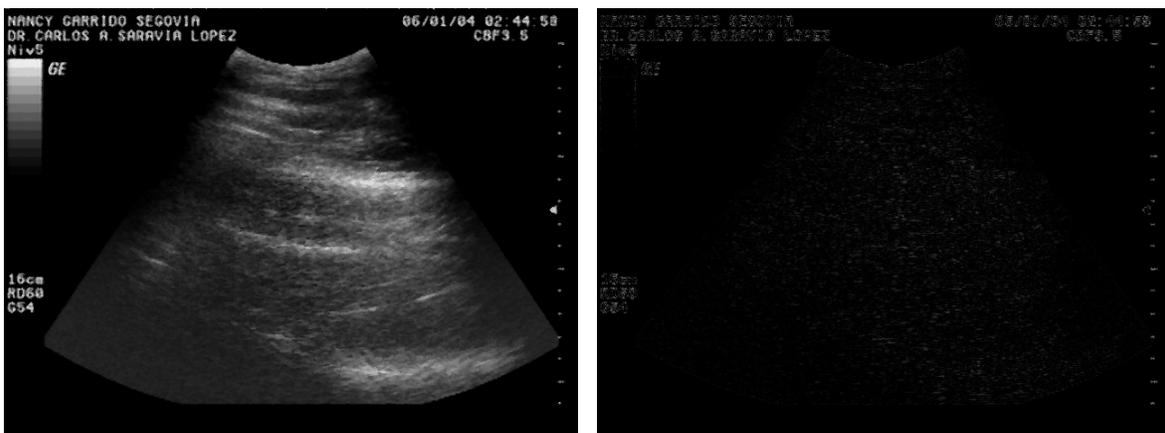


Figura 5.25. Imagen restaurada con el filtro MMKNN con F.I. Cut y su imagen de error, corte del plano XY.

Como se observa en la figura 5.22 hay muy pocas manchas. El filtro LUM Smooth no suprimió por completo las manchas, pero preservó mejor los detalles finos de la imagen. El desempeño del filtro MSM5, que se muestra en la figura 5.23, suprimió más manchas que el filtro LUM Smooth, y por eso preservó un poco mejor los detalles finos. En la figura 5.24, la imagen restaurada con el filtro MMKNN con F.I. Hampel, suprimió por completo las manchas, y definió más los contornos de la imagen. En la imagen de error del filtro MMKNN con F.I. Hampel, se observa que los detalles finos suprimidos de la imagen son más grandes. La imagen restaurada con el filtro MMKNN con F.I. Cut que se muestra en la figura 5.25, suprimió por completo las manchas y también definió más los contornos de la imagen, y en la imagen de error se observa que también los detalles finos suprimidos de la imagen aumentaron de tamaño.

Los filtros LUM Smooth y MSM5 preservan mejor los detalles finos de la imagen que el filtro MMKNN con funciones de influencia Cut y Hampel, pero no suprimen por completo el ruido y es donde los filtros MMKNN son mejores.

En el anexo B se observan las imágenes resultantes de los planos XZ y YZ.

Como se observó en las imágenes restauradas y de error de los filtros LUM Smooth y MSM5, y los resultados de la tabla 5.3, los filtros propuestos en literatura no son eficientes para suprimir el ruido impulsivo, ni mucho menos para suprimir el ruido ocasionado por el transductor coherente del equipo de ultrasonido que se manifiesta en detalles finos de la imagen [28]. En cambio el filtro MMKNN, con el detector de impulsos y las funciones de influencia, suprimen por completo las manchas y definen los contornos en la imagen 3D.

En las imágenes de los diferentes cortes en los planos XY, YZ y XZ, se observa que el filtro MMKNN suprime mayores porcentajes de ruido impulsivo, sin dejar manchas como los de mas filtros propuestos en literatura, y definen los contornos de la imagen. El filtro MMKNN tiene la propiedad de restaurar el voxel central del cubo de barrido, a pesar de que más de la mitad o todos los datos del cubo de barrido sean ruido, esto se debe al detector de picos que implementa y a las funciones de influencia.

Las imágenes originales adquiridas por el equipo de ultrasonido, contienen manchas debido al transductor coherente que utiliza [28-30]. Este tipo de ruido se llama Speckle, figura 5.26.



Figura 5.26. Imagen de las manchas ocasionadas por el transductor coherente del equipo de US.

Como se observó en las imágenes de error del filtro MMKNN, al suprimir diferentes porcentajes de ruido impulsivo, también suprime los detalles finos de la imagen original o más bien, las manchas ocasionadas por el transductor coherente del equipo de ultrasonido. El filtro MMKNN suprime las manchas o ruido Speckle de la imagen original, como lo hace al suprimir las manchas que se presentan a mayor porcentaje de ruido impulsivo, que los de mas filtros no pueden suprimir. Por esta razón; el filtro MMKNN suprime los detalles finos de la imagen original, ya que estos detalles son manchas ocasionados por el transductor coherente del equipo de ultrasonido [28-30]. Al suprimir las manchas ocasionadas por el transductor coherente del equipo de ultrasonido, se definen los contornos de los órganos, que se encontraban borrosos o difusos por las manchas.

A continuación se presentan las imágenes originales, las imágenes restauradas con el filtro MMKNN con función de influencia Cut y las imágenes de error, de los cortes en los planos XY, YZ y XZ. Las siguientes imágenes pertenecen al músculo del antebrazo derecho.

V.5.4. Imágenes del Músculo del Antebrazo

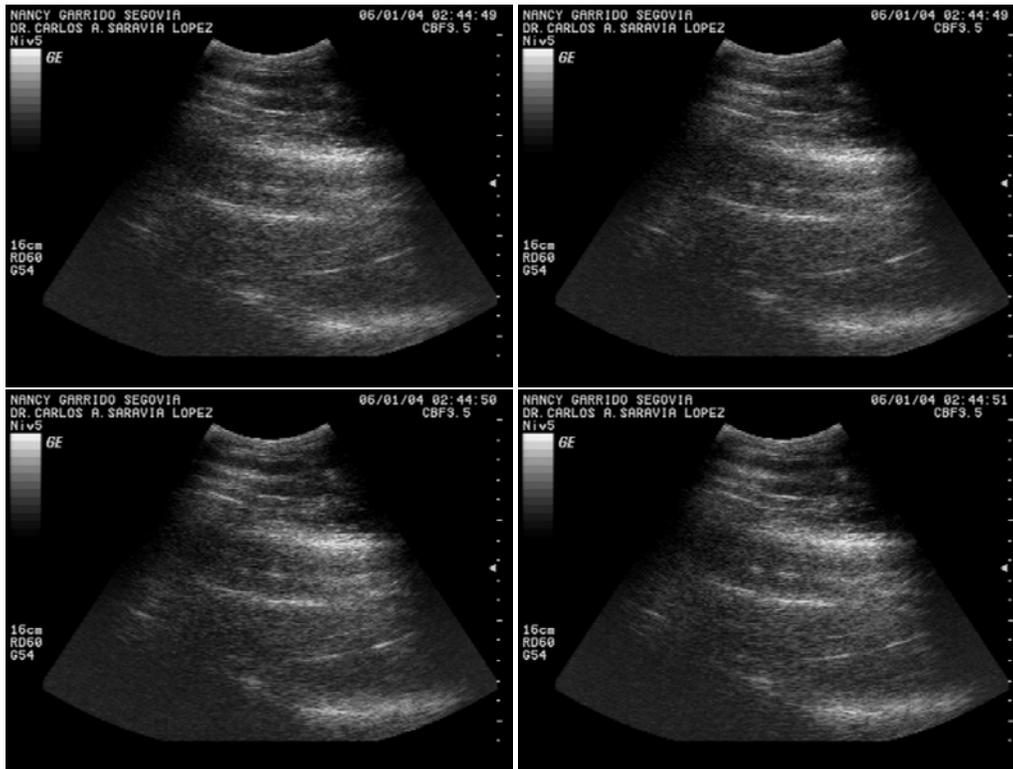


Figura 5.27. Imágenes originales, cortes del plano XY.

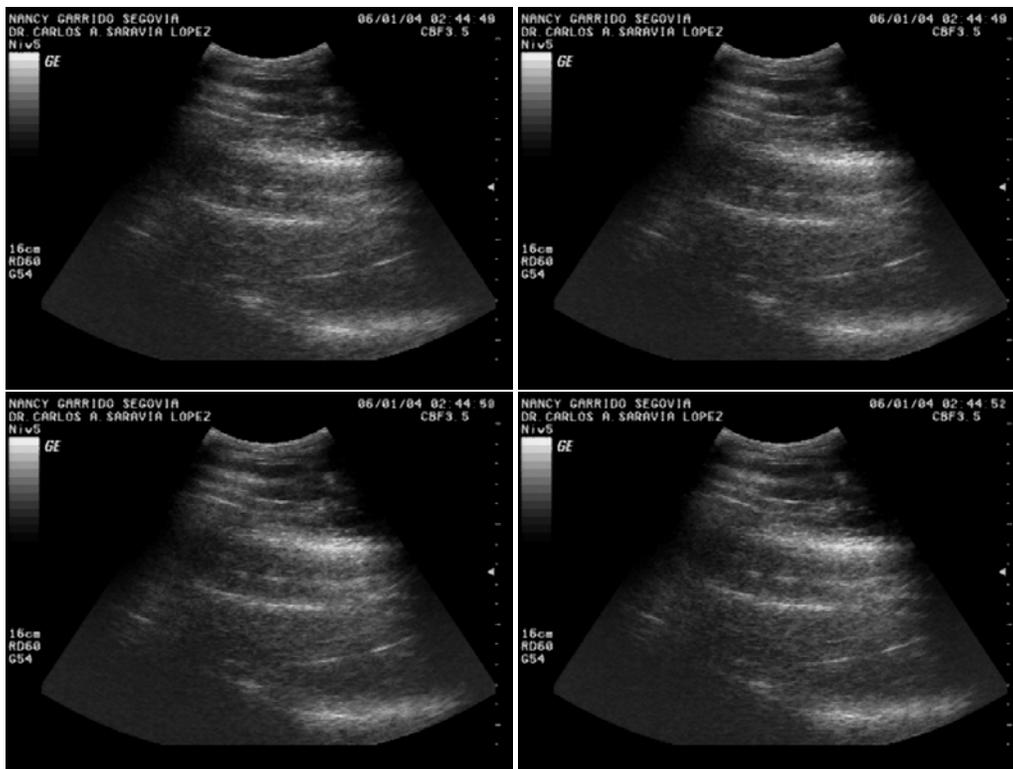


Figura 5.28. Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XY.

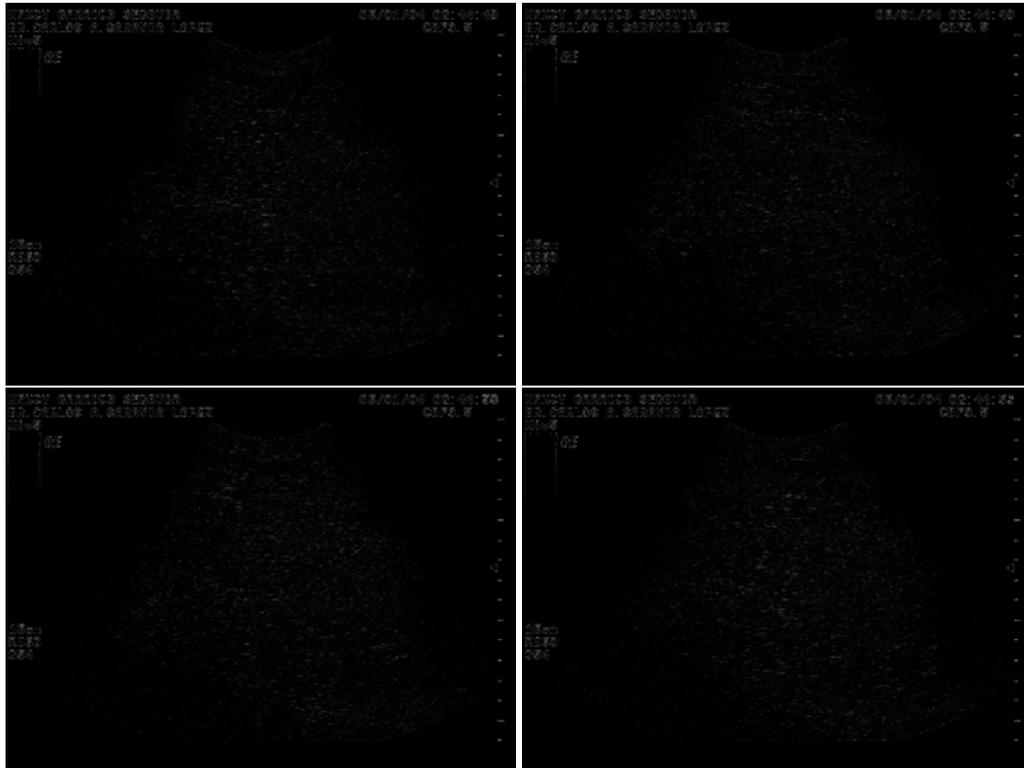


Figura 5.29. Imágenes de error, cortes del plano XY.

En las imágenes restauradas se definieron los detalles finos y los contornos del músculo. Y en las imágenes de error del músculo del antebrazo, se observan las manchas del transductor coherente del equipo de ultrasonido, que suprimió el filtro MMKNN con función de influencia Cut. En el anexo B se muestran las imágenes originales, restauradas y de error de los planos XZ y YZ del músculo del antebrazo. También se puede apreciar en el anexo B las imágenes originales, restauradas y de error del hígado y riñón, en los planos XY, YZ y XZ.

V.6. Ventana versus Cubo

Hay que tomar en cuenta un punto muy importante en el procesamiento digital de imágenes 3D: el cubo de barrido. En las pruebas realizadas anteriormente, los filtros 3D implementaron un cubo de barrido de 3x3x3 voxels. Con los resultados obtenidos, ya tenemos una base o un punto de referencia para implementar y proponer diferentes cubos de barrido.

Se implementaron 8 cubos diferentes y la ventana de barrido, para comparar su desempeño en imágenes y filtros 3D. A continuación se muestran las figuras y los resultados obtenidos al implementarse estos cubos con los filtros MMKNN con funciones de influencia Cut y Hampel, y los filtros Modified Trimmed Mean y Alfa Trimmed Mean, figura 5.30.

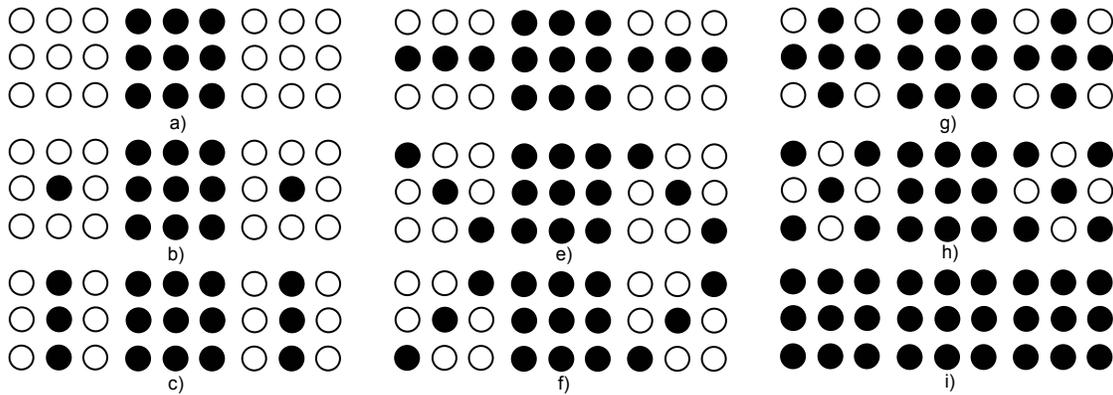


Figura 5.30. Ventana a) y cubos de barrido.

Tabla 5.5. Cubos implementados con el filtro MMKNN con F.I. Cut.

| Cubo de Barrido | Ruido Impulsivo | | | | | |
|-----------------|-----------------|-----------------|----------------|-----------------|----------------|-----------------|
| | 10% | | 20% | | 30% | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| a | 3.64016 | 31.17535 | 4.53838 | 28.40849 | 6.99018 | 23.90324 |
| b | 3.76703 | 31.30555 | 4.41531 | 29.41062 | 6.33257 | 25.35587 |
| c | 4.79816 | 29.59351 | 5.28413 | 28.76898 | 6.49814 | 26.50147 |
| d | 4.70299 | 29.62101 | 5.15630 | 28.85542 | 6.35310 | 26.60073 |
| e | 4.80637 | 28.50828 | 5.28963 | 28.70908 | 6.50239 | 26.46848 |
| f | 4.81058 | 29.49613 | 5.29791 | 28.68377 | 6.51271 | 26.45389 |
| g | 4.75225 | 28.97272 | 5.23398 | 28.43172 | 6.13451 | 27.10414 |
| h | 4.88553 | 28.76109 | 5.38470 | 28.19258 | 6.29676 | 26.89461 |
| i | 4.56200 | 28.51900 | 5.13600 | 27.91900 | 5.81900 | 27.22200 |

En la tabla 5.5 se observa que los cubos *g* e *i* son los más eficientes para suprimir mayor porcentaje de ruido, estos cubos manejan 19 y 27 datos, respectivamente. La diferencia que existe entre el cubo *g* e *i*, en relación al PSNR y MAE es muy poca. Y los cubos *a* y *b* suprimen de manera eficiente porcentajes pequeños de ruido, donde *a* es la ventana de barrido.

Tabla 5.6. Cubos implementados con el filtro MMKNN con F.I. Hampel.

| Cubo de Barrido | Ruido Impulsivo | | | | | |
|-----------------|-----------------|-----------------|----------------|-----------------|----------------|-----------------|
| | 10% | | 20% | | 30% | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| a | 3.64016 | 31.17535 | 4.53838 | 28.40849 | 6.99018 | 23.90324 |
| b | 3.76704 | 31.30552 | 4.41531 | 29.41062 | 6.33257 | 25.35587 |
| c | 4.82689 | 29.54799 | 5.31043 | 28.73098 | 6.52310 | 26.48153 |
| d | 4.73284 | 29.57445 | 5.18283 | 28.81697 | 6.37842 | 26.57975 |
| e | 4.83484 | 29.46396 | 5.31588 | 28.67159 | 6.52729 | 26.44891 |
| f | 4.83910 | 29.45166 | 5.32338 | 28.64825 | 6.53712 | 26.43478 |
| g | 4.78368 | 28.93201 | 5.26115 | 28.39827 | 6.15712 | 27.09630 |
| h | 4.91474 | 28.72463 | 5.41013 | 28.16442 | 6.31765 | 26.89053 |
| i | 4.59600 | 28.48000 | 5.16400 | 27.89000 | 5.84300 | 27.21200 |

En la tabla 5.6 también sobresalen los cubos g e i , siendo los más eficientes para suprimir mayor porcentaje de ruido. La diferencia que existe entre el cubo g e i , en relación al PSNR y MAE también es muy poca. Y los cubos a y b suprimen de manera eficiente porcentajes pequeños de ruido.

Tabla 5.7. Cubos implementados con el filtro Alfa Trimmed Mean.

| Cubo de Barrido | Ruido Impulsivo | | | | | |
|-----------------|-----------------|-----------------|----------------|-----------------|----------------|-----------------|
| | 10% | | 20% | | 30% | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| a | 4.66202 | 30.07507 | 6.97970 | 26.31513 | 11.66430 | 22.28058 |
| b | 4.36141 | 30.85167 | 5.68776 | 28.23672 | 9.17320 | 24.16346 |
| c | 4.96245 | 29.63262 | 5.48613 | 28.74842 | 7.43995 | 26.13527 |
| d | 4.83643 | 29.73005 | 5.34850 | 28.87602 | 7.29035 | 26.23982 |
| e | 4.96895 | 29.53979 | 5.49478 | 28.68096 | 7.44773 | 26.09710 |
| f | 4.97610 | 29.51616 | 5.50289 | 28.65847 | 7.45916 | 26.08362 |
| g | 5.41481 | 28.65049 | 5.68411 | 28.29551 | 6.85965 | 26.86466 |
| h | 5.57233 | 28.40716 | 5.85165 | 28.03793 | 7.03270 | 26.64502 |
| i | 7.41100 | 26.03500 | 7.76400 | 25.74500 | 8.58600 | 25.21500 |

En la tabla 5.7 se observa que el cubo g es el más eficiente para suprimir mayor porcentaje de ruido. Y los cubos a y b suprimen mejor pequeños porcentajes de ruido impulsivo.

Tabla 5.8. Cubos implementados con el filtro Modified Trimmed Mean.

| Cubo de Barrido | Ruido Impulsivo | | | | | |
|-----------------|-----------------|-----------------|----------------|-----------------|----------------|-----------------|
| | 10% | | 20% | | 30% | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| a | 3.67000 | 31.11000 | 4.55000 | 28.39000 | 6.98000 | 23.90000 |
| b | 3.78000 | 31.26000 | 4.41000 | 29.40000 | 6.30000 | 25.36000 |
| c | 4.66000 | 29.70000 | 5.07000 | 28.89000 | 6.29000 | 26.55000 |
| d | 4.52000 | 29.82000 | 4.93000 | 29.01000 | 6.13000 | 26.68000 |
| e | 4.67000 | 29.61000 | 5.07000 | 28.82000 | 6.30000 | 26.51000 |
| f | 4.68000 | 29.60000 | 5.08000 | 28.79000 | 6.31000 | 26.50000 |
| g | 5.21000 | 28.61000 | 5.51000 | 28.23000 | 6.36000 | 26.90000 |
| h | 5.39000 | 28.35000 | 5.69000 | 27.95000 | 6.54000 | 26.67000 |
| i | 7.10000 | 24.91000 | 7.34000 | 24.71000 | 7.79000 | 24.27000 |

En la tabla 5.8 también el cubo g es el más eficiente para suprimir mayor porcentaje de ruido. Y los cubos a y b suprimen mejor porcentajes pequeños de ruido.

Con el cubo de barrido i , el filtro MMKNN obtiene su mayor desempeño para suprimir grandes porcentajes de ruido impulsivo. El cubo de barrido g sobresalió para suprimir porcentajes grandes de ruido impulsivo, implementando menos datos. A continuación se presentan las imágenes restauradas y de error con los diferentes cubos de barrido, implementando el filtro MMKNN con la función de influencia Cut.

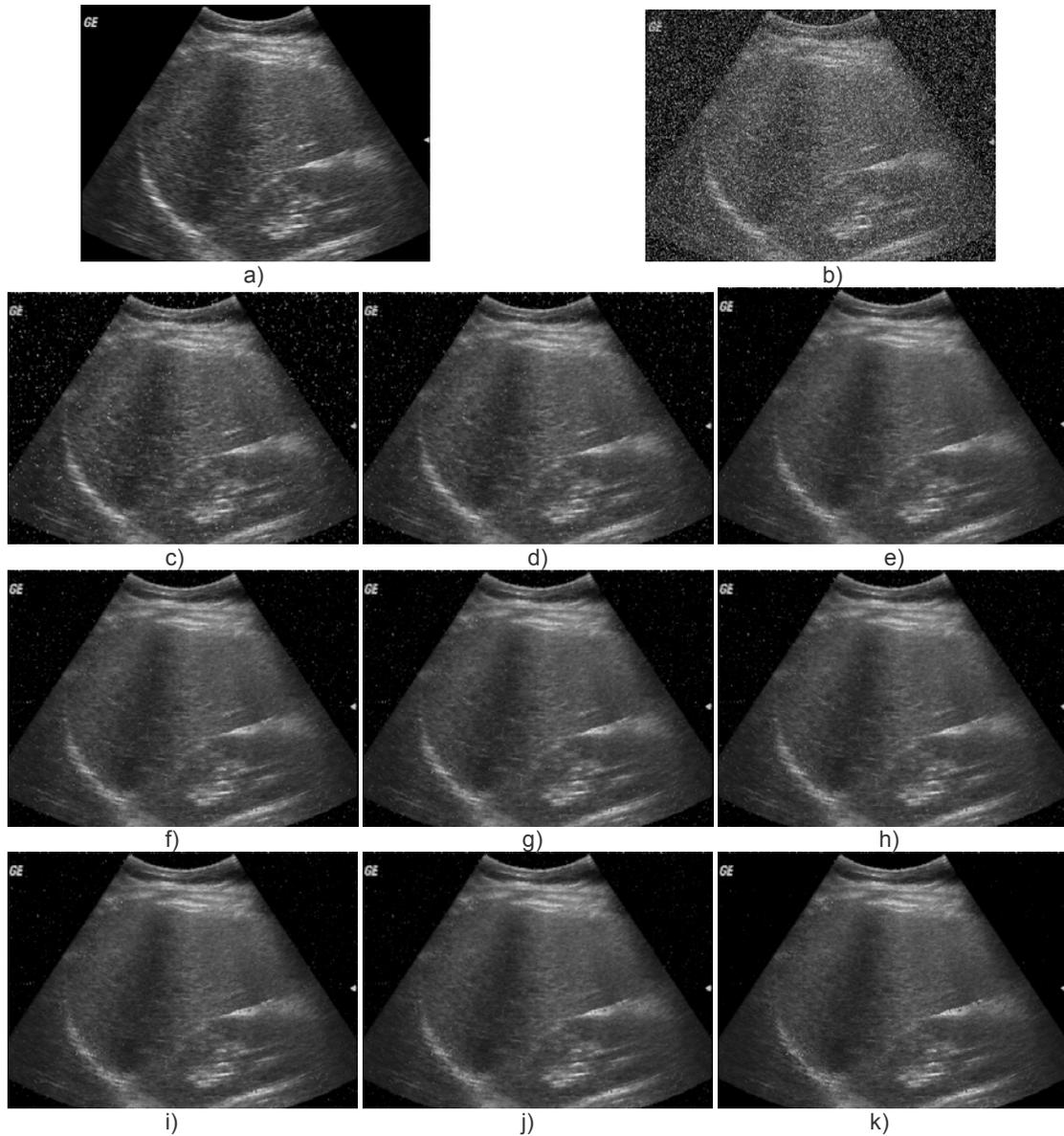


Figura 5.31. a) imagen original, b) imagen con 30% de ruido impulsivo, imagen restaurada con el filtro MMKNN con F.I. Cut implementando la c) ventana "a", d) el cubo "b", e) el cubo "c", f) el cubo "d", g) el cubo "e", h) el cubo "f", i) el cubo "g", j) el cubo "h", k) el cubo "i".

Analizando la figura 5.31, la imagen del c) tiene muchas manchas de ruido impulsivo, y no se definen los contornos, del d) las manchas disminuyen un poco, del e), f), g) y h) hay menos manchas y se definen los contornos, del i) disminuyeron las manchas de ruido impulsivo y se definieron mejor los contornos, del j) tiene manchas y se definen los contornos y el k) tiene menos manchas en comparación con los demás casos y se definen mucho mejor los contornos.

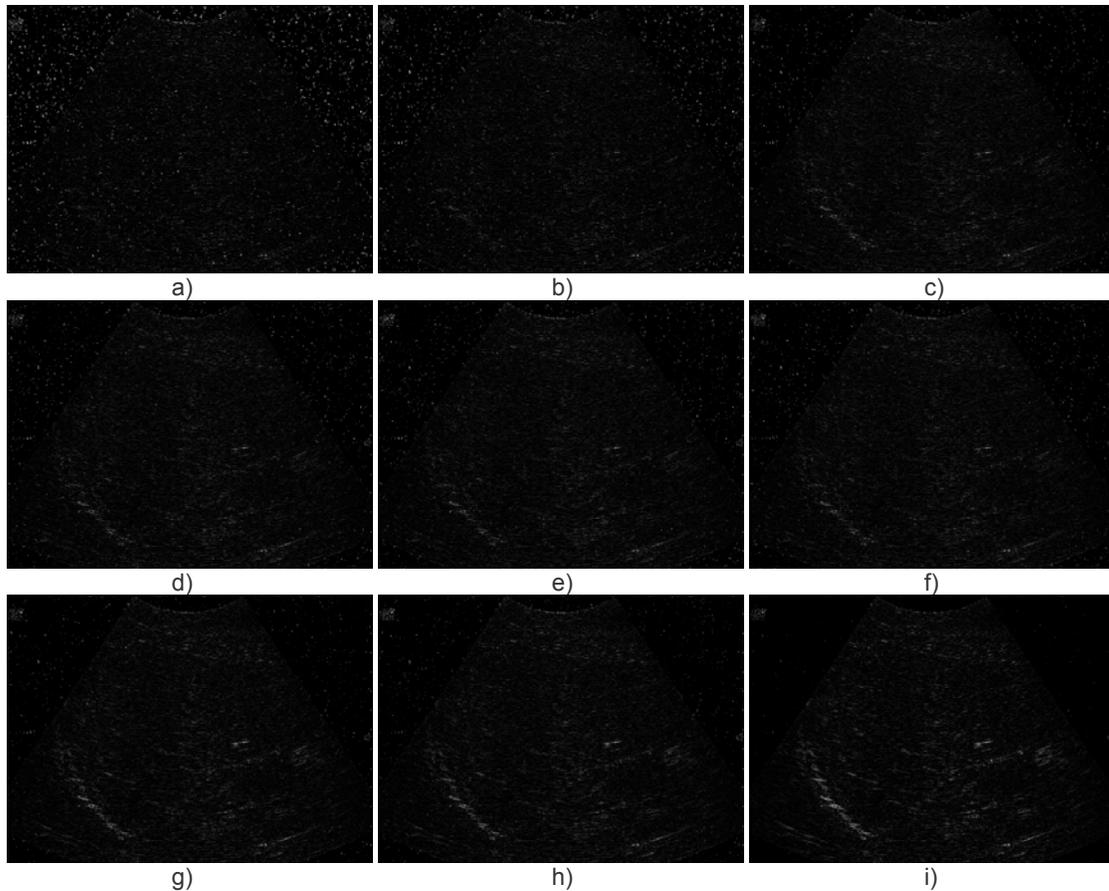


Figura 5.32. Imágenes de error del filtro MMKNN con F.I. Cut implementando la a) ventana "a", b) el cubo "b", c) el cubo "c", d) el cubo "d", e) el cubo "e", f) el cubo "f", g) el cubo "g", h) el cubo "h", i) el cubo "i".

En la figura 5.32, el filtro MMKNN con F.I. Cut a) no suprime las manchas del transductor coherente, por este motivo no se definen los contornos, del b), c), d), e) y f) se suprimen un poco las manchas del transductor y se definen los contornos, del g), h) y el i) suprimen mejor las manchas del transductor y se definen mejor los contornos.

V.7. Conclusiones

Se presentó el filtrado en 3D, y la adaptación de los filtros 2D a filtros 3D.

Se implementaron los filtros 3D para suprimir el ruido Gaussiano en las imágenes 3D, obteniendo como mejor filtro el Alfa Trimmed Mean 3D.

Las imágenes originales tenían manchas ocasionadas por el transductor coherente del equipo de ultrasonido. Por estas manchas los contornos estaban borrosos o difusos. Este problema se presenta en todos los equipos de ultrasonido [28-30]. Lo importante en el procesamiento de las imágenes de ultrasonido; es definir los contornos de los órganos en la imagen.

En el caso de ruido impulsivo, el mejor filtro es el MMKNN 3D con las funciones de influencia Cut y Hampel. Este filtro suprimió por completo los diferentes porcentajes de ruido impulsivo y también suprimió el ruido Speckle. Al suprimir las manchas ocasionadas por el transductor, se definieron los contornos de la imagen, que se encontraban difusos o borrosos.

El filtro MMKNN 3D con función de influencia Cut es el más óptimo para restaurar las imágenes 3D de ultrasonido. Este filtro se implementó para restaurar dos imágenes 3D: la primera imagen 3D es del músculo del antebrazo y la segunda imagen 3D es del hígado y riñón. En las dos imágenes 3D restauradas se definieron los contornos de los órganos.

Se presentaron diferentes cubos de barrido y se implementaron en algunos filtros 3D. El cubo *g* que sobresalió como el mejor para suprimir mayores porcentajes de ruido impulsivo, implementa 19 datos. Esta es una ventaja muy grande que se refleja en el tiempo de procesamiento.

Lo importante de las pruebas realizadas con los diferentes cubos de barrido, es para buscar un equilibrio entre la calidad de la imagen restaurada y el tiempo de procesamiento que se analiza en el capítulo VI.

CAPITULO VI. IMPLEMENTACION DE LOS FILTROS DE ESTADÍSTICA DE ORDEN EN EL DSP TMS320C6711 PARA PROCESAR IMÁGENES 3D

VI.1. Introducción

La mayoría de los sistemas de captura de imágenes médicas 3D, procesan estas imágenes en computadoras personales, por este motivo se ven obligados en disminuir las dimensiones de la imagen 3D original, limitar las dimensiones de la ventana de barrido, limitar los datos implementados en los filtros de estadística de orden o utilizar algoritmos de aproximación, todas estas restricciones se realizan para disminuir el tiempo de procesamiento [28,36,37].

En este capítulo se describe la implementación del DSP TMS320C6711 en imágenes 3D, utilizando los diferentes filtros de estadística de orden con diferentes porcentajes de ruido impulsivo.

VI.2. Procesamiento de Imágenes 3D

El procesamiento de una imagen 3D consiste en varias etapas, figura 6.1:

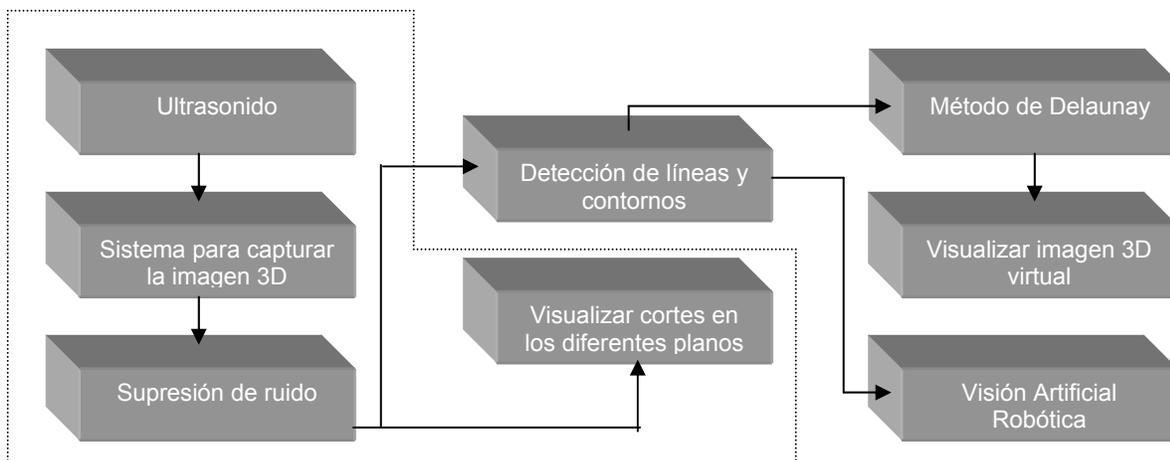


Figura 6.1. Diagrama a bloques del procesamiento de imágenes 3D.

En la figura 6.1, los bloques encerrados con la línea discontinua, representan el sistema propuesto en esta investigación.

En el procesamiento de imágenes 3D; cada etapa tiene que terminar por completo, para iniciar la siguiente, por este motivo el procesamiento de imágenes 3D, no se realiza en tiempo real. Las imágenes 3D se utilizan para diagnósticos, simulaciones o como mapas para la intervención quirúrgica.

Cada etapa expuesta en la figura 6.1, utiliza sus propios componentes, periféricos y algoritmos de procesamiento. A continuación se describe cada bloque a grandes rasgos:

- Etapa de captura de la imagen 3D: utiliza sistemas mecánicos, sistemas de posición, convertidores de analógicos a digitales y puertos de transferencias. El procesamiento del microcontrolador o DSP es de captura, sincronización y posición de cada imagen 2D.

- Etapa de supresión de ruido: es la implementación de los filtros de estadística de orden para supresión de ruido o restauración de la imagen 3D. Para este proceso se necesita un dispositivo rápido y preciso.
- Etapa de visualización de los cortes: es necesario demasiada memoria RAM en el dispositivo, para almacenar en este, toda la imagen 3D.
- Detección de líneas y contornos: depende por completo de la restauración de la imagen 3D. El proceso se lleva a cabo detectando las líneas y contornos de cada imagen 2D. Es muy diferente la detección de líneas y contornos de una imagen capturada con una cámara normal, a una imagen capturada con un equipo de ultrasonido, tomografía o resonancia magnética.
- Método de Delaunay: Se necesita memoria RAM para almacenar de 2 a 3 veces la imagen 3D, y un dispositivo diseñado para generar más de 800 000 tetraedros.
- Visualizar la imagen 3D: Es necesario suficiente memoria RAM.
- Robótica: esta etapa depende de la aplicación.

Cada etapa del procesamiento de la imagen 3D, se tiene que realizar con un dispositivo especial que cumpla con las exigencias. Los sistemas de tomografía o resonancia magnética utilizan varios microcontroladores o DSP's para las diferentes tareas.

VI.3. Dispositivos Usados para el Procesamiento de Imágenes 3D

Las ventajas de un DSP en comparación de una computadora personal son:

- Aplicaciones cíclicas, de duración acotada, donde se requiere alta eficiencia de ejecución.
- No suelen usar sistemas operativos, para evitar el overhead del scheduling.
- Uso de Assembler y dialectos especiales del lenguaje C para optimizar el código.
- Disponibilidad de modos de direccionamiento especializados.
- Varias unidades de procesamiento operando en forma paralela (MAC, Barrel Shift).
- Algunos DSPs con unidades de punto flotante.
- Pocos o nulos recursos que generan latencias, como memoria virtual, caches, etc.
- Arquitecturas tipo HARVARD con mapas de datos e instrucciones separados.
- Dos o más mapas de memoria de datos que permiten leer paralelamente operandos y coeficientes.
- Recursos internos o dispositivos periféricos especializados para la conversión A/D y D/A.
- Elevada capacidad de procesamiento aritmético de datos en tiempo real, con elevada precisión, para evitar problemas de redondeo y truncamiento.
- Etapas Multiplicadora/Acumuladora (**MAC**).
- Circuitos BS (**Barrel Shifter**) para desplazar un dato varios bits a derecha/izquierda en un único ciclo de instrucción.
- Una **ALU** operando en forma independiente al MAC y al BS.
- Códigos de operación para controlar al MAC, ALU y BS en una única instrucción (varias operaciones concurrentes).

Por estas razones se investigaron los DSP's de la Texas Instruments. Las características de la familia TMS320 son [38]:

1. Arquitectura Harvard.
2. Cuenta con un CPU avanzado, diseñado con un alto grado de paralelismo y un hardware con aplicaciones lógicas específicas para aumentar el desempeño.

3. Un juego de instrucciones, especializado para algoritmos rápidos (procesamiento en tiempo real).
4. Diseño de la arquitectura modular, para el desarrollo rápido de los dispositivos.
5. Circuitos integrados tecnológicamente avanzados para incrementar el desempeño y el bajo consumo de potencia.
6. Alta inmunidad a la radiación, debido a nuevas técnicas de planes de estática.

La arquitectura Harvard no utiliza un mismo bus para datos y direcciones, sino al contrario; utiliza un bus para datos y un bus para direcciones, haciendo más rápido la transferencia de datos. El CPU se diseñó para realizar procesos paralelos o cargar un dato y ejecutar una instrucción al mismo tiempo. Estas características lo hacen ideal para el procesamiento de imágenes 3D.

VI.3.1. TMS320C6711

La capacidad del TMS320C6711 es de alto desempeño, la arquitectura VLIW (advanced very long instruction word) hace al DSP una buena herramienta para aplicaciones en multicanales y multifunciones. La ejecución de hasta 1 GFLOPS con una velocidad de reloj de 167MHz, el 'C6711 ofrece soluciones efectivas de alto desempeño para los desafíos en la programación de DSPs [39].

El 'C6711 tiene 32 registros de propósito general de 32 bits cada uno; 8 unidades que proveen 4 ALUs de punto flotante/fijo, 2 ALUs de punto fijo y 2 multiplicadores de punto flotante/fijo. El 'C6711 incluye una memoria de 1Mb y un completo conjunto de periféricos. La memoria del programa es de 64Kb, que puede ser usada como cache o mapa de memoria para el programa. La memoria de datos tiene 2 bloques de RAM de 32Kb. Los periféricos incluyen 2 puertos multicanales serie, 2 "timers" de propósito general y una interfase de memoria externa capaz de conectar SDRAM y periféricos asíncronos [39].

El 'C6711 tiene un conjunto completo de herramientas de desarrollo que incluyen: un compilador de C, un optimizador de ensamblador para simplificar la programación y las tareas, y una interfase de depurador para Windows para visualizar el código en ejecución. En la figura 6.2 se muestra el modulo de evaluación del TMS320C6711, en el cual algunos de sus principales componentes son señalados [39].

VI.3.2. FPGA

Los dispositivos FPGA permiten integrar sistemas en un solo chip equivalentes a diseños de millones de compuertas digitales. El diseño con FPGAs es similar a los dispositivos ASIC, con las ventajas de tener un ciclo de desarrollo mucho mas corto (semanas en lugar de meses) y un costo reducido. Las nuevas herramientas y los lenguajes VHDL y Handel-C acercan el diseño hardware con metodologías de diseño software permitiendo la creación de sistemas hardware-software en un solo chip.

Un FPGA es un arreglo de bloques lógicos programables colocados en una infraestructura de interconexiones programable; es posible programar la funcionalidad de los bloques lógicos, las interconexiones entre bloques y las conexiones entre entradas y salidas. Un FPGA es programable a nivel hardware. Así, un FPGA proporciona las ventajas de un procesador de propósito general y un circuito especializado que puede reconfigurarse las veces que sea necesario para depurar su funcionalidad. El tamaño y velocidad de los FPGAs son equiparables a los ASICs, pero los FPGAs son más flexibles y su ciclo de diseño es más corto.

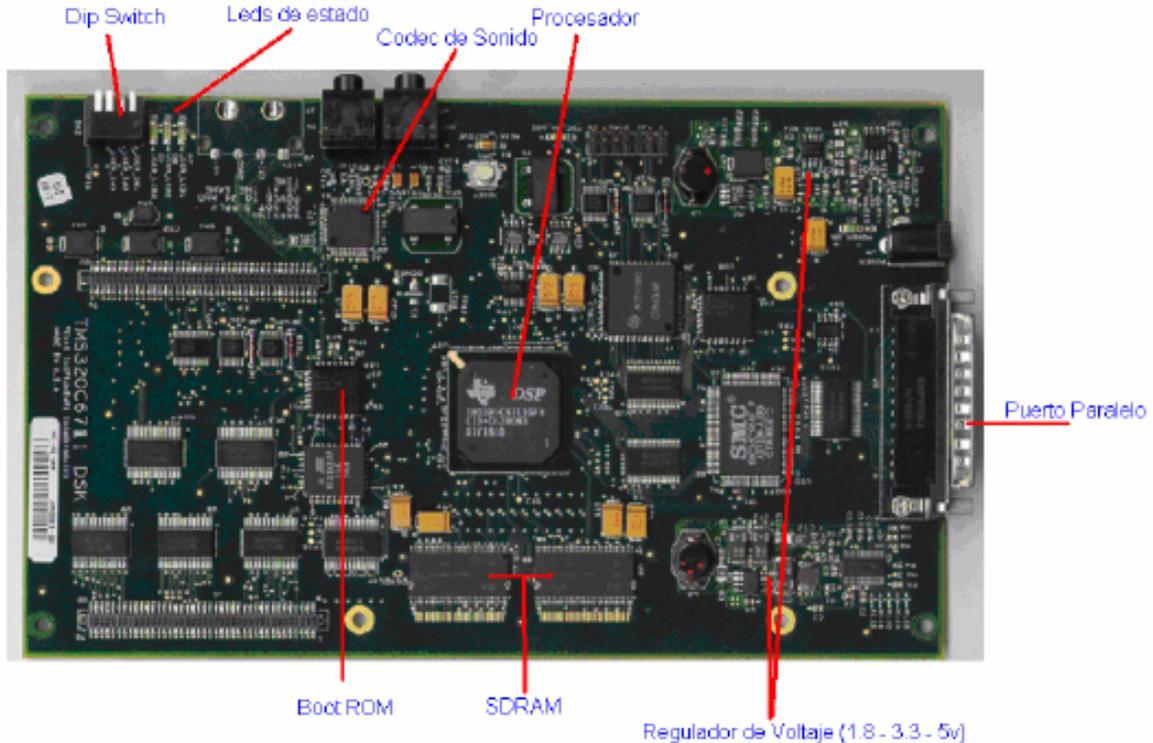


Figura 6.2. Tarjeta de evaluación del TMS320C6711.

Las aplicaciones de los FPGAs van más allá de la simple implementación de lógica digital. Los FPGAs pueden ser utilizados para la implementación de arquitecturas específicas para acelerar algún algoritmo. Los sistemas basados en FPGAs proporcionan un mejor desempeño que sus correspondientes implementaciones software. Las aplicaciones que requieren de un gran número de **operaciones simples** son adecuadas para su implementación en FPGA puesto que un elemento de procesamiento puede diseñarse para efectuar esta operación y varias instancias de éste pueden reproducirse para llevar a cabo procesamiento paralelo. Los algoritmos de bajo nivel en visión por computadora y análisis de imágenes son candidatos ideales para explorar arquitecturas específicas que aceleren su rendimiento [40].



Figura 6.3. Tarjetas de desarrollo de FPGA.

VI.4. FPGA Versus DSP

Existen dos opciones para implementar el filtrado 3D: el FPGA y el DSP.

El FPGA está orientado:

- A la captura de imágenes en tiempo real.
- Visualización en pantalla.
- Detección de líneas y contornos en tiempo real.
- Reconstrucción tridimensional de objetos.

En sí, el FPGA se utiliza cuando se requieren procesar un gran número de operaciones simples y en la visualización de imágenes, donde no se implementan operaciones complejas relacionados con números fraccionarios [40].

El DSP nos da muchas ventajas:

Partiendo del DSP TMS320C54x, donde se implementaron:

- Filtros no lineales en audio en tiempo real.
- En el procesamiento de dos imágenes paralelamente para detección de líneas y contornos.
- En el procesamiento del gradiente X y Y paralelamente en una imagen.
- Visualización en pantalla por medio del Host Port Interface del DSP.

Estos proyectos se realizaron utilizando el ensamblador algebraico, que es un lenguaje mucho más sencillo y fácil que el ensamblador que utiliza nemotécnicos. Se utilizó el procesamiento paralelo, donde lo complicado en esta herramienta es la distribución de los datos en la memoria. Se utilizó el formato Q15 y Q31, estos formatos se implementan de manera sencilla en el DSP para representar los números fraccionarios [38].

El DSP TMS320C6711 tiene las características del TMS320C54X, pero es más rápido, mayor memoria RAM, está diseñado para procesos más complejos en imágenes. Es muy importante mencionar que utiliza varios formatos para números fraccionarios. El DSP está orientado para manejar el punto flotante y realizar operaciones complejas (seno, coseno) [38, 39].

VI.5. Programación del TMS320C6711

Los algoritmos de todos los filtros 3D fueron implementados en lenguaje C mediante el uso del TURBO C++ 3.0 para todas las rutinas, estructuras de procesamiento de datos y operaciones de bajo nivel de I/O [41].

Existen dos formas para ejecutar los programas en lenguaje C dentro del DSP TMS320C6711, las cuales se describen a continuación:

- Ambiente DOS. Compilamos estos programas usando las herramientas de desarrollo como el compilador de C del 'C6711 para crear el archivo de ensamblador (*.asm), el archivo objeto (*.obj) y el archivo ejecutable COFF (*.out). El archivo *.out fue simulado en la PC usando el simulador del 'C6711.
- Ambiente Windows. Compilamos y ejecutamos los programas en C mediante el software de comunicación con el DSP llamado Code Composer Studio 2.0 de Texas Instruments, el cual contiene las mismas herramientas del ambiente DOS, solo que la diferencia consiste en que este software esta en ambiente Windows y las anteriores herramientas están en ambiente DOS lo que permite un mejor manejo de las herramientas de desarrollo.

VI.6. Implementación del TMS320C6711 en imágenes 3D

Las primeras pruebas realizadas con los filtros 3D (cubo de barrido de 3x3x3 voxels) implementando una imagen de 525x382x12 voxels, donde se hallaron los rendimientos de cada filtro 3D (MAE y PSNR), se realizaron en una computadora la cual tiene un procesador Pentium III a 1 GHz. y 128 Mbytes de memoria RAM. El tiempo de procesamiento por cada prueba en esa computadora en el entorno de MatLab (versión 6.0.0.88), es de aproximadamente de 4 a 5 horas, y en el entorno de MS-DOS utilizando C++ (versión 3.0) era de 20 a 30 minutos. En MatLab los tiempos de procesamiento son muy altos, debido a que se encuentra en el entorno de Windows, y en C++ los tiempos disminuyen considerablemente por el entorno de MS-DOS. Por estas razones se realizaron las pruebas de los filtros 3D (cubo de barrido de 3x3x3 voxels) con la misma imagen 3D (525x382x12 voxels) en el DSP TMS320C6711, para hallar los tiempos de procesamiento y compararlos con los de la computadora personal.

A continuación se presentan los tiempos de procesamiento de cada filtro, obtenidos por medio de simulaciones.

Tabla 6.1. Tiempo en segundos de los filtros 3D.

| Ruido Impulsivo (%) | Alfa T.M. | Modified T.M. | Ranked Order | MSM1 | MSM2 | MSM3 | MSM4 | MSM5 |
|---------------------|-----------|---------------|--------------|--------|--------|--------|--------|--------|
| 30 | 13.466 | 2.1716 | 1.6836 | 0.5846 | 0.5773 | 1.2681 | 1.2367 | 1.2198 |
| 25 | 12.599 | 2.1716 | 1.6836 | 0.5846 | 0.5773 | 1.2681 | 1.2367 | 1.2198 |
| 20 | 12.074 | 2.1716 | 1.6836 | 0.5846 | 0.5773 | 1.2681 | 1.2367 | 1.2198 |
| 15 | 11.741 | 2.1716 | 1.6836 | 0.5846 | 0.5773 | 1.2681 | 1.2367 | 1.2198 |
| 10 | 11.315 | 2.1716 | 1.6836 | 0.5846 | 0.5773 | 1.2681 | 1.2367 | 1.2198 |
| 5 | 10.660 | 2.1716 | 1.6836 | 0.5846 | 0.5773 | 1.2681 | 1.2367 | 1.2198 |
| 0 | 10.077 | 2.1716 | 1.6836 | 0.5846 | 0.5773 | 1.2681 | 1.2367 | 1.2198 |

Tabla 6.2. Tiempo en segundos de los filtros 3D.

| Ruido Impulsivo (%) | MSM5 | MaxMed | Comp. Selec. | Selec. Prom. | Selec. Mediana | LUM Smooth | LUM Sharp | LUM | Wilcoxon |
|---------------------|--------|--------|--------------|--------------|----------------|------------|-----------|-------|----------|
| 30 | 1.1667 | 1.1981 | 4.742 | 1.9620859 | 2.324 | 7.123 | 7.257 | 7.402 | 23.564 |
| 25 | 1.1667 | 1.1981 | 4.742 | 1.9620859 | 2.324 | 6.511 | 6.638 | 6.770 | 23.564 |
| 20 | 1.1667 | 1.1981 | 4.742 | 1.9620859 | 2.324 | 6.047 | 6.165 | 6.285 | 23.564 |
| 15 | 1.1667 | 1.1981 | 4.742 | 1.9620859 | 2.324 | 5.754 | 5.867 | 5.984 | 23.564 |
| 10 | 1.1667 | 1.1981 | 4.742 | 1.9620859 | 2.324 | 5.355 | 5.469 | 5.582 | 23.564 |
| 5 | 1.1667 | 1.1981 | 4.742 | 1.9620859 | 2.324 | 4.705 | 4.812 | 4.915 | 23.564 |
| 0 | 1.1667 | 1.1981 | 4.742 | 1.9620859 | 2.324 | 4.122 | 4.224 | 4.317 | 23.564 |

En la tabla 6.1 se observa que los tiempos de procesamiento del filtro Alfa Trimmed Mean son los más altos y varían de acuerdo al porcentaje ruido, esto se debe al algoritmo de ordenamiento de los 27 datos, que detecta cuando ya están ordenados sin tener que realizar n^2 ($n = \text{No. de datos}$) procesos de comparación, y con los resultados se observa que hay una relación con respecto al porcentaje de ruido.

En la tabla 6.1 y 6.2, están los tiempos de procesamiento de los filtros de multi-etapas, estos filtros implementan sub-cubos de barridos para disminuir el tiempo de procesamiento al hallar la mediana, y lo logran obteniendo los menores tiempo de procesamiento.

En la tabla 6.2, los filtros de selección de mediana y de promedio, tienen tiempos muy pequeños, esto se debe a que dividen el cubo de barrido en 2 grupos, hallando la mediana y el promedio de cada grupo, con mayor rapidez. Los filtros LUM Smooth, LUM Sharp y LUM, sus tiempo son un poco altos debido a que tienen que ordenar los 27 datos del cubo de barrido, e implementan el mismo algoritmo de ordenamiento que usa el filtro Alfa Trimmed Mean.

Tabla 6.3. Tiempo en segundos de los filtros 3D.

| Ruido | MMKNN | | | | |
|-------|--------|--------|--------|-----------|--------|
| | CUT | HAMPEL | SINE | BERNOULLI | TUKEY |
| 30 | 20.873 | 21.752 | 22.710 | 22.979 | 22.742 |
| 25 | 20.776 | 21.508 | 22.707 | 22.735 | 22.741 |
| 20 | 20.658 | 21.263 | 22.484 | 22.734 | 22.257 |
| 15 | 20.629 | 21.261 | 22.240 | 22.492 | 21.771 |
| 10 | 20.609 | 21.017 | 21.996 | 22.250 | 21.764 |
| 5 | 20.585 | 20.533 | 21.742 | 22.006 | 21.741 |

Tabla 6.4. Tiempo en segundos de los filtros 3D.

| Ruido | WMKNN | | ABSTM | |
|-------|--------|--------|--------|--------|
| | CUT | HAMPEL | CUT | HAMPEL |
| 30 | 45.170 | 47.549 | 39.699 | 42.134 |
| 25 | 45.093 | 47.537 | 39.627 | 40.498 |
| 20 | 44.993 | 46.052 | 37.066 | 38.082 |
| 15 | 44.918 | 45.061 | 36.800 | 38.000 |
| 10 | 44.912 | 42.622 | 32.440 | 36.631 |
| 5 | 42.489 | 40.194 | 32.392 | 35.522 |

En la tabla 6.3 se presenta el tiempo de procesamiento del filtro MMKNN, con las funciones de influencia Cut, Hampel, Sine, Bernoulli y Tukey. El tiempo de procesamiento es mayor que los primeros filtros implementados, pero los resultados obtenidos al suprimir el ruido impulsivo, son los mejores.

En la tabla 6.4 se presentan los filtros WMKNN y ABSTM con las funciones de Influencia Cut y Hampel. El tiempo de procesamiento es demasiado grande, por la implementación del estimador Wilcoxon.

Las tablas 6.5 y 6.6. presentan los resultados de los tiempos de los dos mejores filtros con los diferentes cubos de barrido presentados en el capítulo V, figura 6.4.

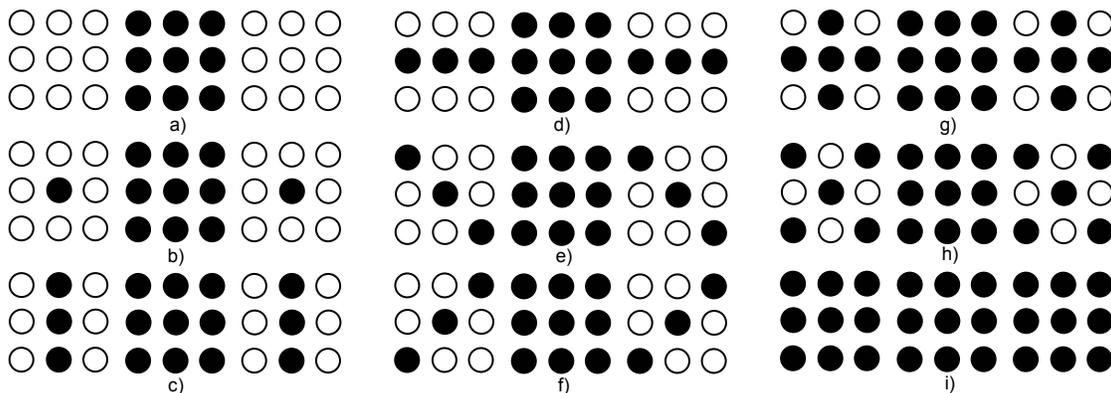


Figura 6.4. Ventana a) y cubos de barrido.

Tabla 6.5. Tiempo en segundos de los filtros 3D.

| Cubo de Barrido | Filtro MMKNN F.I. Hampel | | |
|-----------------|--------------------------|------------|------------|
| | Ruido Impulsivo (%) | | |
| | 10 | 20 | 30 |
| A | 1.7633303 | 1.7874855 | 1.8116407 |
| B | 2.0531928 | 2.077348 | 2.1015032 |
| C | 4.6184278 | 4.6349982 | 4.6926809 |
| D | 4.6957727 | 5.2010273 | 5.263855 |
| E | 4.6160606 | 4.6278966 | 4.6903137 |
| F | 4.6160606 | 4.6420999 | 4.6926809 |
| G | 9.93986874 | 10.0364896 | 10.0606448 |
| h | 9.96885499 | 10.0171654 | 10.0413206 |
| i | 21.0174479 | 21.2626232 | 21.7517662 |

En la tabla 6.5 se presentan los tiempos del filtro MMKNN con la función de influencia Hampel implementando los diferentes cubos de barrido. Para el 30% y 20%, el resultado más importante es del cubo *g*, comparándolo con el cubo *i*, es un poco menos de la mitad de tiempo. Y para el 10% de ruido, los cubos *a* y *b*, obtienen tiempos muy pequeños comparados con el cubo *i*.

Tabla 6.6. Tiempo en segundos de los filtros 3D.

| Cubo de Barrido | Filtro MMKNN F.I. Cut | | |
|-----------------|-----------------------|------------|------------|
| | Ruido Impulsivo (%) | | |
| | 10 | 20 | 30 |
| a | 1.5942438 | 1.6425543 | 1.6590039 |
| b | 1.8661349 | 1.9082616 | 1.9163052 |
| c | 4.8067659 | 4.8228291 | 4.8717434 |
| d | 4.7537452 | 5.1989499 | 5.2516083 |
| e | 4.8044712 | 4.8159449 | 4.8694487 |
| f | 4.8044712 | 4.8297134 | 4.8717434 |
| g | 10.0364896 | 10.0552099 | 10.0606448 |
| h | 10.0558137 | 10.0775534 | 10.079969 |
| i | 20.6092248 | 20.6575352 | 20.8728306 |

En la tabla 6.6 se presentan los tiempos del filtro MMKNN con la función de influencia Cut implementando los diferentes cubos de barrido. Para el 30% y 20%, el resultado más importante, también es el cubo *g*, comparándolo con el cubo *i*, es casi la mitad de tiempo. Y para el 10% de ruido, los cubos *a* y *b*, obtienen tiempos muy pequeños comparados con el cubo *i*.

Se compararon con el cubo *i*, ya que este es el cubo que se implementó en las pruebas realizadas en capítulo V.

VI.7. Conclusiones

No es viable implementar el filtrado 3D en una computadora personal, debido a los problemas que se comentaron en la sección VI.3, y que se reflejan en el tiempo de procesamiento. En las tablas 6.1 a 6.6 se observa la razón de utilizar un DSP. El DSP es una herramienta poderosa, enfocada al procesamiento de imágenes.

El procesamiento del filtrado 3D no se realiza en tiempo real, ya que primero se realiza la captura de la imagen 3D y después el filtrado 3D.

Los tiempos de las tablas 6.1 y 6.2, son de los filtros propuestos en literatura. Son tiempos muy pequeños, pero sus resultados con respecto al PSNR y MAE, no se comparan con los filtros de la tabla 6.3, que corresponden al filtro MMKNN.

Al implementarse los diferentes cubos de barrido en el filtro MMKNN con las funciones de influencia Cut y Hampel, se obtuvieron tiempos de procesamiento muy bajos. Donde la calidad de la restauración de la imagen 3D, no es afectada. Haciendo del filtro MMKNN con funciones de influencia Hampel y Cut; eficiente tanto en la calidad de restauración y el tiempo de procesamiento.

CAPITULO VII. CONCLUSIONES

VII.1. Conclusiones

Una imagen 3D se forma con una secuencia de n imágenes 2D, donde cada imagen tiene definida su posición. Partiendo de esta idea, se diseñó y desarrolló un sistema experimental para capturar imágenes de ultrasonido en 3D. Es sistema nos permitió introducirnos al nuevo campo de captura de las imágenes 3D por primera vez aplicado en México.

Las imágenes originales obtenidas con el equipo de ultrasonido, tienen manchas ocasionadas por el transductor coherente, este tipo de ruido se llama Speckle. Por el ruido Speckle, la imagen original es borrosa, debido a que los contornos no están definidos o se encuentran difusos por las manchas. Este problema se observa en todos los equipos de ultrasonido.

Se presentó el filtrado en tres dimensiones para la supresión de ruido de diferente naturaleza, adaptando los filtros de estadística de orden 2D, a filtros 3D. Se realizaron pruebas para suprimir el ruido Gaussiano e impulsivo en imágenes 3D, enfocándonos principalmente al ruido impulsivo.

Con las pruebas realizadas con diferentes porcentajes de ruido impulsivo, el filtro MMKNN 3D con la función de influencia Cut, es el más eficiente para suprimir el ruido impulsivo. Este filtro se implementó en las imágenes 3D originales, para suprimir las manchas ocasionadas por el transductor coherente del equipo de ultrasonido. En las imágenes 3D restauradas se definieron los contornos de los órganos y los detalles, que se encontraban difusos o indefinidos, por el ruido Speckle. Comparando la imagen original con la restaurada, como se observó en el capítulo V, las imágenes originales son borrosas y muchos detalles no se aprecian correctamente. Las imágenes 3D restauradas, permiten realizar un diagnóstico con mayor facilidad y precisión, observando el órgano desde diferentes perspectivas visuales.

Se implementó el filtrado 3D en el DSP TMS320C6711, y se obtuvieron los tiempos de procesamiento de los diferentes filtros 3D. Enfocándonos al filtro MMKNN con función de influencia Cut, al implementar un cubo de 27 elementos, el tiempo de procesamiento es muy alto, pero la restauración que realiza este filtro no se compara con los filtros propuestos en literatura. Para obtener un equilibrio entre la calidad y el tiempo de procesamiento, se propusieron diferentes cubos de barrido. Donde el cubo g definido en los capítulos V y VI, hace un equilibrio entre la calidad de la restauración de la imagen 3D y el tiempo de procesamiento.

El DSP TMS320C6711 nos permite realizar en cuestiones de segundos la restauración de imágenes 3D, ofreciendo una imagen 3D con mayor definición y nitidez para un mejor diagnóstico en un hospital o clínica.

Se diseñó y realizó un sistema con un propósito importante y necesario para el área de medicina. Y se aplicó a un problema real el filtrado 3D, obteniendo resultados muy importantes para el procesamiento de imágenes 3D en el área de medicina.

VII.2. Trabajos a Futuro

El procesamiento de imágenes 3D de ultrasonido se puede dividir en dos grupos muy importantes:

- La restauración de la imagen 3D.
- La visualización de la imagen 3D.

Como posibles temas en futuros trabajos, se consideran los siguientes:

1. Proponer nuevas técnicas de procesamiento en 3D con mejor rendimiento.
2. Proponer e investigar nuevas técnicas de desplazamiento del cubo de barrido.
3. Realizar la alineación de las imágenes 2D, para obtener mejores imágenes al realizar los diferentes cortes en los planos.
4. Desarrollar un método para formar la imagen 3D virtual.

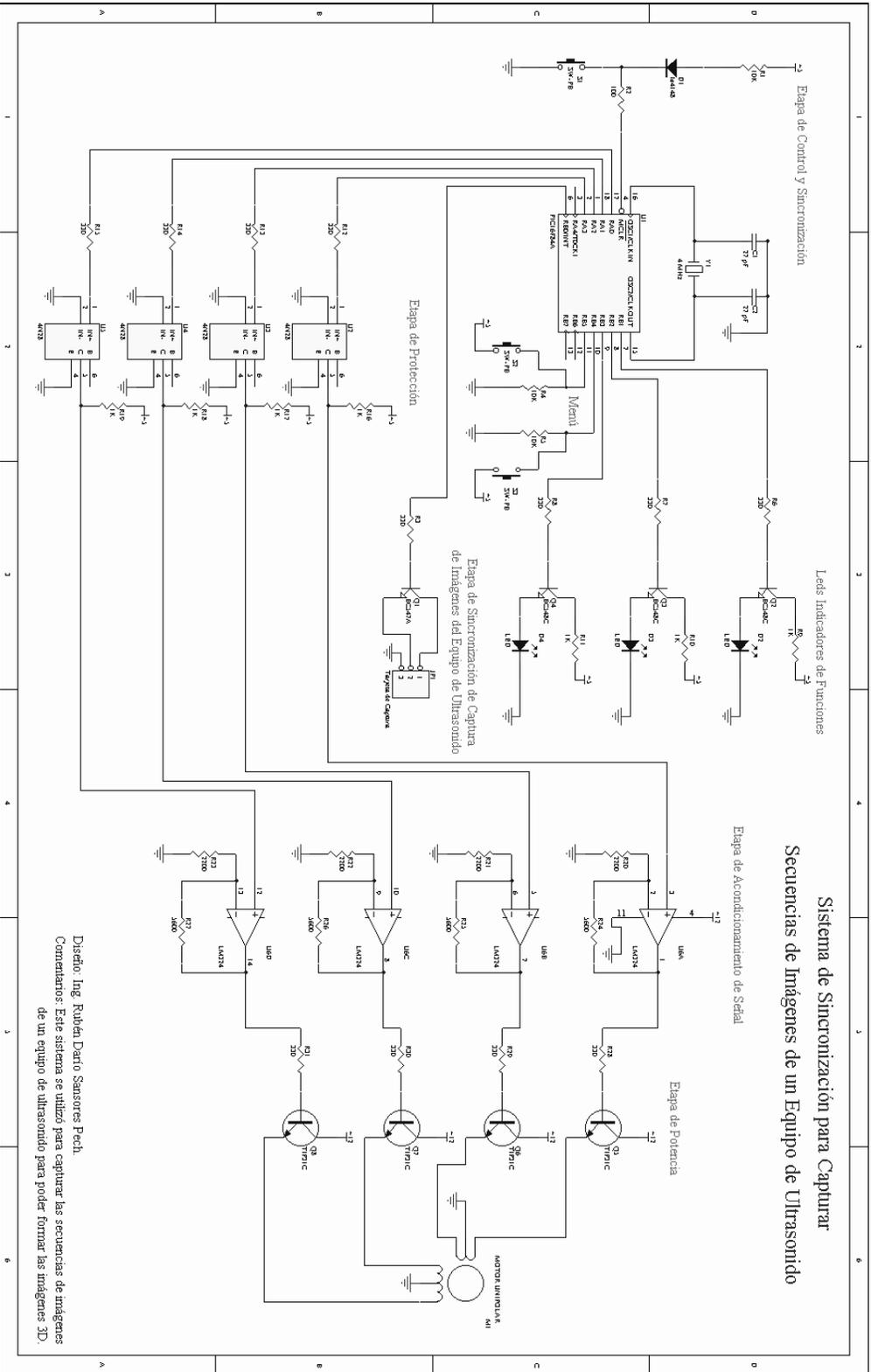
BIBLIOGRAFÍA

- [1] http://news.bbc.co.uk/hi/spanish/science/newsid_2217000/2217276.stm
- [2] Mariano Alcañiz, Vicente Grau, Carlos Monserrat, Carmen Juan, "Sistema de neurocirugía asistida por ordenador mediante computación de altas prestaciones. Desarrollo y validación clínica", Laboratorio de Imagen Médica Computarizada, Universidad Politécnica de Valencia, 1-9, 2001.
- [3] <http://www.conicyt.cl/bases/fondef/PROYECTO/00//D00I1107.HTML>
- [4] Andrew Gee, Richard Prager, Graham Treece, Laurence Berman, "Engineering a freehand 3D ultrasound system", ELSEVIER, Department of Engineering, University of Cambridge, 759-774, 2002.
- [5] Brian C. Porter, Deborah J. Rubens, John G. Strang, Jason Smith, Saara Totterman, and Kevin J. Parker, "Three-Dimensional Registration and Fusion of Ultrasound and MRI Using Major Vessels as Fiducial Markers", IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 20, NO. 4, 354-359, APRIL 2001.
- [6] <http://www.parallelgraphics.com/>
- [7] <http://www.infomed.es/noticias/biblioteca/mayo-2003/diagnosis.asp?id=210&Lista=No>
- [8] LOGIQ α 200, Reference Manual TIP, GE Medical Systems, Training in Partner Ship.
 - [9] http://www.pinnaclesys.com/ListProducts.asp?Division=0&Langua_ID=12
- [10] Microchip, PIC16F8X, 8-Bit CMOS Flash/EEPROM Microcontrollers, www.microchip.com.
- [11] <http://dSPACE.dial.pipex.com/isocom/4n25.htm>
- [12] <http://www.national.com/pf/LM/LM124.html>
- [13] Robert L. Boylestad, Louis Nashelsky, "Electrónica: Teoría de Circuitos", Sexta Edición, PRENTICE HALL, 638-643, 1997.
- [14] <http://www.datasheetcatalog.com/datasheet/T/TIP31C.shtml>
- [15] PM55L-048, <http://www.153korea.co.kr/product/PM55L-048.htm>
<http://www.gmelectronica.com.ar/catalogo/pag65.html>
- [16] DR. JOSÉ NEGRETE HERRERA, "TECNICA DE DISECCIONES Y ATLAS DE ANATOMIA HUMANA", LIBRERÍA DE MEDICINA, DOCEAVA EDICIÓN, 25-163, 1965.
- [17] <http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/>
- [18] <http://www.strangehorizons.com/2001/20011008/steganography.shtml>
- [19] Pitas, A. N. Venetsanopoulos, "Nonlinear Digital Filters: Principles and Applications", Kluwer Academic Publisher, 1990.
- [20] J. Astola and P. Kuosmanen, "Fundamentals of Nonlinear Digital Filtering", CRC Press, Boca Raton-New York, 1997.
- [21] J. B. Bednar, T. L. Watt, "Alpha-trimmed means and their relationship to median filters", IEEE Trans. Acoust., Speech, and Signal Processing, vol. ASSP-32, 145-153, Feb. 1984.
- [22] P. M. Narendra, "A separable median filter for image noise smoothing", IEEE Trans. Pattern Anal. Mach., Intell., vol. PAMI-3, 20-29, Jan. 1981.
- [23] A. Nieminen, Y. Neuvo, "Comments of theoretical analysis of the max/median filter", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-36, 826-827, May 1988.
- [24] R. C. Hardie and C. G. Bonchelet, "LUM filters: a class of rank order based filters for smoothing and sharpening", IEEE Trans. Signal Processing, vol. 41, 1061-1076, March 1993.
- [25] R. J. Crinon, "The Wilcoxon filter: a robust filtering scheme", Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing 85, Tampa, March 1985, 668-671.
- [26] A. Bovik, Handbook of Image and Video Processing, Academic Press, San Diego CA, 2000.
- [27] F. R. Hampel, E. M. Ronchetti, P. J. Rouseew, W. A. Stahel, "Robust Statistics: The approach based on influence function", Wiley, New York, 1986.
- [28] Yan Chen, Ruming Yin, Patrick Flynn, Shira Broschat, "Aggressive region growing for speckle reduction in ultrasound image", ELSEVIER, 677-691, 2003.
- [29] Djamel Boukerroui, Atilla Baskurt, J. Alison Noble, Olivier Basset, "Segmentation of ultrasound images-multiresolution 2D and 3D algorithm based on global and local statistics", ELSEVIER, 779-790, 2003.

- [30] J. S. Lee, "Digital image enhancement and noise filtering by use of local statistic", IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-2, 165-168, March 1980.
- [31] Volodymyr Ponomaryov, Francisco Gallegos Funes, "Real-Time Processing Scheme Based on RM Estimators", Electronic Imaging, SPIE Vol. 5012, 37-48, 2003.
- [32] Volodymyr Ponomaryov, Francisco Gallegos Funes, "Real-time image filtering with retention of small-size details and complex noise mixture", Real-Time Imaging VI, SPIE Vol. 4666, 30-41, 2003.
- [33] Rubén Sansores Pech, Volodymyr Ponomaryov, "Algoritmos de Filtrado con Supresión de Ruido y Preservación de Detalles en Imágenes de Tres Dimensiones", 9a Conferencia de Ingeniería Eléctrica, Septiembre del 2003.
- [34] Rubén Sansores Pech, Volodymyr Ponomaryov, "SISTEMA PARA ADQUIRIR IMÁGENES DE ULTRASONIDO EN 3D", SOMI XVIII Congreso de Instrumentación, Octubre del 2003.
- [35] Rubén Sansores Pech, Volodymyr Ponomaryov, "INVESTIGACIÓN E IMPLEMENTACIÓN DE LOS ALGORITMOS DE ESTADÍSTICA DE ORDEN PARA FILTRAR LAS IMÁGENES 3D", 7° Congreso Nacional de Ingeniería Electromecánica y de Sistemas, Noviembre del 2003.
- [36] Raj Shekhar y Vladimir Zagrodsky, "Mutual Information-Based Rigid and Nonrigid Registration of Ultrasound Volumes", IEEE Transactions on Medical Imaging, Vol. 21, No. 1, 9-22, enero 2002.
- [37] X. Pennec, P. Cachier, N. Ayache, "Tracking brain deformations in time sequences of 3D US images",.
- [38] TMS320C54x DSP, Reference Set Volume 1: CPU and Peripherals Literature Number: SPRU131F, Texas Instruments, April 1999.
- [39] Texas Instruments, TMS320C6711, TMS320C6711B, TMS320C6711C Floating-Point Digital Signal Processors, SPRS088H, Revised 2003.
- [40] Laboratorio de FPGAs y Cómputo Móvil, Reporte anual de actividades 2002, Ciencias Computacionales INAOE.
- [41] Texas Instruments, TMS320C6000 Optimizing Compiler User's Guide, SPRU187I, Texas Instruments Incorporated, 2001.

ANEXO A

En esta sección se presenta el diagrama esquemático del sistema para capturar imágenes 3D presentado y descrito en el capítulo III.



Diseño: Ing. Rubén Darío Saizores Pech.
 Comentarios: Este sistema se utilizó para capturar las secuencias de imágenes de un equipo de ultrasonido para poder formar las imágenes 3D.

ANEXO B

Se presentan las imágenes de los planos XY, YZ y XZ de las simulaciones y restauraciones descritas en el capítulo V.

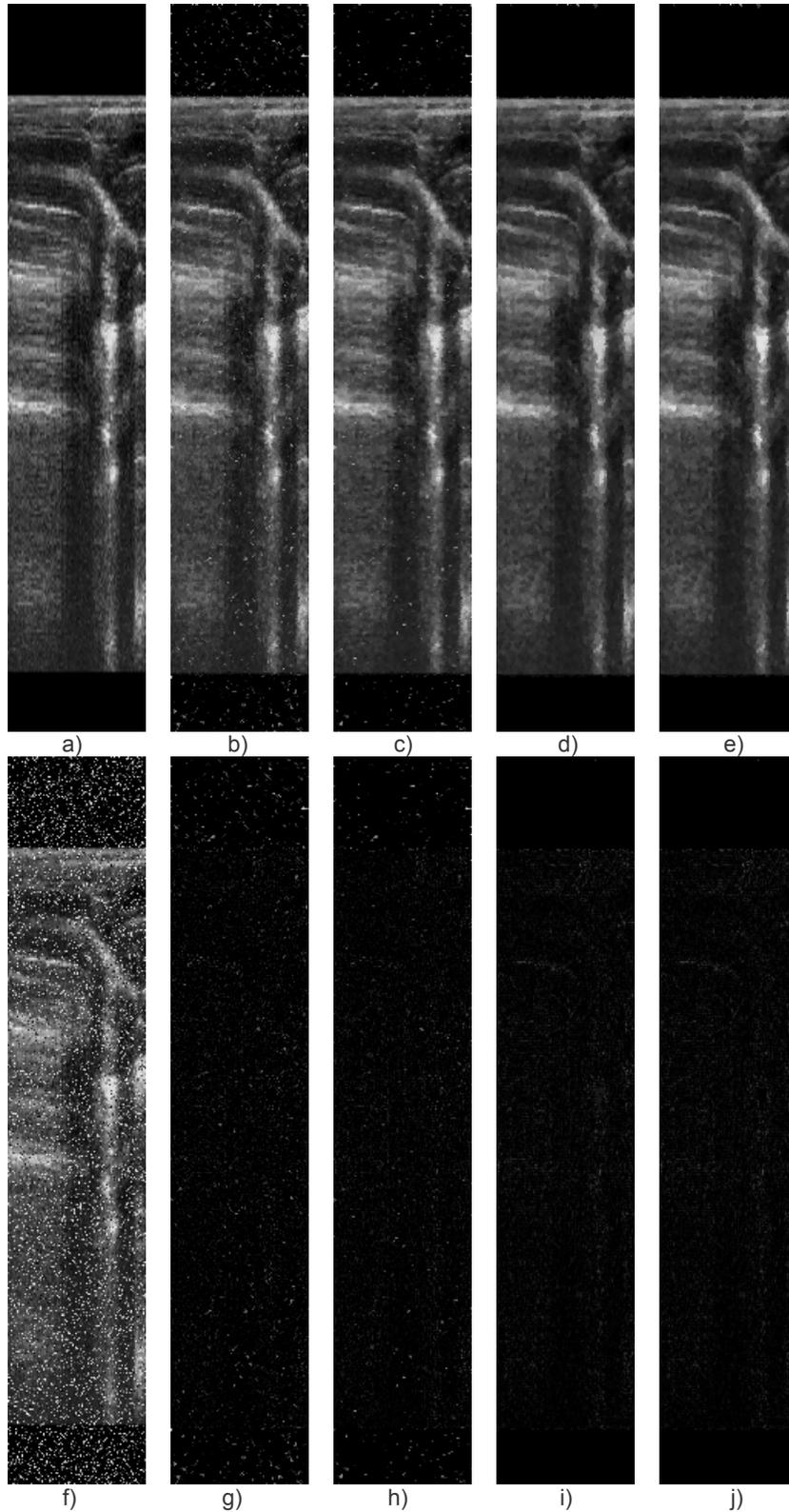


Figura B.1. a) imagen original; b) imagen restaurada con el filtro LUM Smooth; c) imagen restaurada con el filtro MSM5; d) imagen restaurada con el filtro MMKNN con F.I. Hampel; e) imagen restaurada con el filtro MMKNN con F.I. Cut; f) imagen degradada con 20% de ruido impulsivo; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano XZ.

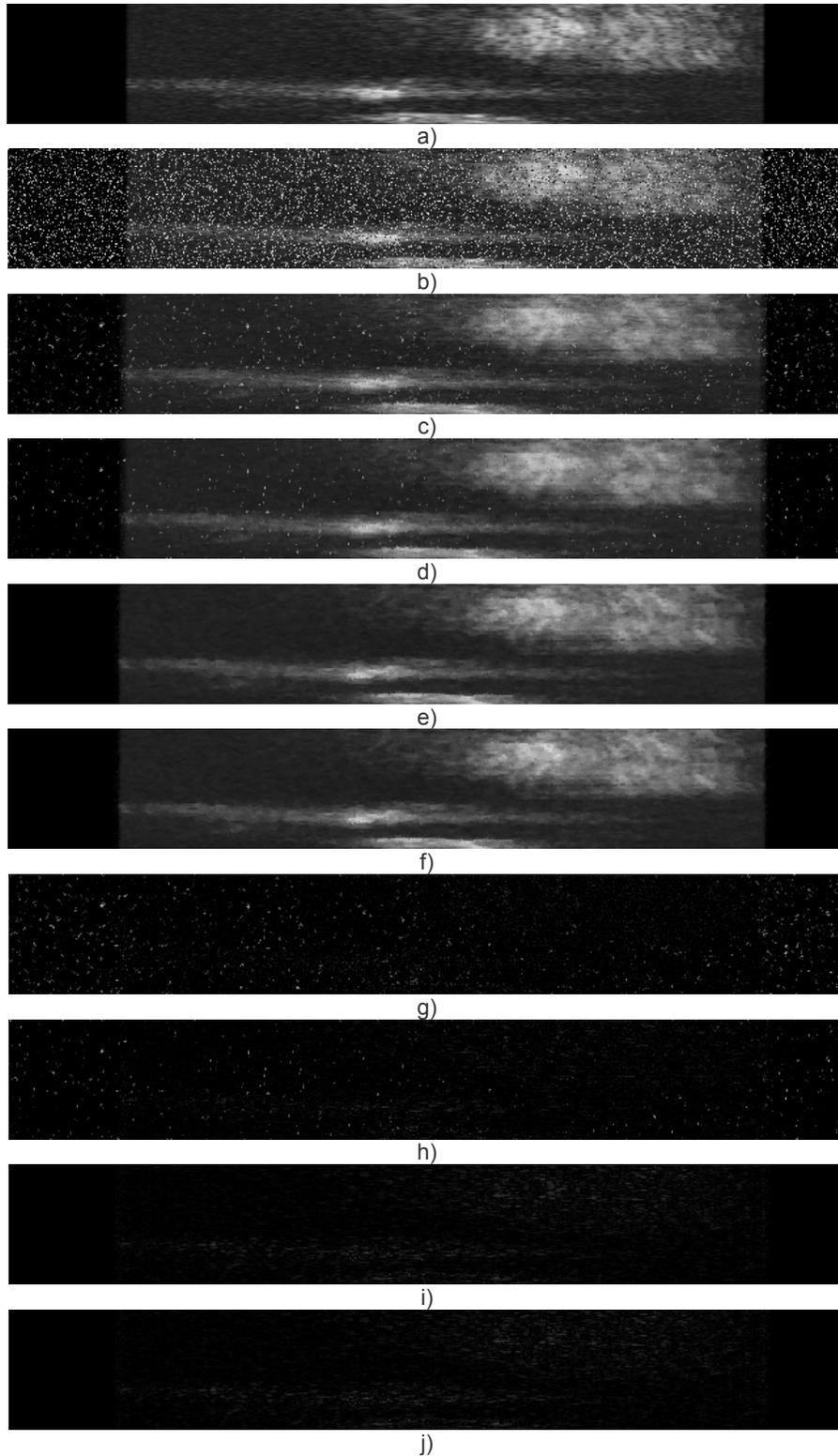


Figura B.2. a) imagen original; b) imagen degradada con 20% de ruido impulsivo; c) imagen restaurada con el filtro LUM Smooth; d) imagen restaurada con el filtro MSM5; e) imagen restaurada con el filtro MMKNN con F.I. Hampel; f) imagen restaurada con el filtro MMKNN con F.I. Cut; g) imagen de error del filtro LUM Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano YZ.

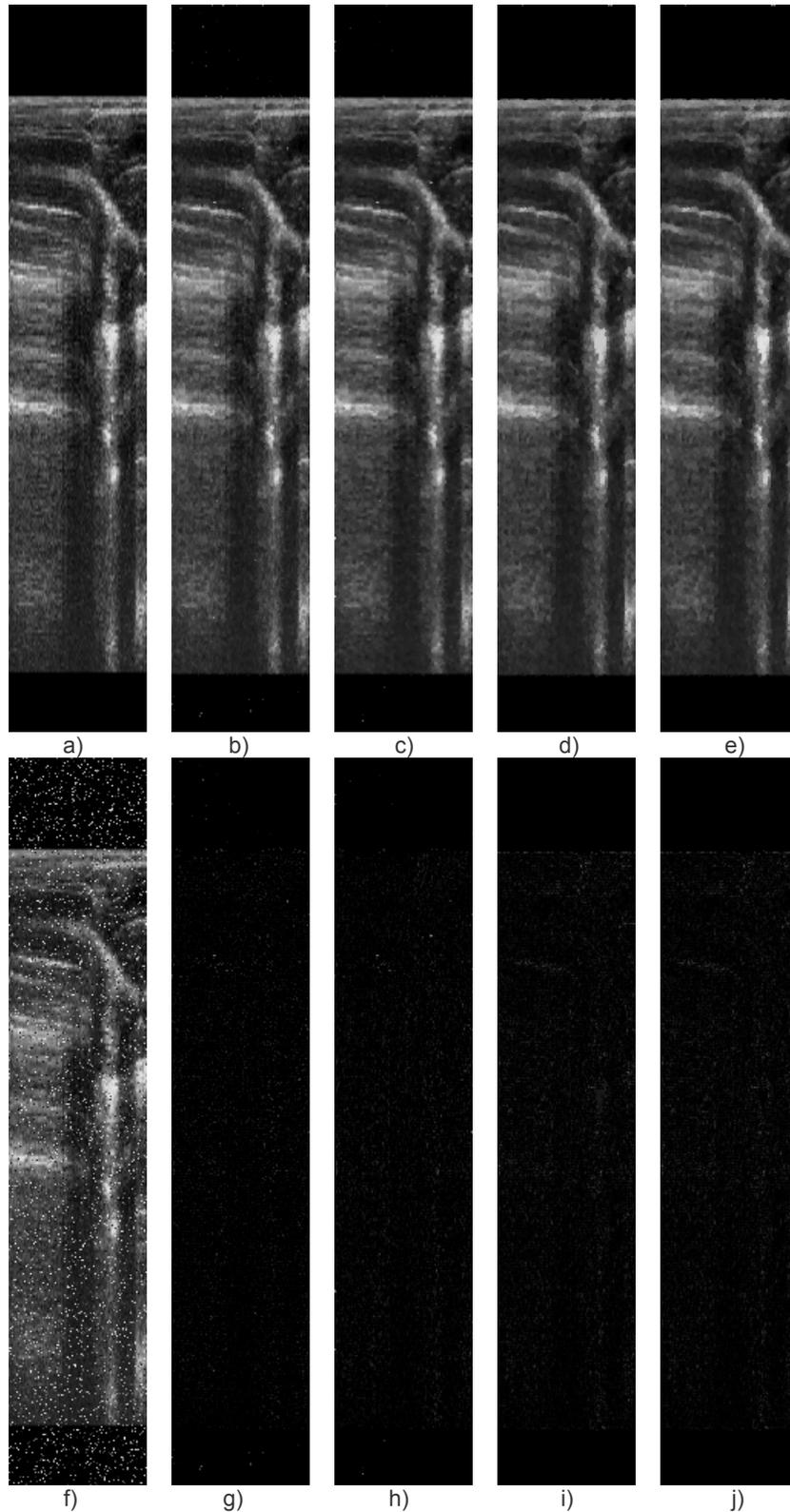


Figura B.3. a) imagen original; b) imagen restaurada con el filtro LUM Smooth; c) imagen restaurada con el filtro MSM5; d) imagen restaurada con el filtro MMKNN con F.I. Hampel; e) imagen restaurada con el filtro MMKNN con F.I. Cut; f) imagen degradada con 10% de ruido impulsivo; g) imagen de error del filtro LUM

Smooth; h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano XZ.

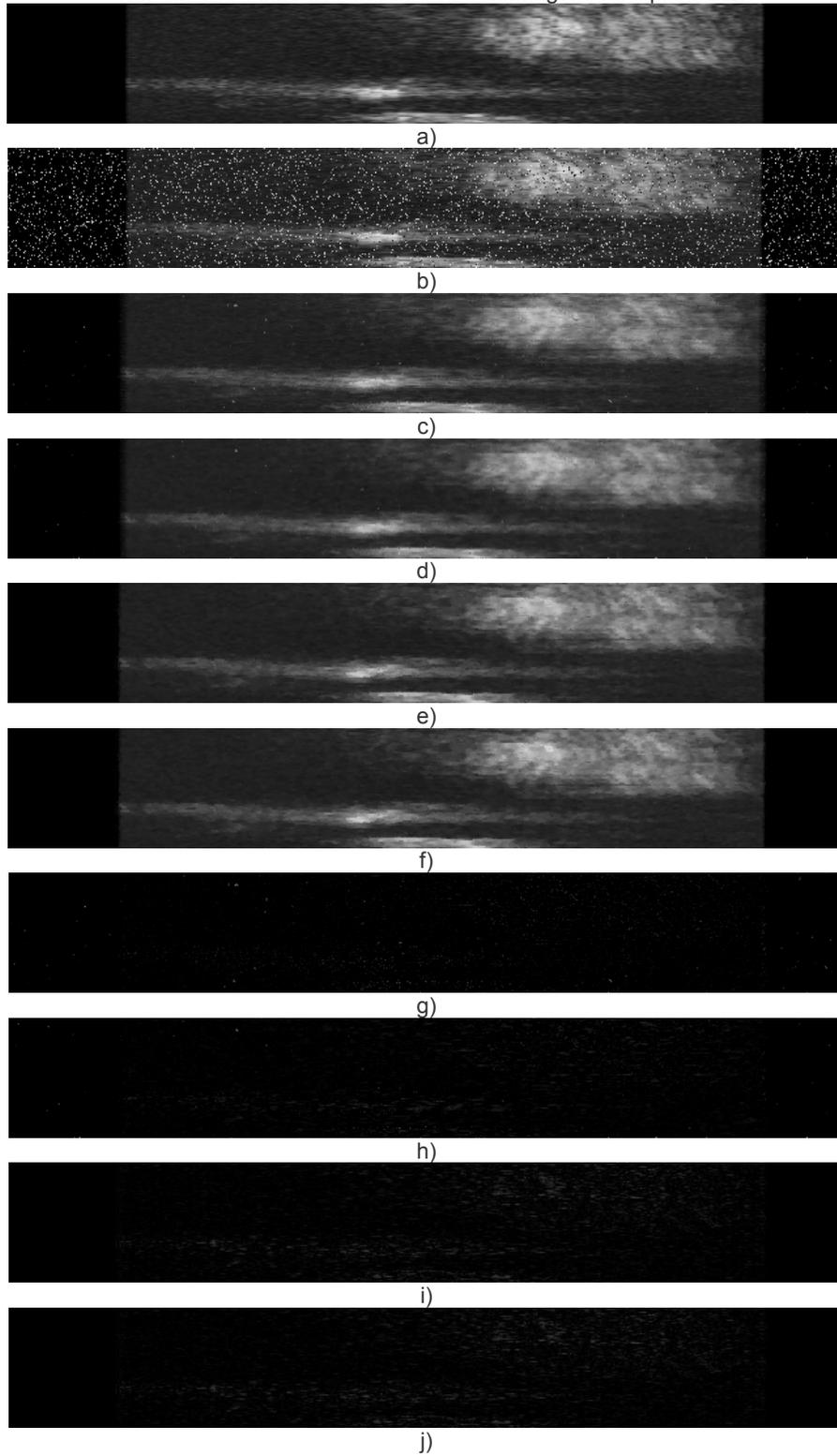


Figura B.4. a) imagen original; b) imagen degradada con 10% de ruido impulsivo; c) imagen restaurada con el filtro LUM Smooth; d) imagen restaurada con el filtro MSM5; e) imagen restaurada con el filtro MMKNN con F.I. Hampel; f) imagen restaurada con el filtro MMKNN con F.I. Cut; g) imagen de error del filtro LUM Smooth;

h) imagen de error del filtro MSM5; i) imagen de error del filtro MMKNN con F.I. Hampel; j) imagen de error del filtro MMKNN con F.I. Cut. Son imágenes del plano YZ.

Imágenes del Músculo del Antebrazo (Restauración)

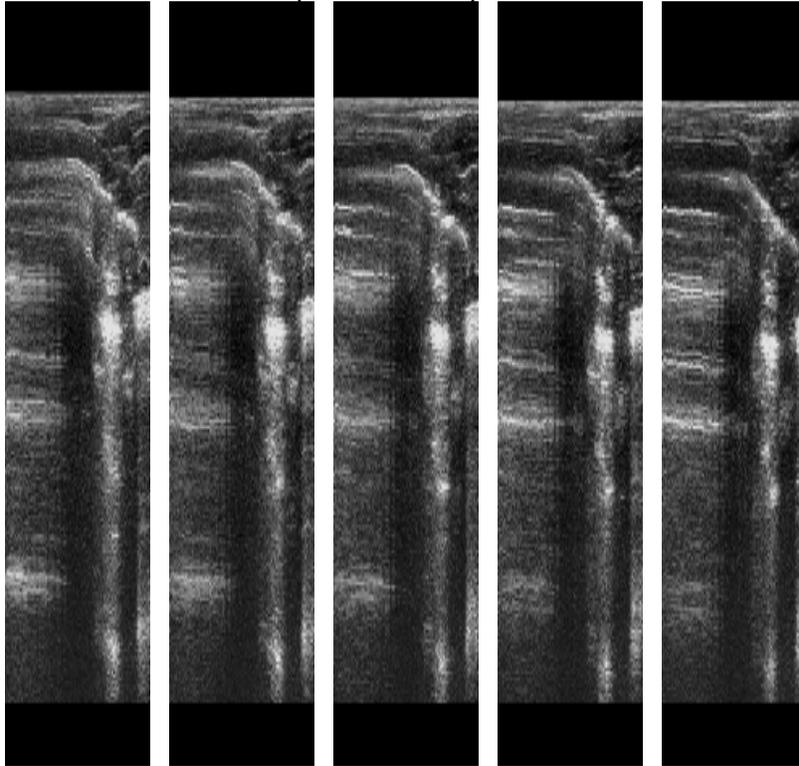


Figura B.5. Imágenes originales, cortes del plano XZ.

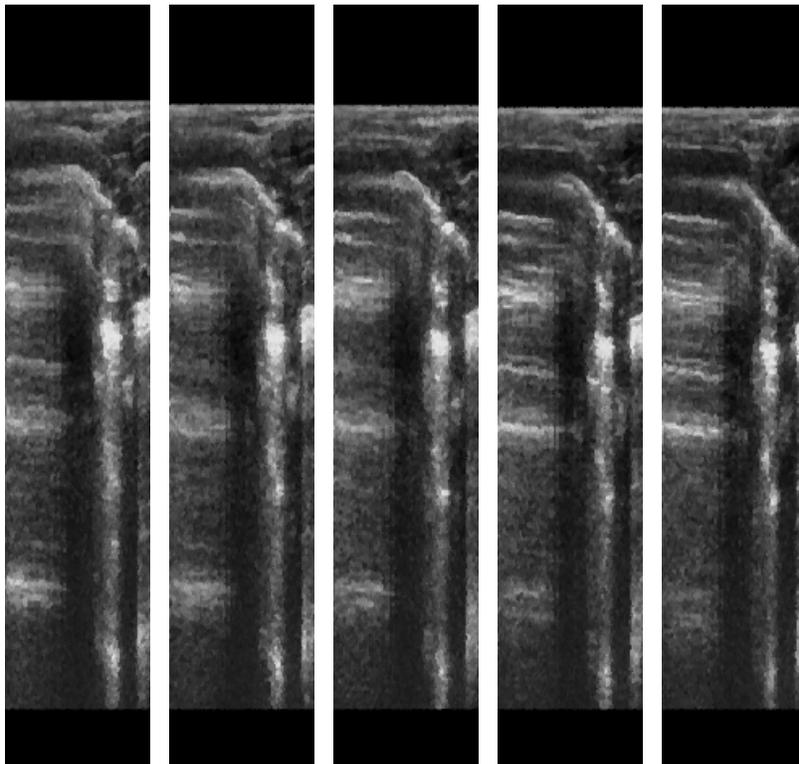


Figura B.6. Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XZ.

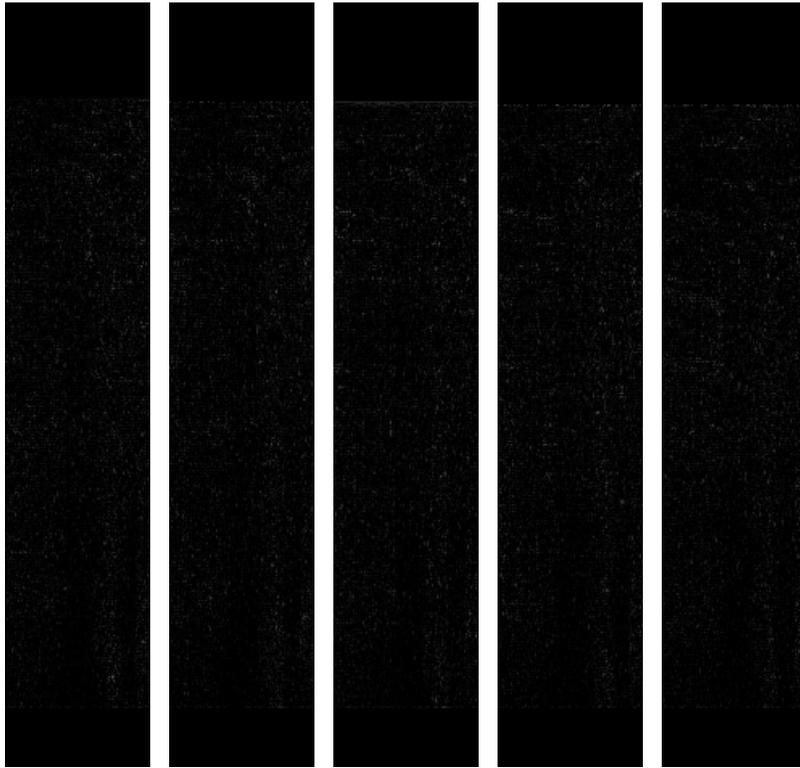


Figura B.7. Imágenes de error, cortes del plano XZ.

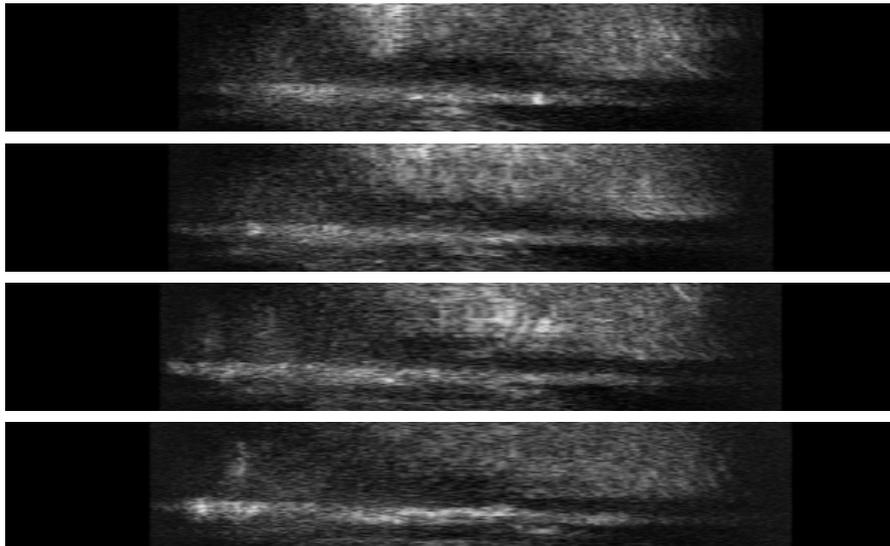


Figura B.8. Imágenes originales, cortes del plano YZ.

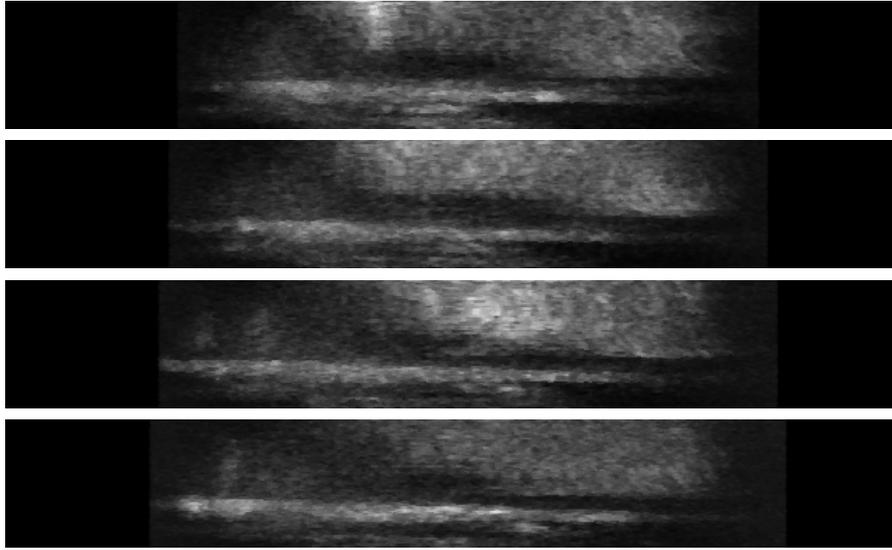


Figura B.9. Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano YZ.

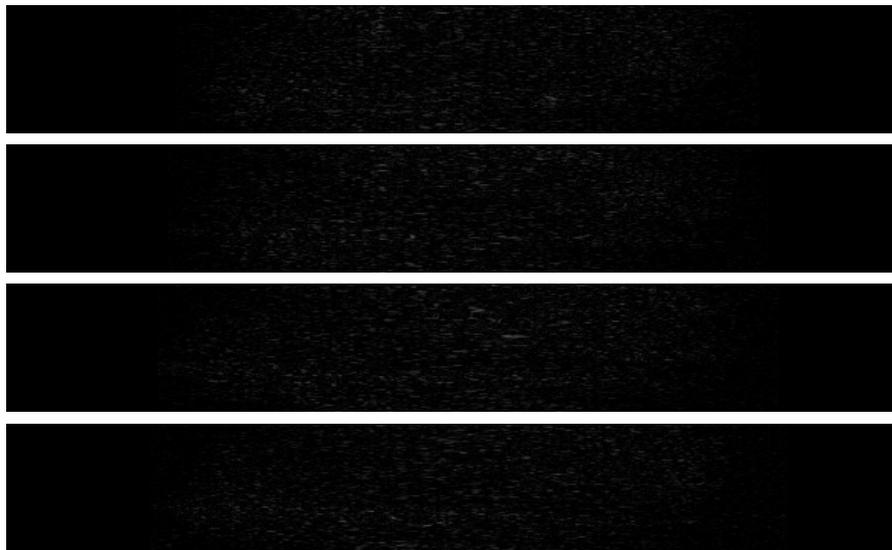


Figura B.10. Imágenes de error, cortes del plano YZ.

Imágenes del Hígado y Riñón (Restauración)

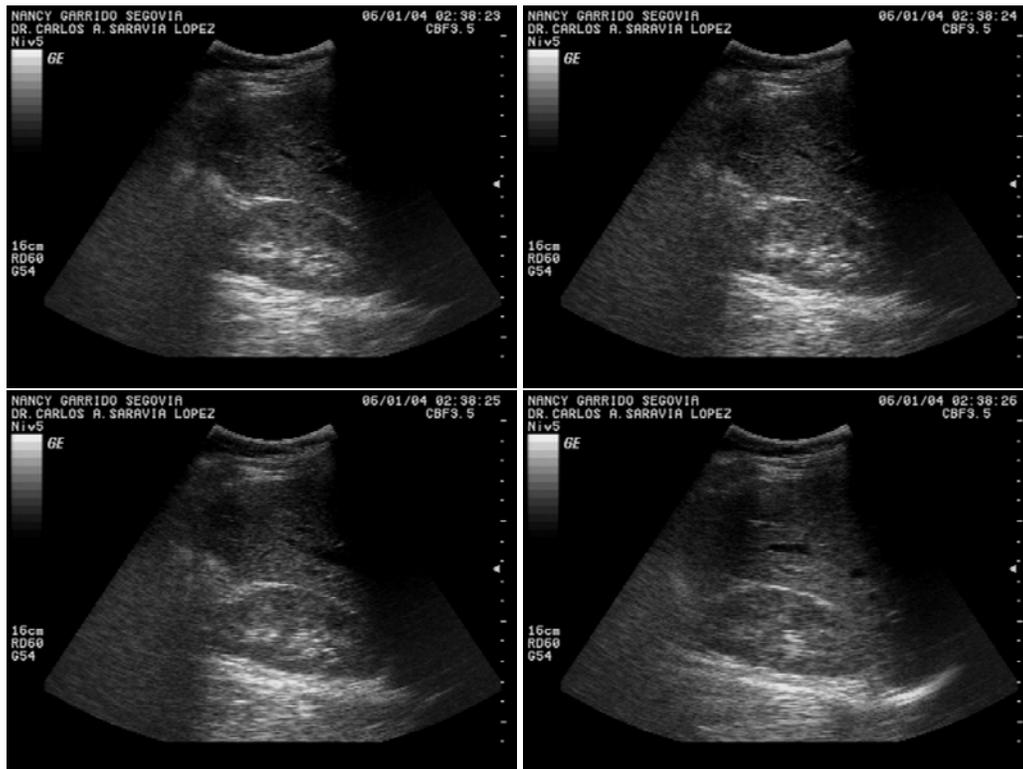


Figura B.11. Imágenes originales, cortes del plano XY.

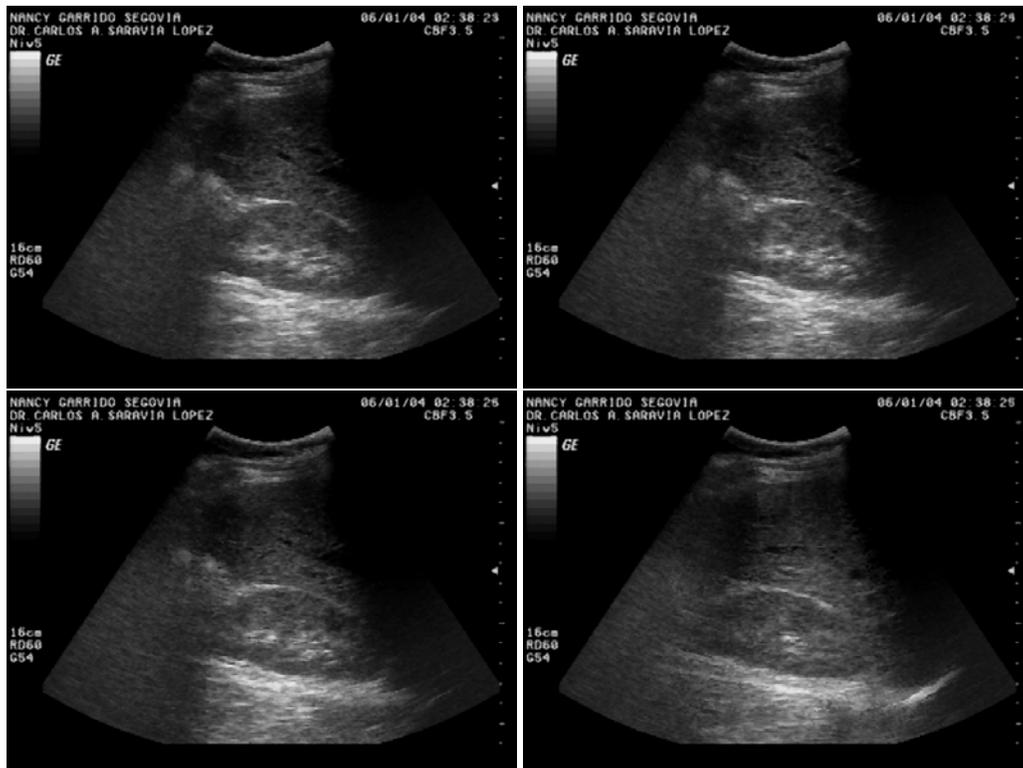


Figura B.12. Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XY.

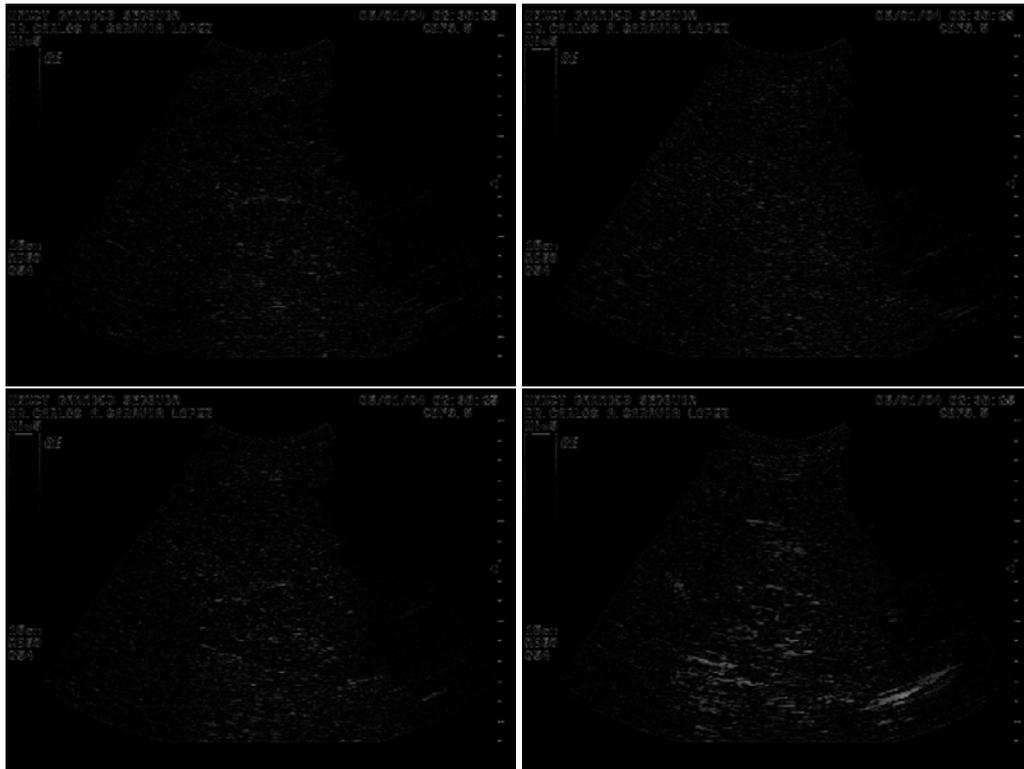


Figura B.13. Imágenes de error, cortes del plano XY.

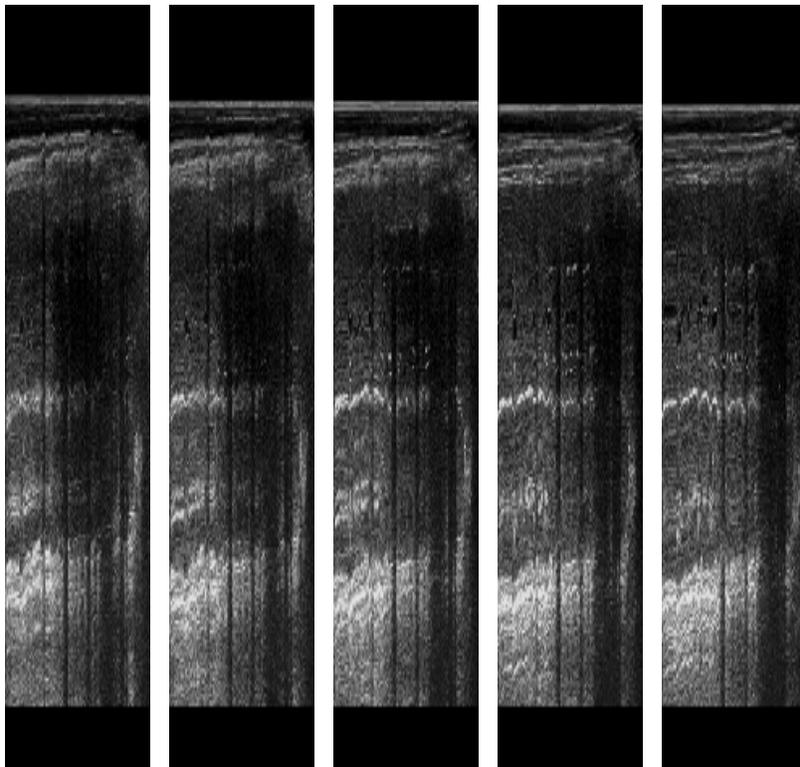


Figura B.14. Imágenes originales, cortes del plano XZ.

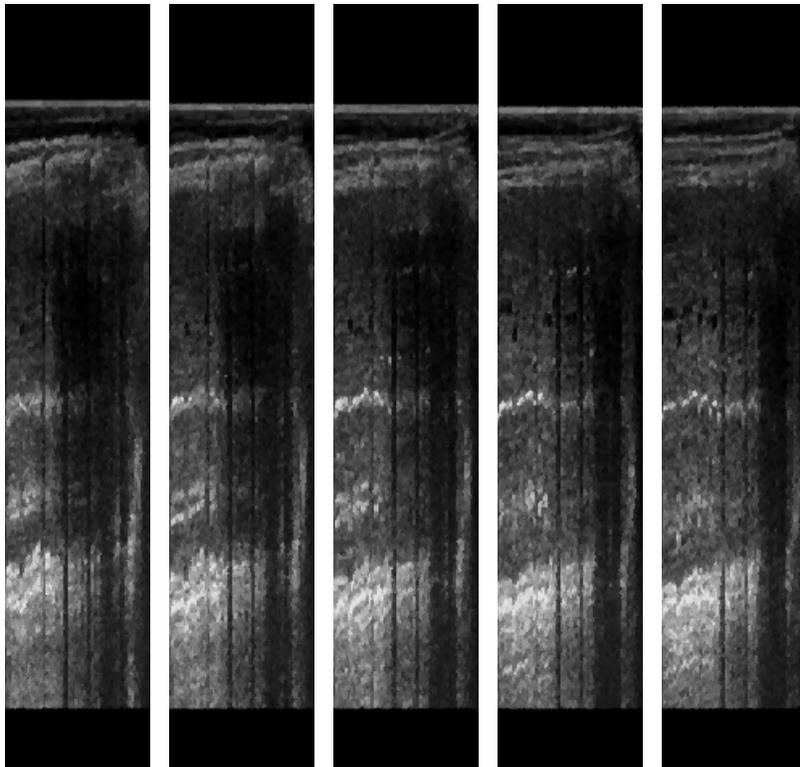


Figura B.15. Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano XZ.

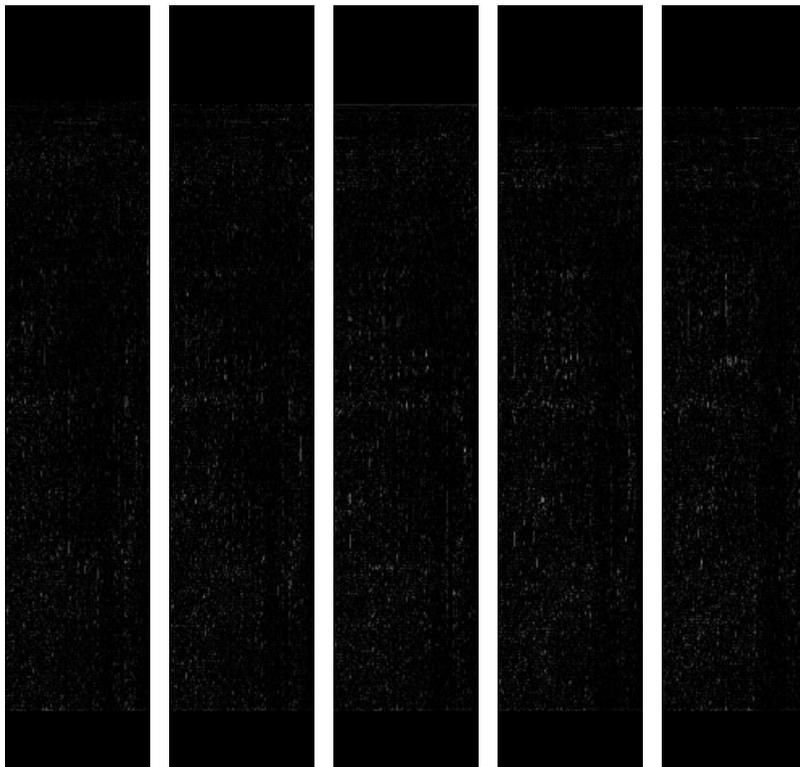


Figura B.16. Imágenes de error, cortes del plano XZ.

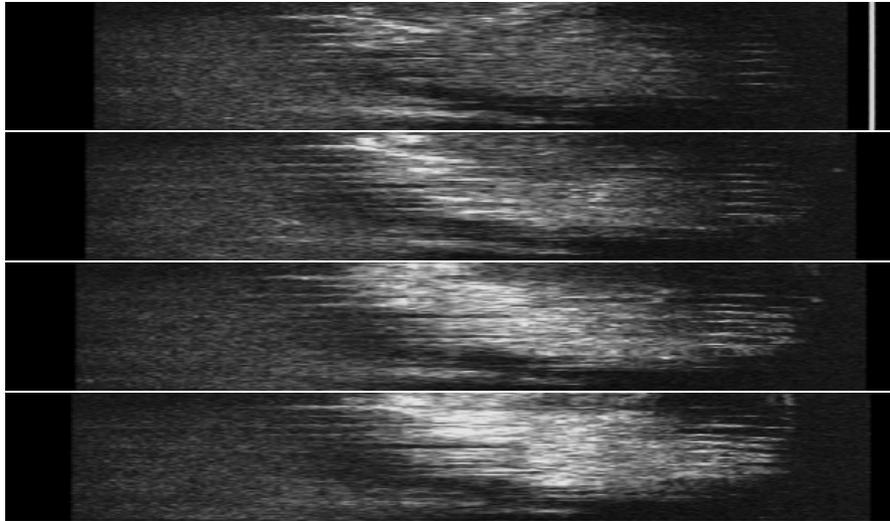


Figura B.17. Imágenes originales, cortes del plano YZ.

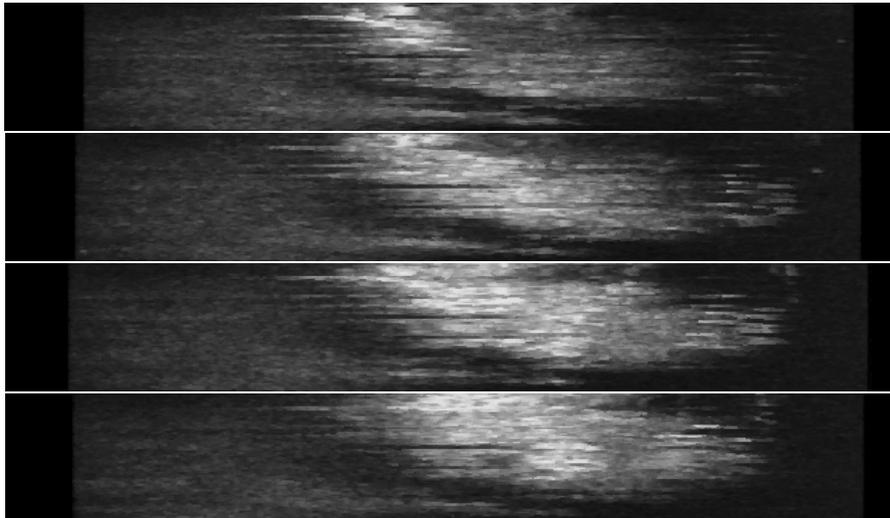


Figura B.18. Imágenes restauradas con el filtro MMKNN con F.I. Cut, cortes del plano YZ.

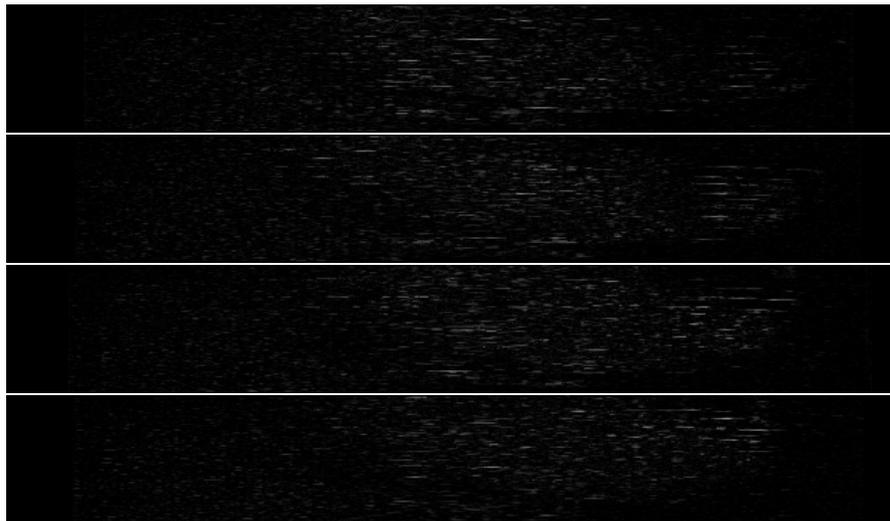


Figura B.19. Imágenes de error, cortes del plano YZ.

ANEXO C

Los programas presentados en esta sección son de los mejores filtros 3D que se describieron en el capítulo V.

Filtro LUM Sharp

```
*****
//                               FILTRO LUM SHARP
//                               Cubo de barrido 3x3x3=27 voxels
*****

iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
  cambios=0;
  numdatos=numdatos-1;
  iteraciones=iteraciones-1;
  for(i=0; i<=numdatos; i++)
  {
    if(cubo333[i]>cubo333[i+1])
    {
      guar=cubo333[i+1];
      cubo333[i+1]=cubo333[i];
      cubo333[i]=guar;
      cambios=cambios+1;
    }
  }
  if(iteraciones<=0)
  {
    salir=0;
  }
  if(cambios<=0)
  {
    salir=0;
  }
}
while(salir!=0);

x=cubo333[13];
xt=cubo333[6];
xnt1=cubo333[20];

tl=(cubo333[6]+cubo333[20])/2.0;

if((xt<x)&&(x<=tl))
{
  resul=xt;
}
else if((tl<x)&&(x<=xnt1))
{
  resul=xnt1;
}
else
{
  resul=x;
}
}
```

Filtro MSM5

```
*****
//                               FILTRO MSM5
//                               Cubo de barrido 3x3x3=27 voxels
*****

voxelstotal=8;
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
  cambios=0;
  numdatos=numdatos-1;
  iteraciones=iteraciones-1;
  for(i=0; i<=numdatos; i++)
```

```

    {
        if(h[i]>h[i+1])
        {
            guar=h[i+1];
            h[i+1]=h[i];
            h[i]=guar;
            cambios=cambios+1;
        }
    }
    if(iteraciones<=0)
    {
        salir=0;
    }
    if(cambios<=0)
    {
        salir=0;
    }
}
while(salir!=0);
m1[0]=h[4];

voxelstotal=8;
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(v[i]>v[i+1])
        {
            guar=v[i+1];
            v[i+1]=v[i];
            v[i]=guar;
            cambios=cambios+1;
        }
    }
    if(iteraciones<=0)
    {
        salir=0;
    }
    if(cambios<=0)
    {
        salir=0;
    }
}
while(salir!=0);
m1[1]=v[4];

m1[2]=cubo333[13];

voxelstotal=2;
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(m1[i]>m1[i+1])
        {
            guar=m1[i+1];
            m1[i+1]=m1[i];
            m1[i]=guar;
        }
    }
}
while(salir!=0);
m1[3]=m1[2];

```

```

        cambios=cambios+1;
    }
}
if(iteraciones<=0)
{
    salir=0;
}
if(cambios<=0)
{
    salir=0;
}
}
while(salir!=0);
m5[0]=m1[1];
//*****
//*****
//*****
voxelstotal=8;
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(d45[i]>d45[i+1])
        {
            guar=d45[i+1];
            d45[i+1]=d45[i];
            d45[i]=guar;
            cambios=cambios+1;
        }
    }
}
if(iteraciones<=0)
{
    salir=0;
}
if(cambios<=0)
{
    salir=0;
}
}
while(salir!=0);
m2[0]=d45[4];

voxelstotal=8;
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(d135[i]>d135[i+1])
        {
            guar=d135[i+1];
            d135[i+1]=d135[i];
            d135[i]=guar;
            cambios=cambios+1;
        }
    }
}
if(iteraciones<=0)
{
    salir=0;
}

```

```

    }
    if(cambios<=0)
    {
        salir=0;
    }
}
while(salir!=0);
m2[1]=d135[4];

m2[2]=cubo333[13];

voxelstotal=2;
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(m2[i]>m2[i+1])
        {
            guar=m2[i+1];
            m2[i+1]=m2[i];
            m2[i]=guar;
            cambios=cambios+1;
        }
    }
}
if(iteraciones<=0)
{
    salir=0;
}
if(cambios<=0)
{
    salir=0;
}
}
while(salir!=0);
m5[1]=m2[1];
//*****
//*****
//*****
m5[2]=cubo333[13];

voxelstotal=2;
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(m5[i]>m5[i+1])
        {
            guar=m5[i+1];
            m5[i+1]=m5[i];
            m5[i]=guar;
            cambios=cambios+1;
        }
    }
}
if(iteraciones<=0)
{
    salir=0;
}
if(cambios<=0)

```

```

    {
        salir=0;
    }
}
while(salir!=0);
resul=m5[1];

```

Filtro MMKNN CON F.I. CUT

```

//*****
//
//          F I L T R O   M M K N N
//          Cubo de barrido 3x3x3=27 voxels
//*****
//          med{[x(i,j)-x(i+m,j+n)]}
//*****
i=0;
for(j=0; j<=voxelstotal; j++)
{
    if(j!=nvc)
    {
        vec1[i]=voxelcentral-cubo333[j];
        if(vec1[i]<0)
        {
            vec1[i]=-1*vec1[i];
        }
        vec2[i]=cubo333[j];
        i=i+1;
    }
}
iteraciones=voxelstotal-1;
numdatos=voxelstotal-1;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(vec1[i]>vec1[i+1])
        {
            guar=vec1[i+1];
            vec1[i+1]=vec1[i];
            vec1[i]=guar;
            cambios=cambios+1;
        }
    }
    if(iteraciones<=0)
    {
        salir=0;
    }
    if(cambios<=0)
    {
        salir=0;
    }
}
while(salir!=0);
med1=(vec1[12]+vec1[13])/2.0;
//*****
//          med{x(i+k,j+l)}
//*****
iteraciones=voxelstotal-1;
numdatos=voxelstotal-1;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;

```

```

for(i=0; i<=numdatos; i++)
{
    if(vec2[i]>vec2[i+1])
    {
        guar=vec2[i+1];
        vec2[i+1]=vec2[i];
        vec2[i]=guar;
        cambios=cambios+1;
    }
}
if(iteraciones<=0)
{
    salir=0;
}
if(cambios<=0)
{
    salir=0;
}
}
while(salir!=0);
med2=(vec2[12]+vec2[13])/2.0;
//*****
//                               MAD{x(i,j)}
//*****
i=0;
for(j=0; j<=voxelstotal; j++)
{
    if(j!=nvc)
    {
        vec3[i]=med1-cubo333[j];
        if(vec3[i]<0)
        {
            vec3[i]=-1*vec3[i];
        }
        i=i+1;
    }
}
iteraciones=voxelstotal-1;
numdatos=voxelstotal-1;
salir=1;
do
{
    cambios=0;
    numdatos=numdatos-1;
    iteraciones=iteraciones-1;
    for(i=0; i<=numdatos; i++)
    {
        if(vec3[i]>vec3[i+1])
        {
            guar=vec3[i+1];
            vec3[i+1]=vec3[i];
            vec3[i]=guar;
            cambios=cambios+1;
        }
    }
    if(iteraciones<=0)
    {
        salir=0;
    }
    if(cambios<=0)
    {
        salir=0;
    }
}
while(salir!=0);
MAD=(vec3[12]+vec3[13])/2.0;
//*****
//                               S(x(i,j)) y Kc((i,j))
//*****
if((MAD==0)||med2==0)

```

```

    {
    S=0;
    kc=351;
    }
else
    {
    S=(med1/MAD)+0.5*(MAD/med2);
    kc=5+8.0*S;
    }
}
//*****
//                               Kc((i,j))
//*****
//                               kc>350
//*****
if(kc>350)
{
iteraciones=74;
numdatos=74;
salir=1;
do
{
cambios=0;
numdatos=numdatos-1;
iteraciones=iteraciones-1;
for(i=0; i<=numdatos; i++)
{
if(cubo553[i]>cubo553[i+1])
{
guar=cubo553[i+1];
cubo553[i+1]=cubo553[i];
cubo553[i]=guar;
cambios=cambios+1;
}
}
}
if(iteraciones<=0)
{
salir=0;
}
if(cambios<=0)
{
salir=0;
}
}
while(salir!=0);
resul=cubo553[37];
}
//*****
//                               kc>24 & kc<=350
//*****
if((kc>24)&&(kc<=350))
{
iteraciones=voxelstotal;
numdatos=voxelstotal;
salir=1;
do
{
cambios=0;
numdatos=numdatos-1;
iteraciones=iteraciones-1;
for(i=0; i<=numdatos; i++)
{
if(cubo333[i]>cubo333[i+1])
{
guar=cubo333[i+1];
cubo333[i+1]=cubo333[i];
cubo333[i]=guar;
cambios=cambios+1;
}
}
}
if(iteraciones<=0)

```

```

        {
            salir=0;
        }
        if(cambios<=0)
        {
            salir=0;
        }
    }
    while(salir!=0);
    resul=cubo333[13];
}
//*****
//                               kc<=24
//*****
    if(kc<=24)
    {
        resul=voxelcentral;
        do
        {
            voxelcentral=resul;
//*****
//                               voxelcentral - cubo 3x3x3
//*****
            i=0;
            for(j=0; j<=voxelstotal; j++)
            {
                if(j!=nvc)
                {
                    vec1[i]=voxelcentral-cubo333[j];
                    if(vec1[i]<0)
                    {
                        vec1[i]=-1*vec1[i];
                    }
                    vechold[i]=cubo333[j];
                    i=i+1;
                }
            }
//*****
//                               Ordenar vec1
//*****
            for(j=0; j<=24; j++)
            {
                for(i=0; i<=24; i++)
                {
                    if(vec1[i]>vec1[i+1])
                    {
                        guar=vec1[i+1];
                        vec1[i+1]=vec1[i];
                        vec1[i]=guar;

                        guar1=vechold[i+1];
                        vechold[i+1]=vechold[i];
                        vechold[i]=guar1;
                    }
                }
            }
//*****
//                               Funcion de Influencia
//*****
            i=0;
            for(j=0; j<kc; j++)
            {
                if(vechold[j]<=255)
                {
                    vec3[i]=vechold[j];
                    i=i+1;
                }
            }
//*****
//                               Resultado final

```

```

//*****
numdatos=i-1;
contfinal=i;
for(j=0; j<numdatos; j++)
{
    for(i=0; i<numdatos; i++)
    {
        if(vec3[j]>vec3[i+1])
        {
            guar=vec3[i+1];
            vec3[i+1]=vec3[j];
            vec3[j]=guar;
        }
    }
}
residuo=contfinal&0x01;
if(residuo==0)
{
    resul=(vec3[contfinal/2.0]+vec3[(contfinal/2.0)-1])/2.0;
}
else
{
    resul=vec3[(contfinal-1)/2.0];
}
}
while(resul!=voxelcentral);
}

```

ANEXO D

En esta sección se presentan los artículos publicados en diferentes congresos.



Algoritmos de Filtrado con Supresión de Ruido y Preservación de Detalles en Imágenes de Tres Dimensiones

Rubén Sansores Pech^a Volodymyr I. Ponomaryov^b

^aInstituto Politécnico Nacional, ESIME, U.P. Zacatenco, Unidad Profesional Adolfo López Mateos Av. Instituto Politécnico Nacional s/n.

^bInstituto Politécnico Nacional, ESIME, U.P. Culhuacán, Av. Santa Ana # 1000, Col. San Francisco Culhuacán, C.P. 04400, México D.F.

Resumen

Se presenta el sistema que se utilizó, para capturar las imágenes en tres dimensiones; y la implementación del filtro tridimensional de orden modificado, para suprimir ruido y preservar los detalles finos de objetos. Se demostró el desempeño del procesamiento de imágenes en 3D, con base en criterios PSNR y MAE, usando simulaciones con ruido impulsivo y ruido gaussiano.

1. INTRODUCCIÓN

Los sistemas y equipos, tanto médicos como de telecomunicaciones, se están diseñando para emitir y desplegar información en imágenes de tres dimensiones (3D), como por ejemplo: los rayos x, el neuroavagador, la tele radiología, la resonancia magnética, el ultrasonido, los sistemas de navegación en 3D, o de percepción remota con radar, los sistemas meteorológicos, etc. La posibilidad de procesar imágenes (campos) en 3D, nos conduce a una nueva aplicación; donde es necesario mejorar la calidad de objetos 3D, dentro de la imagen, con supresión de ruido que siempre afecta al proceso de comunicación o adquisición de campos. Otro problema que no es trivial; es la modificación (adaptación) e implementación de filtros existentes e investigados en literatura en procesamiento bidimensional (2D) a procesamiento en 3D.

Hay muchos filtros no lineales, la mayoría de ellos son de estadística de orden, que se han implementado en imágenes de dos dimensiones, utilizados e investigados [1,2]. La técnica tradicional se maneja con el uso de una ventana de m por n píxeles, la cual recorre, píxel por píxel de toda la imagen.

En este artículo, se presenta el desarrollo y los resultados obtenidos de las pruebas de los filtros no lineales en imágenes 3D, en el cual se utiliza un cubo de n voxels, el cual recorre voxel por voxel de toda la imagen. Los criterios tradicionales usados para presentar el rendimiento de los filtros no lineales, fueron: la relación pico de señal a ruido

(PSNR), que consideriza la supresión de ruido, y el error absoluto medio (MAE), para evaluar la preservación de contornos y detalles finos.

A continuación, en el Capítulo I se describe el sistema para capturar imágenes de tres dimensiones, en el Capítulo II; la adaptación de los filtros no lineales en tres dimensiones, en el Capítulo III; la simulación y los resultados obtenidos, y en el Capítulo V; las conclusiones.

II. SISTEMA PARA ADQUIRIR IMÁGENES DE TRES DIMENSIONES

Basándose en los artículos publicados [3,4], se tomó la idea de cómo adquirir las imágenes de tres dimensiones.

El sistema descrito a continuación, es un sistema experimental; el cual nos permitió introducirnos al nuevo campo de aplicación de las imágenes de tres dimensiones, y llegar a nuestro objetivo principal; la implementación de los filtros no lineales en este tipo de imágenes.

En este sistema, tenemos un grado de libertad, del control del movimiento del transductor; el cual nos permite realizar exploraciones en línea recta, haciendo cortes transversales y longitudinales del órgano humano.

El sistema consiste, en desplazar el transductor del ultrasonido en un carril. Al ir desplazándose el transductor por el carril, va adquiriendo imágenes bidimensionales del órgano explorado, las cuales se almacenan en una computadora, para después unir las n imágenes bidimensionales, y formar la matriz de tres dimensiones.



SOMI XVII Congreso de Instrumentación
Ingeniería Biomédica RSP189

SISTEMA PARA ADQUIRIR IMÁGENES DE ULTRASONIDO EN 3D

R. D. Sansores Pech¹, V. I. Ponomarev²

¹Instituto Politécnico Nacional, ESIME, U.P. Zacatlan, Unidad Profesional Adolfo López
Mateos Av. Instituto Politécnico Nacional s/n.

E-mail: dario_sansores@hotmail.com

²Instituto Politécnico Nacional, ESIME, U.P. Culhuacán, Av. Santa Ana # 1000, Col. San
Francisco Culhuacán, C.P. 04430, México D.F.

E-mail: vponomar@ipn.mx

RESUMEN

Se presenta el sistema que se utilizó, para capturar las imágenes en tres dimensiones y la implementación del filtro tridimensional, usando algoritmos de estadística de orden modificados, para suprimir ruido y preservar los detalles finos de objetos. Se demostró el desempeño del procesamiento de imágenes en 3D, con base en criterios PSNR y MAE, usando simulaciones con ruido impulsivo y ruido gaussiano.

ABSTRACT

It is presented the system used to capture the images in 3D and the implementation of the three-dimensional filter. The filter uses different modified algorithms of order statistics to suppress noise and to preserve the fine details of objects. The testing of the prosecution of images was demonstrated in 3D with base in the criteria PSNR and MAE by using simulations in presence of impulsive and gaussian.

1 INTRODUCCIÓN

Los sistemas y equipos, tanto médicos como de telecomunicaciones se están diseñando para enviar y desplegar información en imágenes de tres dimensiones (3D). La posibilidad de procesar imágenes (campos) en 3D nos conduce a una nueva aplicación donde es necesario mejorar la calidad de objetos 3D, dentro de la imagen, con supresión de ruido que siempre afecta el proceso de comunicación o adquisición de señales. Otro problema que no es trivial es la modificación (adaptación) e implementación de filtros existentes e investigados en literatura en procesamiento bidimensional (2D) para procesar objetos en 3D [1,2].

2 SISTEMA PARA ADQUIRIR IMÁGENES DE TRES DIMENSIONES

Basándose en los artículos publicados [3,4], se tomó la idea de cómo adquirir las imágenes de tres dimensiones.

El sistema descrito a continuación, es un sistema experimental. Esto nos permitió introducirnos al nuevo campo de aplicación de las imágenes de tres dimensiones, y llegar a nuestro objetivo principal: la implementación de los filtros no lineales en este tipo de imágenes.

El sistema tiene un grado de libertad, del control del movimiento del transductor el cual realiza exploraciones en línea recta, haciendo cortes transversales y longitudinales del órgano humano explorado.

El sistema consiste, en desplazar el transductor del ultrasonido en un carril. Al ir desplazándose el transductor por el carril, va adquiriendo imágenes 3D del órgano explorado, las cuales se



INVESTIGACIÓN E IMPLEMENTACIÓN DE LOS ALGORITMOS DE ESTADÍSTICA DE ORDEN PARA FILTRAR LAS IMÁGENES 3D

Luisa María Sánchez Pach¹ Wolodymyr I. Ponomarev²

¹Instituto Politécnico Nacional, ESIME, U.P. Zacatenco, Unidad Profesional Adolfo López Mateos No. Instituto Politécnico Nacional s/n.
e-mail: luisa.sanchez@ipn.mx

²Instituto Politécnico Nacional, ESIME, U.P. Chalchicomula, Av. Santa Ana # 1000, Col. San Francisco Chalchicomula, C.P. 04430, México D.F.
e-mail: wpomarov@ipn.mx

RESUMEN

Se presenta la implementación del filtrado tridimensional, usando algoritmos de estadística de orden modificados, para suprimir ruido y preservar los detalles de los objetos. Se demostró el desarrollo del procesamiento de imágenes en 3D, con base en criterios ISSE y MAE, usando simulaciones con ruido impulsivo y ruido gaussiano.

INTRODUCCIÓN

Los sistemas y equipos, tanto médicos como de telecomunicaciones, se están diseñando para enviar y desplegar información en imágenes de tres dimensiones (3D). La posibilidad de procesar imágenes (campos) en 3D, nos conduce a una nueva aplicación, donde es necesario mejorar la calidad de objetos 3D dentro de la imagen, con supresión de ruido que siempre afecta al proceso de comunicación o adquisición de campos. Otro problema que no es trivial, es la adaptación e implementación de filtros existentes e investigados en literatura [1,2] en procesamiento bidimensional (2D) a procesamiento en 3D.

Se utilizó un sistema experimental [3,4], para capturar imágenes en tres dimensiones; al cual nos permitió introducirnos al nuevo campo de aplicación de las imágenes de tres dimensiones, y llegar a nuestro objetivo principal, la implementación de los filtros no lineales en este tipo de imágenes.

En este artículo, se presenta el desarrollo y los resultados obtenidos de las pruebas de los filtros no lineales en imágenes 3D, en el cual se utiliza un cubo de $n \times n \times n$ voxels, el cual recorre voxel por voxel de toda la imagen. Los criterios tradicionales usados para presentar el rendimiento de los filtros no lineales, fueron: la relación pico de señal a ruido (PSNR), que caracteriza la supresión de ruido, y el error absoluto medio (MAE), para evaluar la preservación de contornos y detalles finos.

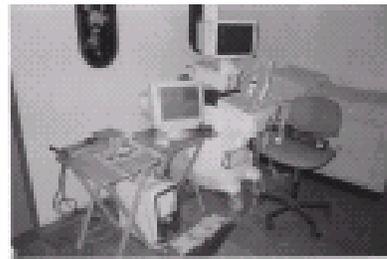


Figura 1. Sistema para adquirir imágenes de tres dimensiones.

ALGORITMOS DEL FILTRADO IMPLEMENTADOS

Para la implementación de los filtros no lineales, en imágenes de dos dimensiones, se utiliza una ventana de $m \times m$, la cual recorre toda la imagen pixel por pixel. Ahora, para implementar los filtros no lineales, en imágenes de tres dimensiones, se utiliza un cubo de $m \times m \times m$ voxels, el cual recorre toda la imagen de tres dimensiones, voxel por voxel. Implementando este método, se utilizaron los filtros de estadística de orden, adaptándose cada uno de ellos a una forma de tres dimensiones. Aquí se utilizó un cubo de 27 voxels, para realizar las simulaciones con ruido impulsivo y ruido gaussiano.

