



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

SÍNTESIS DE TEXTURAS PROCEDURALES
PARA UN SIMULADOR COMPUTARIZADO
DE CIRUGÍA DE PRÓSTATA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A :
GABRIELA GARCÍA GUEVARA

DIRECTOR DE LA TESIS:
M. C. MIGUEL ÁNGEL PADILLA CASTAÑEDA

CO-DIRECTOR DE LA TESIS:
DR. FERNANDO ARÁMBULA COSÍO



DEDICÓ ESTA TESIS A:

Mis padres:

Raquel Guevara y Efraín G. García,
que con su amor, cariño, paciencia,
ejemplo y apoyo incondicional se ve
culminada una de mis grandes metas.

Mi hermana:

Angélica, *por ser mi amiga y*
compañera de toda mi vida.

Mi tía:

Vange, *como un testimonio de cariño*
por estar siempre al pendiente de mí.
Dándome consejos, palabras de aliento y
superación.

Mi familia:

A mis abuelos, tíos y primos, *por*
todas sus muestras de afecto y cariño;
mejor familia no podía tener.

AGRADECIMIENTOS

*Como un testimonio de gratitud y eterno reconocimiento a **M.C. Miguel Ángel Padilla Castañeda**, por su valiosa dirección e interés en la culminación de este proyecto de tesis; y que a pesar de nuestros distintos compromisos y actividades siempre me apoyo y alentó a concluir mis estudios.*

*Agradezco al **Dr. Fernando Arámbula Cosío** por su apoyo, ayuda y colaboración en la finalización de este trabajo.*

*Mi más sincero reconocimiento al apoyo y la colaboración de todos los integrantes del equipo del Simulador RTU, que gracias al trabajo de todos, se ha ido mejorando dicho proyecto. Y en especial a **Sergio Teodoro Vite** por sus consejos y su gran ayuda que recibí al final, en la incorporación de mi trabajo de tesis al simulador.*

*También agradezco al laboratorio de **Análisis de Imágenes y Visualización del CCADET, UNAM**. Por las facilidades otorgadas para la realización de este trabajo.*

*Sin olvidar mi más sincero agradecimiento a mis amigos de la facultad: **Fer, Erika, Guillermo, Juan Carlos y Argelia** por ayudarme y estar conmigo a lo largo de la carrera.*

En general agradezco a todas las personas que de manera desinteresada me brindaron su ayuda e influyeron positivamente para la terminación de este trabajo.

*Finalmente agradezco enormemente a la **UNAM**, y en especial a la **Facultad de Ingeniería**, por proporcionarme las bases y las herramientas de mi carrera profesional y poder llegar a ser lo que soy. ¡Gracias!*

ÍNDICE

ÍNDICE DE FIGURAS	4
ÍNDICE DE TABLAS	7
RESUMEN.....	9
CAPÍTULO 1.....	10
1. INTRODUCCIÓN	10
1.1 <i>Texturas</i>	11
1.1.1 <i>Texturas sintéticas</i>	11
1.2 <i>Formas de obtención de texturas sintéticas</i>	12
1.3 <i>Aplicación de textura sintética a un modelo computarizado</i>	14
1.3.1 <i>Textura plana</i>	14
1.3.2 <i>Textura sólida</i>	15
1.3.2.1 <i>Texturas generadas por funciones o procedimientos</i>	16
1.4 <i>Síntesis de texturas procedurales para un simulador virtual</i>	16
1.5 <i>Hiperplasia prostática benigna (HPB)</i>	17
1.5.1 <i>Resección transuretral de próstata</i>	18
1.5.2 <i>Muestra de imágenes medicas reales de una RTU</i>	19
1.6 <i>Objetivos en la generación de texturas</i>	20
1.6.1 <i>Textura del tejido de la uretra con vascularidades</i>	20
1.6.2 <i>Textura del tejido interno de la próstata (apariciencia amarillo - pardo)</i>	21
1.6.3 <i>Textura del tejido de la cápsula de la próstata (apariciencia rugosa)</i>	23
CAPÍTULO 2.....	25
2. FUNDAMENTOS TEÓRICOS	25
2.1 <i>Definición de textura procedural</i>	26
2.1.1 <i>Breve cronología de texturas procedurales</i>	27
2.1.2 <i>Ventajas de las texturas procedurales</i>	28
2.1.3 <i>Desventajas de las texturas procedurales</i>	29
2.2 <i>Generación de patrones procedurales</i>	30
2.2.1 <i>Función de ruido</i>	31
2.2.2 <i>Función de ruido de Perlin</i>	32
2.2.3 <i>Fractales</i>	36
2.2.3.1 <i>Uso de fractales en la generación de texturas</i>	37
2.2.4 <i>Conceptos básicos para la generación de texturas procedurales</i>	38
2.2.4.1 <i>Exponente de Hurst (H)</i>	38
2.2.4.2 <i>Lacunaridad</i>	39
2.2.4.3 <i>Octavas</i>	39
2.2.4.4 <i>Offset</i>	40
2.2.4.5 <i>Umbral</i>	40
2.2.5 <i>Patrones utilizados para generación de texturas procedurales</i>	40
2.2.5.1 <i>Función movimiento Browniano fraccional</i>	40
2.2.5.2 <i>Función de turbulencia</i>	42
2.2.5.3 <i>Función de terreno heterogéneo</i>	43
2.3 <i>Aliasing</i>	45
2.3.1 <i>Antialiasing</i>	46
2.4 <i>Atributos de las superficies</i>	47
2.4.1 <i>Color</i>	47
2.4.2 <i>Uso del color</i>	47
2.4.3 <i>Matiz, Saturación y Brillo</i>	48

2.5	<i>Espacio de color en software grafico</i>	49
2.5.1	Modelo de color RGB	49
2.5.2	Modelo de color HSV	50
2.5.3	Conversión entre modelos HSV y RGB	52
CAPÍTULO 3.....		53
3.	SÍNTESIS DE TEXTURAS PROCEDURALES PARA UN SIMULADOR VIRTUAL DE CIRUGÍA	53
3.1	<i>Funciones básicas para la generación de texturas procedurales</i>	54
3.1.1	Funciones procedurales básicas.....	54
3.1.1.1	Función de ruido.....	55
3.1.1.2	Función movimiento Browniano fraccional.....	57
3.1.1.3	Función terreno heterogéneo	58
3.1.1.4	Función turbulencia.....	59
3.1.1.5	Función turbulencia modificada (apariencia de pliegues)	60
3.2	<i>Uso de los canales del modelo HSV</i>	61
3.3	<i>Generación de texturas procedurales de tejidos de la próstata</i>	63
3.3.1	Textura del tejido de la uretra (vascularidades).....	63
3.3.1.1	Diagrama de flujo para la generación de textura con vascularidades	64
3.3.2	Textura del tejido interno de la próstata (apariencia amarillo - pardo).....	69
3.3.2.1	Diagrama de flujo para la generación de textura (apariencia amarillo - pardo).....	70
3.3.3	Textura del tejido de la cápsula de la próstata (apariencia rugosa).....	75
3.3.3.1	Diagrama de flujo para la generación de textura del tejido cápsula	76
3.3.3.2	Conversión de modelos HSV a RGB para textura del tejido de la cápsula	78
3.4	<i>Generación de textura volumétrica procedural de la próstata</i>	82
3.4.1	Procesos para la generación de la textura volumétrica	83
3.4.2	Extensión de los algoritmos generadores de textura 2D a un volumen en 3D.....	84
3.4.2.1	Agregando parámetro "z" a los algoritmos generadores de textura	84
3.4.2.2	Rotación de textura 3D del tejido de la cápsula.....	85
3.4.2.3	Integración de las tres texturas 3D con imágenes previas de contorno de próstata	85
3.4.2.4	Obtención de la pila: textura volumétrica final de la glándula prostática	89
3.4.2.5	Verificación del número de imágenes previas de contorno de próstata	90
3.4.2.6	Agregar tapas de textura de tejido con vascularidades en los extremos del cubo	91
3.5	<i>Mapeo de texturas al simulador de cirugía virtual</i>	91
CAPÍTULO 4.....		94
4.	PRUEBAS Y RESULTADOS OBTENIDOS EN LA GENERACIÓN DE TEXTURAS PROCEDURALES	94
4.1	<i>Matlab: generación de texturas</i>	95
4.2	<i>Resultados del modelado de texturas procedurales al variar sus parámetros</i>	95
4.2.1	Textura del tejido de la uretra con vascularidades	96
4.2.1.1	Función terreno heterogéneo	96
4.2.1.2	Función turbulencia.....	99
4.2.1.3	Función movimiento Browniano fraccional (fBm)	102
4.2.2	Tejido colaginoso interno de la próstata (apariencia amarillo-pardo).....	105
4.2.2.1	Función turbulencia.....	106
4.2.2.2	Función movimiento Browniano fraccional (fBm1)	108
4.2.2.3	Función movimiento Browniano fraccional (fBm2)	111
4.2.3	Tejido fibroso de la cápsula (apariencia rugosa)	114
4.2.3.1	Función turbulencia modificada	114
4.3	<i>Modelos procedurales de las texturas generadas</i>	117
4.4	<i>Comparación de la textura generada con la muestra de tejido real</i>	119
4.4.1	Textura tejido de la uretra con vascularidades.....	119
4.4.2	Textura del tejido interno de la próstata (amarillo-pardo)	121
4.4.3	Textura del tejido de la cápsula de la próstata (apariencia rugosa)	123

4.5	<i>Comparación de tiempos en la generación de texturas</i>	125
4.6	<i>Interfaz gráfica para la generación de texturas procedurales para el simulador virtual</i>	128
4.6.1	<i>Funcionamiento básico de la interfaz</i>	128
4.7	<i>Pruebas y resultados en el simulador de RTU</i>	131
CAPÍTULO 5.....		135
5.	CONCLUSIONES Y PERSPECTIVAS	135
5.1	<i>Alcances generales</i>	136
5.2	<i>Trabajo a futuro</i>	137
APÉNDICE		138
A.	<i>Código fuente: Pila de Tejido de la Próstata en 3D</i>	138
BIBLIOGRAFÍA Y REFERENCIAS WEB		144

ÍNDICE DE FIGURAS

CAPÍTULO 1

Figura 1.1 Diferentes texturas sintéticas que nos dan la apariencia de distintos materiales las cuales pueden ser aplicadas a diferentes superficies.	12
Figura 1.2 a) Observamos que la textura 2D se “dibuja”. b) Concepto básico de mapeo de textura, con la correspondencia de una imagen 2D a una superficie 3D.	15
Figura 1.3 a) Observamos que la textura 3D se “esculpe”. b) Textura sólida en 3D mapeada en un modelo 3D, dando la apariencia de un material continuo de mármol.	16
Figura 1.4 Imagen izquierda: agrandamiento benigno (no canceroso) de la próstata. Imagen derecha: la misma próstata una vez realizado el procedimiento de resección transuretral.	18
Figura 1.5 Vista del agrandamiento de la próstata con ayuda del endoscopio antes del procedimiento. Se puede observar el tejido con vascularidades.	19
Figura 1.6 Vista de los cortes del procedimiento resección transuretral de la próstata (RTU) con el uso de un resectoscopio. Se pueden apreciar los cambios en las tonalidades de la textura después de realizar algunos cortes sobre el tejido.	19
Figura 1.7 Apariencia del tejido cauterizado de la próstata una vez realizado el procedimiento RTU.	19
Figura 1.8 Imagen de tejido real correspondiente al tejido de la uretra prostática.	20
Figura 1.9 Imagen de tejido real del efecto de cauterización sobre el tejido de la próstata con el procedimiento de resección transuretral.	21
Figura 1.10 Imagen de tejido real, muestra la apariencia del tejido de la cápsula de la próstata.	23
Figura 1.11 Esquema de la anatomía zonal de la próstata.	24

CAPÍTULO 2

Figura 2.1 Obtención de variaciones en texturas procedurales con la modificación del algoritmo base.	26
Figura 2.2 Imagen izquierda: función recursiva de cilindros. Imagen derecha: la misma función de cilindros pero añadiendo una función de ruido con lo cual obtenemos la simulación de un tronco de un árbol. ...	30
Figura 2.3 a) Función senoidal. b) Función de aleatoriedad (ruido).	32
Figura 2.4 Obtención de la función de Ruido de Perlin.	33
Figura 2.5 Diferentes resultados de la función de Ruido de Perlin con diferentes valores de persistencia.	34
Figura 2.6 Imagen que muestra la combinación de 6 octavas para crear la función de ruido de Perlin 2D.	35
Figura 2.7 Texturas 3D generadas en base a la función de Ruido de Perlin.	36
Figura 2.8 Concepto de la generación de fractal con la construcción del copo de nieve de von Koch.	37
Figura 2.9 Resultados de la variación del parámetro H (exponente de Hurst), variando de 1.0 a 0 con incrementos de 0.2.	38
Figura 2.10 Imágenes de fractales en donde se aprecia el concepto de lacunaridad.	39
Figura 2.11 Ejemplo del concepto de octavas, teniendo la función f original sumada a $\frac{1}{2}$ y $\frac{1}{4}$ de la misma, dando como resultado una textura con más detalles.	40
Figura 2.12 Esquema de la función movimiento Browniano fraccional (fBm).	41
Figura 2.13 Cubos texturizados a base de la función movimiento Browniano fraccional (fBm).	42
Figura 2.14 Cubos texturizados a base de la función Turbulencia.	43
Figura 2.15 Cubo texturizado con la función de terreno heterogéneo.	45
Figura 2.16 Muestro de una señal, en donde la reconstrucción de la señal presenta problemas de aliasing.	45
Figura 2.17 Modelo de color RGB, el cual define los colores con un proceso aditivo en el cubo unitario.	49
Figura 2.18 Modelo HSV representado por un hexágono.	50
Figura 2.19 Imagen en donde se visualiza el matiz del modelo de color HSV.	50
Figura 2.20 Resultado de variaciones de saturación y valor (brillo) del modelo HSV.	51
Figura 2.21 Imagen representativa de la conversión de modelos de color HSV y RGB.	52

CAPÍTULO 3

Figura 3.1 Texturas 2D generadas por computadora con la implementación de la función de ruido de Perlin a diferentes escalas.	56
Figura 3.2 Texturas 2D generadas por computadora. a) Textura uniforme. b) Textura uniforme con ruido aleatorio c) Textura uniforme con un valor fijo en la entrada de la función ruido.	57
Figura 3.3 Texturas 2D generadas con la función fBm.	58
Figura 3.4 Texturas 2D generadas con la función terreno heterogéneo.	59
Figura 3.5 Texturas 2D generadas con la función turbulencia.	60
Figura 3.6 Texturas 2D generadas con la función turbulencia modificada.	61
Figura 3.7 Con ayuda de la herramienta ImageJ se logra la conversión de una imagen a color RGB al modelo de color HSV. Con el modelo HSV se logra obtener tres capas en escalas de grises de la imagen a color correspondientes a cada canal de matiz, saturación y brillo (valor).	62
Figura 3.8 Descomposición de la imagen de tejido real de vascularidades al modelo de color HSV.	63
Figura 3.9 Resultado del producto de la función terreno heterogéneo y la función turbulencia, para la generación de la textura de vascularidades.	67
Figura 3.10 Resultado de la atenuación de la función que resultó del producto de funciones básicas: turbulencia y terreno heterogéneo.	67
Figura 3.11 Conversión de modelos HSV a RGB, obteniendo el resultado final de la textura sintética de vascularidades.	68
Figura 3.12 Descomposición de la imagen de tejido real interno de la próstata (apariencia amarillo - pardo) al modelo de color HSV.	69
Figura 3.13 Resultado del producto de la función turbulencia y la función fBm1, para textura (tejido colaginoso interno de la próstata).	73
Figura 3.14 Resultado al aplicar el umbral al resultado de la multiplicación de las funciones turbulencia y fBm1 para obtener el color de la textura final del tejido interno de la próstata.	73
Figura 3.15 Conversión de modelos HSV a RGB, obteniendo una textura sintética (tejido interno de la próstata).	74
Figura 3.16 Descomposición de la imagen de tejido real de la cápsula de la próstata (apariencia rugosa) al modelo de color HSV.	75
Figura 3.17 Resultados de las diferentes tonalidades en la generación de textura de la cápsula dependiendo del valor que se elija para el color.	77
Figura 3.18 Resultado de la conversión de modelos HSV a RGB del tejido de la cápsula en escala de grises.	79
Figura 3.19 Resultado de la conversión de modelos HSV a RGB del tejido de la cápsula en color negro con rosa.	80
Figura 3.20 Resultado de la conversión de modelos HSV a RGB del tejido de la cápsula en color rojo pálido.	80
Figura 3.21 Resultado de la conversión de modelos HSV a RGB, obteniendo una imagen de textura final para la cápsula de la próstata.	81
Figura 3.22 Diferencias entre un mapeo 2D y 3D sobre una superficie sólida en 3D.	82
Figura 3.23 Cubos 3D generados con las tres texturas del tejido de la próstata (vascularidades, apariencia amarilla y apariencia rugosa), estos resultados fueron obtenidos con la extensión de los resultados previos en 2D.	84
Figura 3.24 Rotación de textura de tejido de la próstata con apariencia rugosa.	85
Figura 3.25 Lectura de una imagen de segmento de la próstata, la cual posteriormente se escala al tamaño de la textura generada.	86
Figura 3.26 Separación de contornos de la imagen de próstata, a las cuales posteriormente se les aplicaran las operaciones morfológicas de erosión y dilatación.	86
Figura 3.27 Imagen de uretra texturizada. La texturización se logra con la mezcla de la textura de apariencia vascular y la textura de apariencia amarillo-pardo.	87
Figura 3.28 Imagen de próstata final texturizada. Se pueden apreciar las tres texturas simuladas, las cuales dan la forma de la glándula prostática.	88
Figura 3.29 Cubo 3D. Próstata texturizada para ser aplicada al simulador virtual de cirugía.	89
Figura 3.30 Ejemplificación del algoritmo cuando el número de imágenes de próstata en blanco y negro es menor que el cubo a generar.	90

Figura 3.31 Cubo generado en 3D, agregando dos capas de textura de tejido vascular para lograr una mejor visualización sobre el simulador.	91
Figura 3.32 Diferentes resultados del mapeado de la textura volumétrica sobre el simulador de cirugía de próstata. Se pueden apreciar las vascularidades de la textura, apareciendo zonas con venas “inflamadas”	93

CAPÍTULO 4

Figura 4.1 Comparación de la conversión del modelo de color HSV a RGB de la textura real y sintética del tejido con vascularidades.....	119
Figura 4.2 Resultados de los histogramas de la textura real y sintética del tejido con apariencia de vascularidades.....	120
Figura 4.3 Comparación de la conversión del modelo de color HSV a RGB de la textura con apariencia amarillo-pardo.	121
Figura 4.4 Resultados de los histogramas de la textura real y sintética del tejido con apariencia amarillo-pardo.	122
Figura 4.5 Comparación de la conversión del modelo de color HSV a RGB de la textura con apariencia rugosa.	123
Figura 4.6 Resultados de los histogramas de la textura real y sintética del tejido con apariencia rugosa.....	124
Figura 4.7 Interfaz gráfica que facilita la generación de las texturas procedurales para el simulador virtual de cirugía del procedimiento RTU.....	128
Figura 4.8 Botones y partes básicas de la interfaz gráfica para la generación de texturas procedurales en 2D. 129	
Figura 4.9 Botones y partes básicas de la interfaz gráfica para la generación de texturas procedurales en 3D.130	
Figura 4.10 Imágenes del simulador virtual RTU, CCADET. Con y sin textura.....	131
Figura 4.11 Imagen del modelo del simulador RTU , CCADET. Sin textura.	132
Figura 4.12 Imagen del modelo del simulador RTU, CCADET. Texturizado una sola imagen uniforme.	132
Figura 4.13 Imagen del modelo del simulador RTU, CCADET. Modelo texturizado con la síntesis generada a partir de tres texturas procedurales.	132
Figura 4.14 Imágenes de diferentes tomas del simulador virtual RTU, texturizado con la síntesis de texturas procedurales.	133
Figura 4.15 Imagen del simulador virtual RTU, una vez realizados algunos cortes.	134
Figura 4.16 Imágenes del simulador virtual RTU, una vez realizados algunos cortes y cauterización del tejido.....	134

ÍNDICE DE TABLAS

CAPÍTULO 4

Tabla 1. Resultados de la modificación del parámetro H, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.....	97
Tabla 2. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.	97
Tabla 3. Resultados de la modificación del parámetro OFFSET, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.....	98
Tabla 4. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.....	99
Tabla 5. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural turbulencia en la generación de la textura con vascularidades.	100
Tabla 6. Resultados de la modificación del parámetro OCTAVA INICIAL, dentro de la función procedural turbulencia en la generación de la textura con vascularidades.	101
Tabla 7. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural turbulencia en la generación de la textura con vascularidades.....	101
Tabla 8. Resultados de la modificación del parámetro FACTOR (valor constante), dentro de la función procedural turbulencia en la generación de la textura con vascularidades.....	102
Tabla 9. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural fBm en la generación de la textura con vascularidades.	103
Tabla 10. Resultados de la modificación del parámetro H, dentro de la función procedural fBm en la generación de la textura con vascularidades.	104
Tabla 11. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural fBm en la generación de la textura con vascularidades.	104
Tabla 12. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural fBm en la generación de la textura con vascularidades.....	105
Tabla 13. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural turbulencia en la generación de la textura con apariencia amarillo-pardo.....	107
Tabla 14. Resultados de la modificación del parámetro OCTAVA INICIAL, dentro de la función procedural turbulencia en la generación de la textura con apariencia amarillo-pardo.....	107
Tabla 15. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural turbulencia en la generación de la textura con apariencia amarillo-pardo.....	108
Tabla 16. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.	109
Tabla 17. Resultados de la modificación del parámetro H, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.	110
Tabla 18. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.....	110
Tabla 19. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.....	111

Tabla 20. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.	112
Tabla 21. Resultados de la modificación del parámetro H, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.	113
Tabla 22. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.....	113
Tabla 23. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.....	114
Tabla 24. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural turbulencia modificada en la generación de la textura con apariencia rugosa.....	115
Tabla 25. Resultados de la modificación del parámetro OCTAVA INICIAL, dentro de la función procedural turbulencia modificada en la generación de la textura con apariencia rugosa.....	116
Tabla 26. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural turbulencia modificada en la generación de la textura con apariencia rugosa.	116
Tabla 27. Visualización de resultados, en donde se puede apreciar las diferentes texturas sintéticas generadas a la largo de este trabajo de tesis. En donde se aprecia una de las ventajas de los métodos procedurales, es decir, la generación de texturas a diferentes tamaños, las cuales no pierden sus características visuales no importando su tamaño o dimensión.	118
Gráfica 1. Comparación de tiempos de 4 equipos, en la generación de la textura sólida en 3D de la glándula prostática.....	127

RESUMEN

En este trabajo de tesis se presenta el desarrollo de un modelo de síntesis de texturas procedurales, para ser utilizado en un simulador virtual de entrenamiento de resección transuretral de próstata (RTU) diseñado en el Laboratorio de Análisis de Imágenes y Visualización CCADET UNAM, con la finalidad de conseguir una textura de tejido lo más parecido visualmente al tejido mostrado en imágenes médicas reales de dicho procedimiento.

La obtención y generación de las texturas procedurales se puede dividir en: *textura del tejido de la uretra (vascularidades)*, *textura del tejido interno de la próstata (apariencia amarillo-pardo)* y *textura del tejido de la cápsula de la próstata (apariencia rugosa)*. Para posteriormente obtener una textura sólida procedural 3D del modelo anatómico de la próstata; la cual se obtiene con la composición de las tres texturas antes mencionadas, colocadas de acuerdo a la zona que le corresponde dentro del modelo anatómico (textura del tejido de la cápsula, textura del tejido de la uretra y textura del tejido interno).

Las texturas procedurales se desarrollaron a partir de funciones primitivas de patrones procedurales, con uso de funciones de ruido para la aleatoriedad, lo que nos proporcionó un aspecto más realista; con el uso de conceptos aplicables a los modelos fractales para la repetición de patrones, para obtener formas más naturales; y con uso de funciones trigonométricas y aritméticas con las cuales se pudo conseguir diferentes efectos sobre una misma función procedural. Una vez contando con todas las funciones y características antes mencionadas, se obtuvo una herramienta de software para la generación de dichas texturas en 2D y 3D de manera interactiva y rápida para el simulador de cirugía.

Los resultados obtenidos con este proceso de generación de texturas, lograron conseguir visualmente el efecto de continuidad sobre el modelo del simulador de cirugía, permitiendo realizar cortes sobre el tejido de la próstata sin perder las características y efectos visuales. Además de contar con una textura de vascularidades, lo que da pie a que en un futuro se lleve a cabo la implementación de la simulación de sangrado, basándose en la superficie de la textura vascular disparando dicho efecto cuando se corte una vena inflamada o un vaso sanguíneo de la textura.

Finalmente se obtuvieron las funciones y procedimientos de generación de texturas procedurales capaces de ser aplicables a diferentes modelos virtuales.

En el presente capítulo se comenzará explicando el concepto de textura. Se mencionarán brevemente los medios con los cuales se puede obtener una textura sintética, la cual suele ser muy útil, en el texturizado de modelos virtuales generados por computadora, dando origen a superficies visualmente muy parecidas a las que encontramos en la realidad.

Así mismo se plantearán los objetivos de este trabajo de tesis, que consiste en la realización de una síntesis de texturas procedurales para un simulador de entrenamiento de resección transuretral de próstata, con la cual se desea obtener una apariencia lo más parecido visualmente al tejido real del procedimiento antes mencionado.

1.1 Texturas

Las formas que frecuentemente encontramos en la naturaleza no muestran regiones con una intensidad de color o material uniforme. Por el contrario, suelen encontrarse con una rica y muy variada colección de texturas.

La superficie de cualquier objeto visible tendrá una cierta escala de textura. La cual se suele describir como suave, áspera, blanda, dura, gruesa, mate o brillante, etc.; patrones naturales que a menudo tienen una cualidad táctil.

En términos generales, la textura se refiere a la apariencia y característica de la superficie de un objeto dado por el tamaño, forma, densidad, disposición y la proporción de sus partes elementales. Esta propiedad se relaciona enormemente con la reflexión y absorción de la luz sobre la superficie del objeto, es decir, absorben y reflejan colores. La textura, por lo tanto, es una cualidad diferencial que ayuda a distinguir y reconocer los objetos, enriqueciendo la apariencia de los mismos. (Chen, *et al.*, 1998)

Una clasificación utilizada en el arte para identificar los tipos de texturas es la siguiente:

- **Naturales:** son las texturas que pertenecen a elementos de la naturaleza, como la piel de un elefante o la superficie de un pétalo de rosa.
- **Artificiales:** son las texturas de materiales fabricados por el hombre, como una pared o un papel de regalo.
- **Táctiles:** son las texturas que se percibe a través de la vista y del tacto, generalmente poseen relieve. Y son tridimensionales.
- **Visuales:** son las texturas que se perciben a través de la vista y no tienen relieve. Son bidimensionales y lo que intentan, a menudo, es imitar las texturas de los objetos reales para dar la sensación de realismo. Por ello también se les conoce como *texturas sintéticas*, existiendo diversos procedimientos para la obtención de estas texturas. (Ahearn, 2006)

1.1.1 Texturas sintéticas

Aprovechando las cualidades que nos dan nuestros sentidos, principalmente el del tacto y la vista. Es posible hacer simulaciones de diferentes texturas causando al espectador una apariencia “real” de la misma. Esta puede estar pintada, fotografiada, escaneada o generada por funciones o algoritmos por computadora. En la figura 1.1 se muestran diferentes texturas generadas por computadora, con las cuales es posible lograr efectos visuales muy interesantes aplicándolas a superficies de objetos tridimensionales.

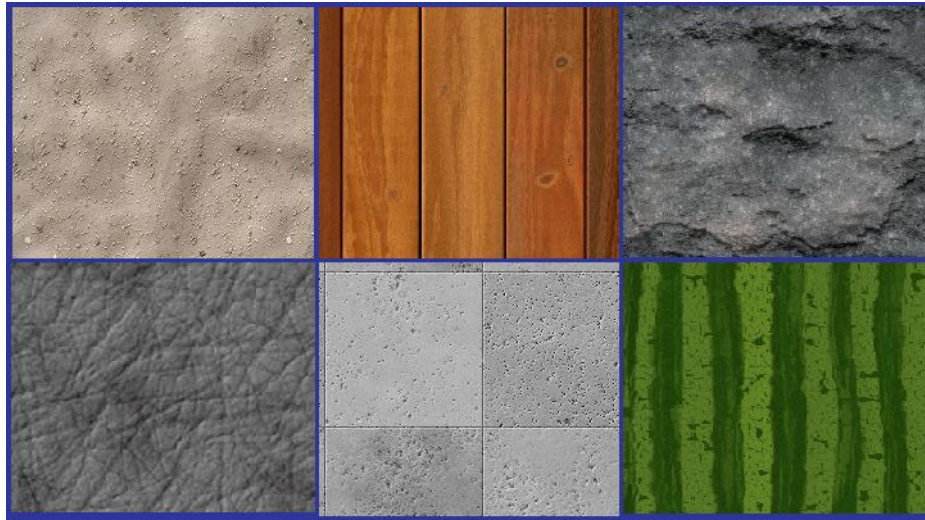


Figura 1.1 Diferentes texturas sintéticas que nos dan la apariencia de distintos materiales las cuales pueden ser aplicadas a diferentes superficies. *(Imágenes obtenidas de una base de datos de imágenes generadas por computadora)*

Así mismo la definición de textura en el área computacional se describe como:

- El número de tipos de tonalidades primitivas sobre un objeto y la organización espacial en donde se encuentran dichas tonalidades. *(Chen, et al., 1998)*

1.2 Formas de obtención de texturas sintéticas

Un factor importante para lograr el efecto de realismo en nuestras superficies generadas por computadora es agregar una textura adecuada. La obtención de dicha textura podrá ser mediante una fotografía, una imagen pintada o escaneada; o bien se puede obtener una textura generada por computadora usando funciones o formulas predefinidas, teniendo la ventaja de que estas últimas también las podemos obtener en 3D.

En el caso de una textura que proviene de una fotografía digital, deberá contar con una elevada resolución, en lo posible haber sido fotografiada desde una posición frontal, no atribuir la presencia ni de haces de luz reflejados ni de reflejos, e igualmente estar libre de distorsiones originadas por la lente fotográfica. *(ERCO-LightScout, 1997)*

Con una imagen de textura pintada, se requerirá indiscutiblemente mucho trabajo de un artista calificado, el cual debe incluir todos los detalles que se necesiten para la textura final. Las texturas pintadas son creadas casi siempre para un solo objeto específico; son generalmente uno de los tipos de obras que no pueden ser utilizadas en otras situaciones o para otros objetos (a menos que se requiera de un duplicado exacto).

En el caso de una textura escaneada tampoco es siempre conveniente, ya que puede ser difícil adquirir un área lo suficientemente grande del material que se requiera. Además,

de que en el proceso de escaneo de la imagen puede llegar a presentar deformaciones sobre la textura final que obtenemos, que generalmente no es lo que se desea.

Las formas de obtención de textura previamente mencionadas se refieren a imágenes en dos dimensiones, por lo cual podemos tener el problema de su limitada resolución, es decir, si se acerca demasiado a la imagen, puede presentar falta de detalles o incluso hacer visible los píxeles¹ originales de la textura.

Otro problema con este tipo de adquisición de textura, es que al mapearla sobre la superficie de nuestro objeto, puede presentar costuras o una repetición notable sobre el texturizado final. (*Apodaca, et al., 1999 págs. 243-245*)

Muchos de estos inconvenientes se corrigen con el uso de técnicas de generación de textura por medio de una computadora, las cuales se generan por funciones o formulas predefinidas.

Algunas de las ventajas de utilizar la generación de texturas por medios computarizados, es que pueden presentar los suficientes detalles a varias escalas sin tener distorsiones. Otra ventaja es que con estas texturas se pueden cubrir grandes áreas sin una repetición notable en el resultado final del mapeo.

Otra ventaja importante de las texturas generadas, es que se puede texturizar una modelo tridimensional, obteniendo una textura sólida de tres dimensiones, y al ser mapeadas a las superficies, podremos obtener vistas transversales de los modelos texturizados, dándonos la apariencia de continuidad del material como sucede en la realidad. (*Foley, et al., 1996*)

La mejor forma de obtener una textura dependerá en gran medida del resultado final que queramos obtener sobre el mapeado de nuestra superficie, así como las formas de adquisición de textura que tengamos más factible.

¹ **Pixel:** Unidad menor homogénea en color que forma parte de una imagen digital. Ampliando lo suficiente la imagen (zoom), puede observarse los píxeles que la componen.

1.3 Aplicación de textura sintética a un modelo computarizado

Como se menciona en el punto anterior, en la actualidad encontramos principalmente dos enfoques para la obtención de texturas para ser utilizadas en un modelo generado por computadora.

- El primer enfoque consiste en adquirir una textura plana la cual se define en un espacio de dos dimensiones, y puede ser obtenida de cualquiera de estas fuentes: parte de un dibujo, una imagen escaneada, o una fotografía digital.
- El segundo enfoque es obtener una textura sólida la cual se define en un espacio de dos o tres dimensiones por una composición de funciones matemáticas generadas por computadora. (*Turk, 1991*)

1.3.1 Textura plana

Se considera que la obtención de estas texturas puede hacerse muy rápidamente, aunque el resultado no siempre puede ser lo suficientemente bueno. Si un objeto es pequeño o no es un elemento central de la escena, una textura plana puede añadir bastante detalle. Pero en los objetos de mayor tamaño, dichas texturas probablemente tengan un relieve incorrecto.

El único requerimiento de la textura plana es que debe permitir el acceso a los distintos valores de los píxeles en la imagen. La textura debe almacenarse en una matriz bidimensional de tamaño $T_u \times T_v$ en donde a cada uno de los elementos de esta matriz de textura se le llama Texel².

Para pintar la textura sobre el objeto, es necesario que haya una relación entre las coordenadas de la textura con respecto a las del objeto. Es decir, debe existir una función de transformación de los puntos de un objeto dentro del rango de coordenadas de la textura.

Los valores de la textura se accede por medio de una matriz mediante un mapeo de las funciones (x,y) de los componentes del objeto y los índices (u,v) de la textura. Con lo cual se puede hacer una analogía y concluir que una textura de dos dimensiones es "pintada" sobre una superficie (ver figura 1.2) (*McConnell, 2006*)

² **Texel:** Unidad fundamental de un espacio de textura usado en graficación por computadora.

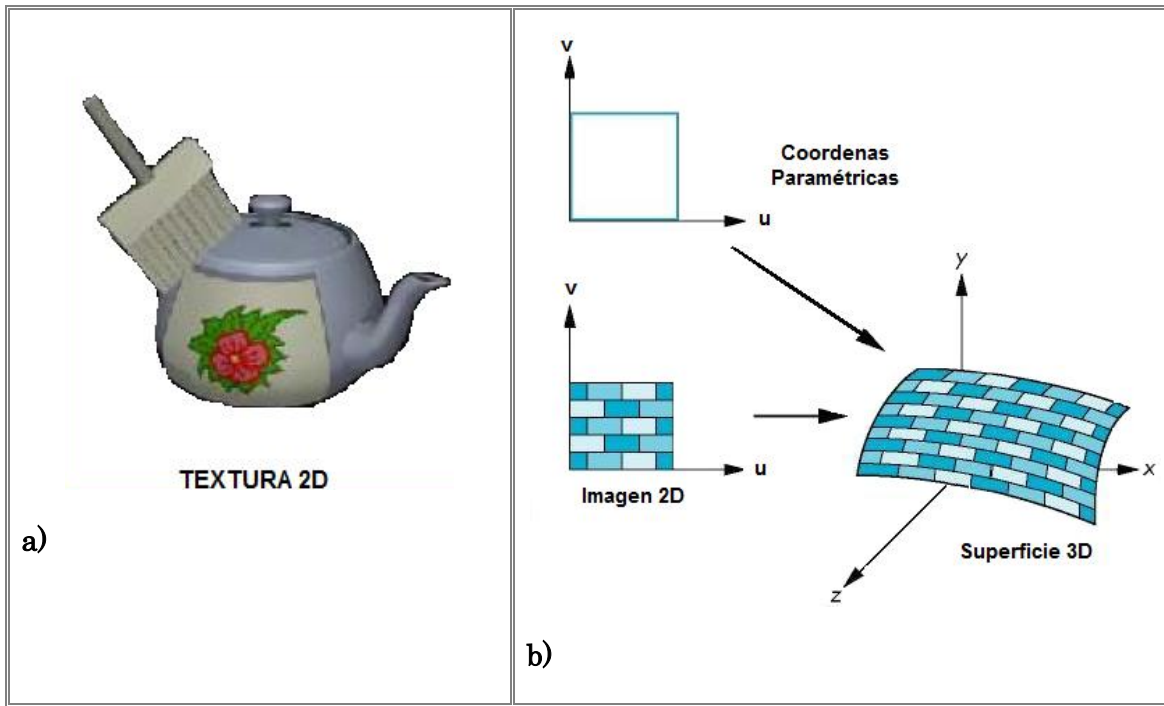


Figura 1.2 a) Observamos que la textura 2D se “dibuja”. b) Concepto básico de mapeo de textura, con la correspondencia de una imagen 2D a una superficie 3D. (Universidad Nacional del Sur, 2008)

1.3.2 Textura sólida

Un aspecto importante cuando se agrega textura a un objeto, es que debe existir continuidad; es decir, una vez mapeada una textura a un objeto se pueda diseñar de tal manera que no debe tener costuras visibles y una buena apariencia.

Por lo cual, si a una imagen cuadrada se intenta mapear alrededor de una esfera, las cosas se tornarán más complicadas. Y para entenderlo mejor se puede hacer la siguiente analogía: al intentar envolver una hoja de papel cuadrado en torno a un balón, tendremos como resultado golpes, arrugas e incluso la modificación de la geometría del objeto.

Por lo cual resulta difícil, pero no imposible, crear una textura que no que cause una costura visible cuando esta se envuelve alrededor de formas complejas. El problema está en el proceso de envolver la textura plana alrededor de un objeto que tiene curvatura. La alternativa es tener una textura de tres dimensiones. Ya que en vez de envolver la textura alrededor del objeto, el objeto se coloca en el espacio de la textura. Esto elimina la necesidad de crear una textura que tiene aristas que pueden ser reunidas en una variedad de maneras, con el resultado final de tener continuidad.

Para el modelo 3D a cada punto en la superficie del objeto se le asigne el color de la textura en función del valor de ese punto en el espacio. Este enfoque se adapta bien para la creación de objetos que parecen ser tallados de un material sólido como la madera o el mármol. (Turk, 1991)

Así una textura de tres dimensiones es un bloque en donde el objeto es "tallado" como se ilustra en la figura 1.3.

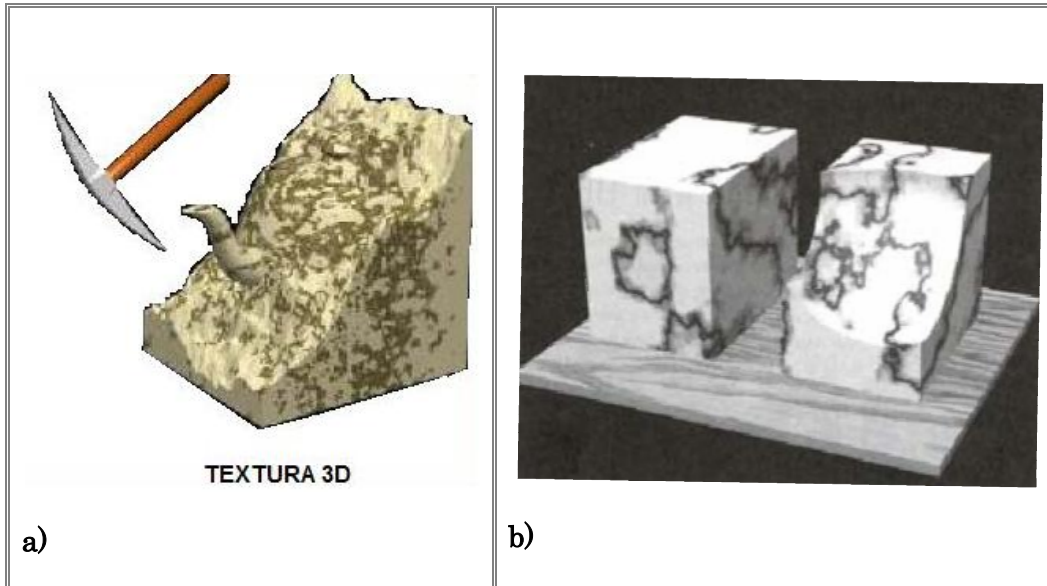


Figura 1.3 a) Observamos que la textura 3D se "esculpe". b) Textura sólida en 3D mapeada en un modelo 3D, dando la apariencia de un material continuo de mármol. (Universidad Nacional del Sur, 2008)

1.3.2.1 Texturas generadas por funciones o procedimientos

Dentro de las texturas sólidas se encuentran las texturas generadas por funciones o procedurales (procedimiento); en términos generales son texturas generadas a base de cálculos matemáticos. Utilizan generalmente el valor de (x, y) para el caso de dos dimensiones y en el caso de tres dimensiones utilizan los componentes (x,y,z) . (McConnell, 2006)

Las texturas procedurales han sido utilizadas para generar texturas de mármol, piedra, ladrillos, madera y papel tapiz. También se usan para definir humo, vapor, fuego, nubes, agua, etc. (Ebert, et al., 2002)

1.4 Síntesis de texturas procedurales para un simulador virtual

Conociendo la importancia y el efecto que nos da la aplicación de texturas a la apariencia final de nuestros modelos computacionales. Particularizando en los simuladores para entrenamiento médico, los cuales requieren de un modelado realista del aspecto visual de las texturas de los tejidos de los órganos a operar; y dado que las técnicas de adquisición de las imágenes a partir de las cuales se construyen los modelos, generalmente no proporcionan información de color que permita extraer la información textural del tejido; se requiere utilizar un método para sintetizar texturas que sean muy similares a los tejidos reales. Una alternativa es el uso de generación de textura sintética por medio de funciones procedurales.

El problema es que encontrar la textura correcta implica experimentar con los modelos procedurales que las generan, por lo que se requiere de una herramienta de software que permita modelar las imágenes en 2D y 3D de manera interactiva y rápida.

Comercialmente existen paquetes que generan texturas con imágenes pequeñas y en 2D, pero no proporcionan la información de la “función” que generó tal textura, por lo que no es posible extender el resultado en imágenes más grandes de 2D y 3D; por otro lado el software comercial es muy costoso.

En este trabajo de tesis se desarrollara un modelo de síntesis de texturas procedurales para su utilización en un simulador virtual de resección transuretral de próstata (RTU) (Arámbula, et al., 2006), con la finalidad de conseguir una textura de tejido lo más parecido visualmente al tejido real. Utilizando funciones primitivas de ruido, funciones de patrones procedurales, conceptos de modelos fractales, así como funciones trigonométricas y aritméticas.

El simulador por computadora para RTU por vaporización, es una de las líneas de investigación dentro del CCADET (Centro de Ciencias Aplicadas y Desarrollo Tecnológico) en el laboratorio de Análisis de Imágenes y Visualización, UNAM, que consistente en un ambiente virtual con un modelo en 3D de la próstata que permite simular deformaciones y resecciones de tejido blando con un resectoscopio virtual interfaz electromecánica pasiva) semejante a un resectoscopio real, a través del cual puedan entrenarse residentes en urología. (Padilla, et al., 2004 págs. 767-777)

1.5 Hiperplasia prostática benigna (HPB)

Para entender mejor el procedimiento de resección transuretral de próstata es necesario comentar brevemente que existe un padecimiento llamado *Hiperplasia Prostática Benigna* (HPB). La HPB consiste en un aumento de tamaño de la próstata de origen no cancerígeno

En la mayoría de los varones, la próstata empieza a crecer a partir de los 50 años y, en algunos casos, incluso antes. Con el paso de los años, la prevalencia de HPB histológicamente identificable aumenta, de manera que hacia los 60 años es más del 50% y hacia los 85 alcanza un 90% de los varones. (Clemente Ramos, et al., 2001)

La próstata del adulto tiene una forma similar a una castaña, con la base a nivel del cuello vesical y la uretra membranosa. Sus dimensiones suelen ser: 4 cm de ancho, 3 cm de largo y 2 cm de espesor. Está formada por tejido glandular y estroma³ fibromuscular. En un determinado momento de la vida del individuo, próximo a los 50 años (e incluso antes), en el interior de la glándula prostática se inicia un crecimiento singular que se traduce en la formación de nódulos de tamaño variable, constituyendo lo que conocemos con el nombre de hiperplasia. Este proceso entraña una multiplicación de las células epiteliales y del músculo liso, así como un desarrollo exagerado del tejido conjuntivo en las glándulas que tapizan la

³ **Estroma:** Armazón de un tejido, que sirve para sostener entre sus mallas los elementos celulares.

uretra prostática proximal. Este crecimiento hiperplásico está estimulado por hormonas testiculares y comprime el resto del tejido prostático normal contra la cápsula fibrosa, formando todo ello una “cápsula quirúrgica”, consistente en tejido prostático normal comprimido y vasos venosos. (Clemente Ramos, et al., 2001)

1.5.1 Resección transuretral de próstata

Un procedimiento estándar que se utiliza para tratar la hiperplasia prostática benigna es la resección transuretral de la próstata, sus siglas *RTU*. La RTU, consiste en remover el tejido blando de la próstata hipertrofiada alrededor de la uretra y restablecer el flujo urinario. El cirujano inserta en la próstata del paciente un resectoscopio que cuenta con un endoscopio y un elemento de resección. Con el elemento de resección el cirujano corta el tejido blando de los lóbulos hipertrofiados que obstruyen la uretra, guiándose a través del endoscopio. La vejiga es llenada con una solución, de tal forma que el cirujano pueda ver mejor su interior. En la figura 1.4 se esquematiza el agrandamiento de la próstata debido a la HPB, en comparación con el tejido removido después de hacer la resección transuretral de la próstata. (Stoelting, y et al., 2003)



Figura 1.4 Imagen izquierda: agrandamiento benigno (no canceroso) de la próstata. Imagen derecha: la misma próstata una vez realizado el procedimiento de resección transuretral. (Mayo Clinic, 2000)

1.5.2 Muestra de imágenes médicas reales de una RTU

En las siguientes figuras 1.5, 1.6 y 1.7 podemos observar imágenes médicas reales proporcionadas de una RTU; con la finalidad de analizar los diferentes tejidos que se observan durante el procedimiento.

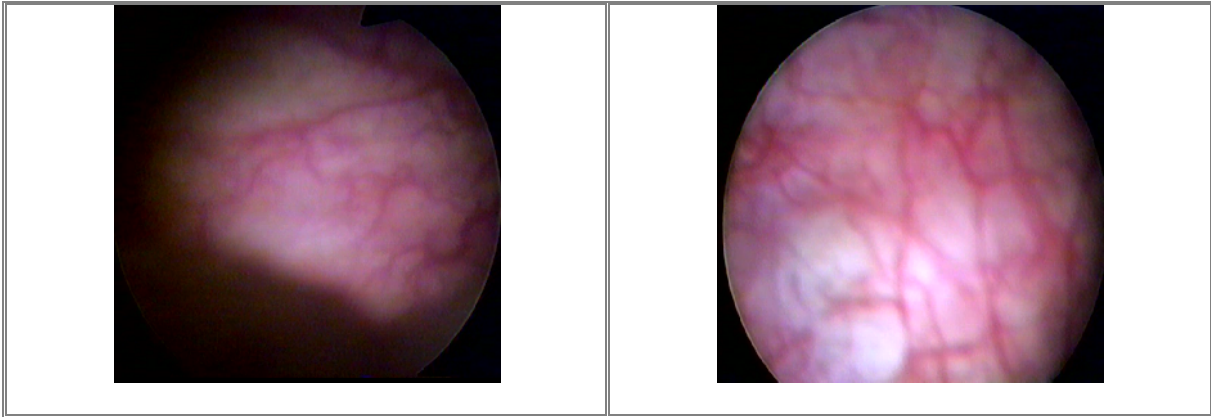


Figura 1.5 Vista del agrandamiento de la próstata con ayuda del endoscopio antes del procedimiento. Se puede observar el tejido con vascularidades. *(Imágenes del procedimiento RTU, obtenidas a partir de un video médico)*

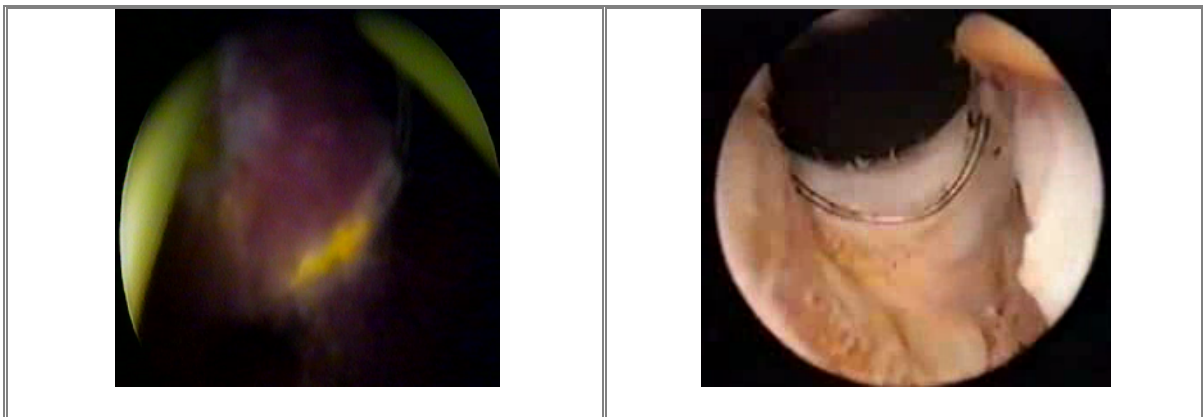


Figura 1.6 Vista de los cortes del procedimiento resección transuretral de la próstata (RTU) con el uso de un resectoscopio. Se pueden apreciar los cambios en las tonalidades de la textura después de realizar algunos cortes sobre el tejido. *(Imágenes del procedimiento RTU, obtenidas a partir de un video médico)*

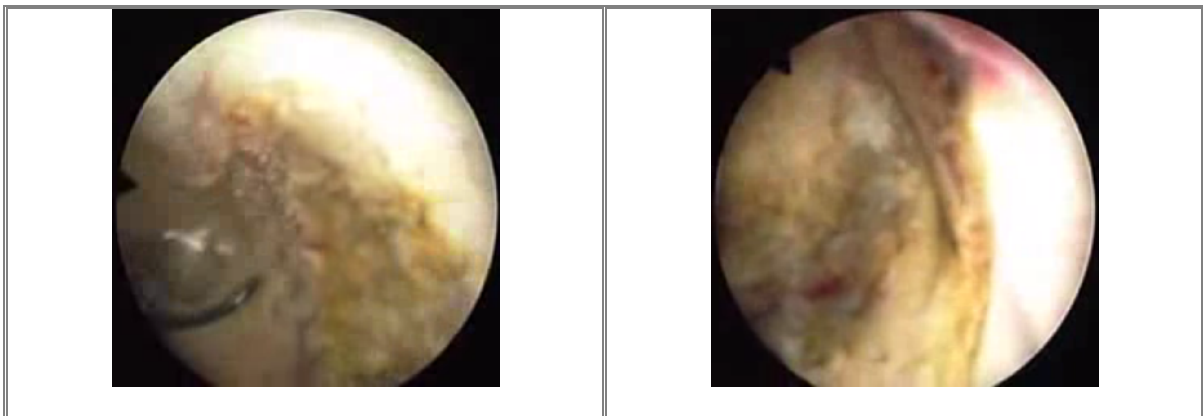


Figura 1.7 Apariencia del tejido cauterizado de la próstata una vez realizado el procedimiento RTU. *(Imágenes del procedimiento RTU, obtenidas a partir de un video médico)*

1.6 Objetivos en la generación de texturas

El objetivo principal de este trabajo de tesis, es generar tres texturas de tejido visualmente muy parecidas a las imágenes médicas reales de los tejidos mostrados en videos del procedimiento de resección transuretral de próstata (RTU).

Dichos tejidos se dividieron en tres texturas principales:

- textura del tejido de la uretra (vascularidades)
- textura del tejido interno de la próstata (apariencia amarillo - pardo)
- textura del tejido de la cápsula de la próstata (apariencia rugosa)

Las texturas se generaran con base en modelos formados por la combinación de funciones procedurales básicas, utilizando los antecedentes y fundamentos teóricos planteados en este trabajo.

Una vez contando con dichas texturas, se procederá a extender los resultados en un modelo sólido en tres dimensiones para ser finalmente aplicado al simulador virtual de cirugía.

1.6.1 Textura del tejido de la uretra con vascularidades

En la figura 1.8 se muestra la imagen del tejido real, que corresponde visualmente al tejido de la uretra.

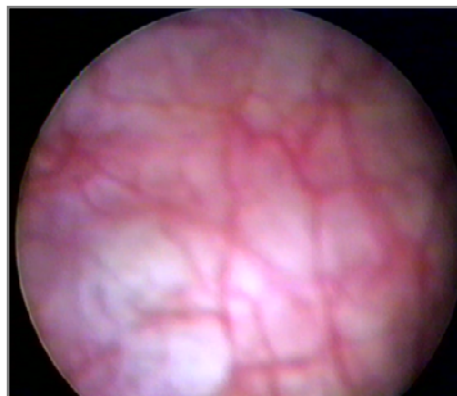


Figura 1.8 Imagen de tejido real correspondiente al tejido de la uretra prostática. *(Imagen del procedimiento RTU, obtenida a partir de un video médico)*

Después de estudiar varios ejemplares de este tipo de imagen se observaron las siguientes características a sintetizar:

- Primordialmente una apariencia general de color rosa.
- Con apariencia de vascularidades en color rojo.
- Y algunas manchas en blanco.

Durante el procedimiento de la RTU, el cirujano no debe tolerar ninguna hemorragia. Por lo cual debe realizar la coagulación de cualquier vaso que sangre tras cada corte, entendiendo que un chorro rojo y entre-cortado se coagulara de inmediato, una arteria de mayor calibre requerirá una técnica más precisa de coagulación. Es importante la práctica y la técnica de la coagulación, ya que el cirujano debe contar siempre con un campo quirúrgico limpio el cual le ayudara a tener un avance rápido y sin complicaciones. (*Abbou, et al., 2007*)

Entendiendo lo anterior, y sabiendo la característica del tejido vascular en esta área. Es importante la síntesis de una textura con apariencia de vascularidades para que en un futuro se pueda llevar a cabo la simulación de sangrado cuando “corten” una vena inflamada o vaso sanguíneo en el simulador de cirugía; texturalmente hablando se mostrara el efecto de sangrado cuando toquen o se acerquen al intervalo de los valores de color rojo en la textura de vascularidades. Esto con la finalidad de contar con un sistema cada vez más parecido a las situaciones que se presentan al realizar el procedimiento como sucede en la realidad.

1.6.2 Textura del tejido interno de la próstata (apariencia amarillo – pardo)

En la figura 1.9 se muestra una imagen en donde se aprecia al efecto visual de la cauterización del tejido causado por el procedimiento de resección transuretral de próstata.

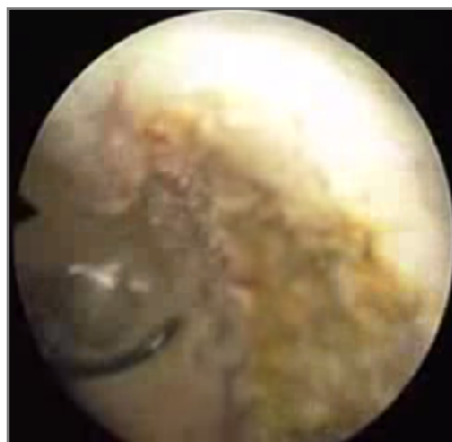


Figura 1.9 Imagen de tejido real del efecto de cauterización sobre el tejido de la próstata con el procedimiento de resección transuretral. (*Imagen del procedimiento RTU, obtenida a partir de un video médico*)

Después de estudiar varios ejemplares de este tipo de imagen se observaron las siguientes características a sintetizar:

- Tejido de apariencia de color amarillo.
- Algunas áreas blancas.
- Con pocas manchas en rojo.

Como se menciona con anterioridad el procedimiento de la RTU se realiza con la ayuda de un resectoscopio, el cual tiene integrado un elemento de resección metálico en forma de asa que se encuentra conectado a una fuente de energía eléctrica con la posibilidad de cortar y coagular el tejido prostático. Para el corte se utiliza una potencia de 120-300 watts, mientras que para la cauterización se utiliza una potencia de 50-116 watts, esta medida será establecida dependiendo del grosor de la resección y la técnica del cirujano. *(CEC Electromedicina)*

Se pueden realizar cortes o cauterizar al tejido prostático debido a la corriente que pasa a través de la resistencia eléctrica del elemento de resección metálico, el cual calienta y eleva la temperatura del tejido. En el corte, el objetivo es calentar las células del tejido tan rápidamente como sea posible para que exploten vaporizándose, dejando una cavidad. Cuando se mueve el elemento de resección y hace contacto en otra parte del “tejido fresco”, explota nuevas células realizando otra incisión.

En el caso de la cauterización puede centellear al tejido sin un efecto de corte significativo porque el calor es más ampliamente dispersado por las chispas largas y porque el efecto de calentamiento es intermitente. La temperatura del agua en las células no se eleva lo suficiente para despedir vapor. De esta forma las células se deshidratan lentamente, pero no se rompen para formar una incisión o corte. Lo que se aprecia visualmente es un tejido que se torna de un color amarillento-marrón claro. *(CEC Electromedicina)*

Por lo cual es importante la síntesis de una textura de un color amarillo-pardo para poder simular el efecto de cauterización del tejido, el cual está presente en el procedimiento real de una RTU, lo que nos daría en el simulador un aspecto visual más real cuando se trabaja con el elemento de resección, con el efecto de cauterización sobre el tejido prostático.

1.6.3 Textura del tejido de la cápsula de la próstata (apariencia rugosa)

En la figura 1.10 podemos apreciar una imagen médica real de la cápsula de la próstata, cuando se hace la exploración utilizando un endoscopio.

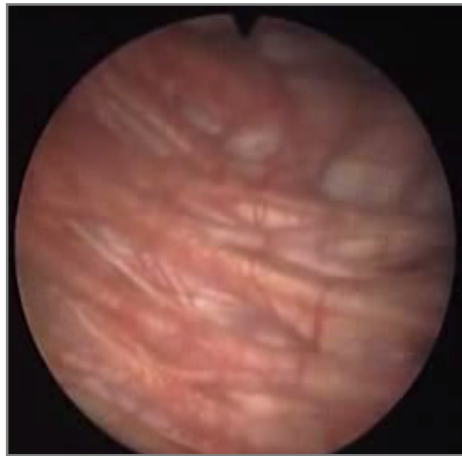


Figura 1.10 Imagen de tejido real, muestra la apariencia del tejido de la cápsula de la próstata. *(Imagen obtenida a partir de un video médico)*

Se estudiaron un par de ejemplares de este tipo, pudiendo sintetizar las siguientes características:

- Tejido de apariencia rugosa.
- Color predominante en rosa pálido.
- Algunas áreas en blanco.

La glándula prostática está dividida principalmente en dos regiones de tejido fibromuscular y glandular. Y anatómicamente se encuentra dividida en cuatro zonas: periférica, central, de transición y anterior, las cuales podemos apreciar en la figura 1.11. *(Azcárraga, 2000)*

- La zona periférica representa el 70% de la masa glandular prostática, rodea la zona central, por detrás, lateralmente y por debajo.
- La zona central rodea los conductos eyaculadores y reagrupa el 25% del tejido glandular prostático.
- La zona de transición está formada por dos lóbulos alrededor del segmento proximal de la uretra, en la parte media de la próstata.
- La zona anterior se encuentra en la porción anterior de la próstata por delante de la uretra. Es la segunda en tamaño desde de la periférica, y está formada de estroma fibromuscular extendiéndose a los lados y atrás de la próstata en forma de una delgada capa que constituye la capsula prostática. *(Abbou, et al., 2007)*

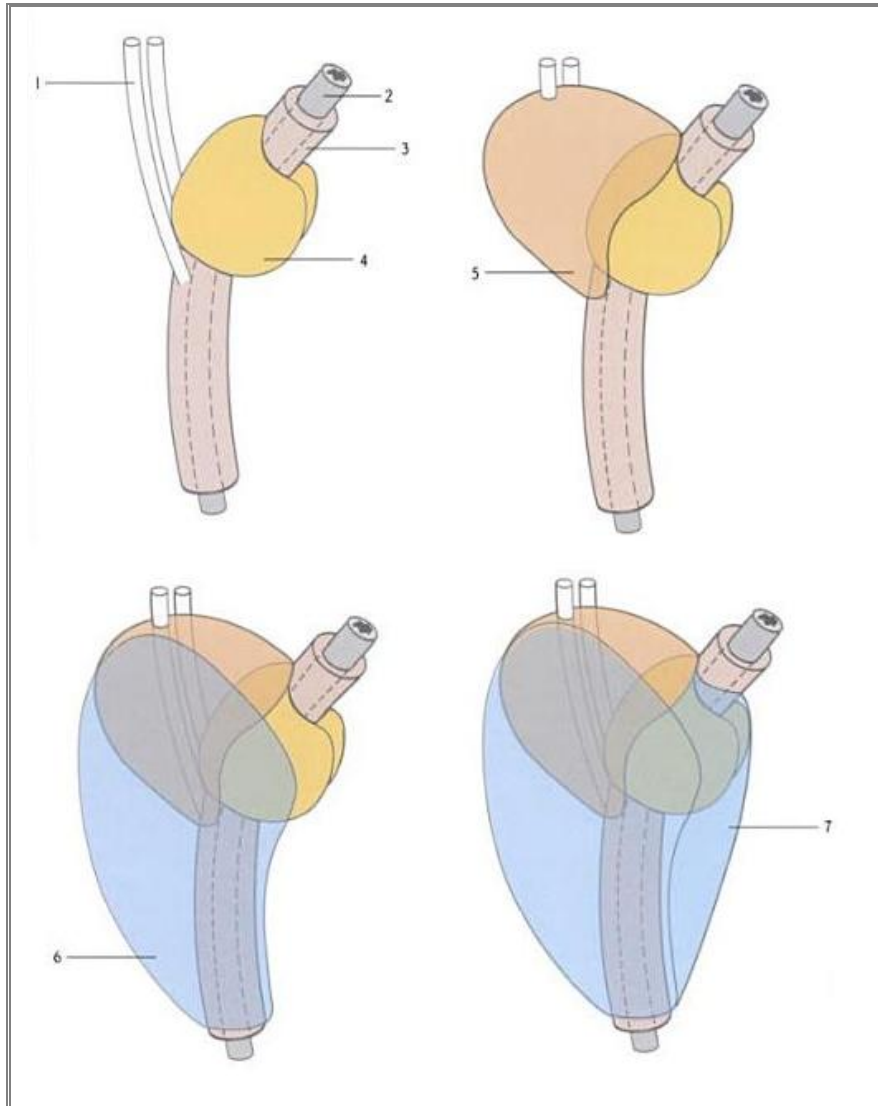


Figura 1.11 Esquema de la anatomía zonal de la próstata. (Abbou, et al., 2007 pág. 3)

- 1. Conductos eyaculadores. 2. Uretra. 3. Zona periuretral. 4. Zona de transición. 5. Zona central.*
6. Zona periférica. 7. Zona anterior fibromuscular.

La capsula prostática se compone de colágeno, elastina y abundante músculo liso. Teniendo un grosor de aproximadamente 0.5 mm. Visualmente se aprecia como un tejido con pliegues y ondulaciones. (Wein, et al., 2008)

Se generara una síntesis de una textura con ondulaciones (textura rugosa), para colocar en el contorno de la próstata, tratando de obtener una apariencia visual lo más parecida a las características de la composición real del tejido de la cápsula prostática.

2

2. FUNDAMENTOS TEÓRICOS

CAPÍTULO

En el presente capítulo se plantearán los fundamentos y antecedentes teóricos en los que se ha basado el desarrollo de este trabajo de tesis.

Se profundizará más en la generación de texturas procedurales, conociendo sus ventajas y desventajas; además de conocer algunas de las funciones y patrones que ayudan en su generación. Para finalmente definir otros atributos importantes como son: color, matiz, saturación y brillo, los cuales también sirvieron como fundamentos y antecedentes en este trabajo.

2.1 Definición de textura procedural

El adjetivo de procedural se utiliza en ciencias computacionales para distinguir las entidades que se describen en el código de un programa; es decir el conjunto de acciones y el orden en que se ejecutan estas acciones. Por lo cual casi cualquier cosa que hagamos en la computadora, tendrá un aspecto procedural en algún nivel.

Por ejemplo en el trazado de mapas de textura en una superficie, el componente procedural o de procedimiento es la creación del módulo de la textura.

Por lo cual podemos definir una *textura procedural* como una imagen sintética, la cual es generada a partir de un algoritmo⁴ o modelo. Sin embargo se puede incluir imágenes en la generación de texturas procedurales si a estas se les incorporaran algunas de las operaciones primitivas. Ya que algunas texturas procedurales con una buena apariencia visual, se han obtenido gracias a la combinación de procedimientos modificación o distorsión a partir de una imagen. (Ebert, et al., 2002)

Como se mencionó con anterioridad las texturas procedurales, son muy útiles en la creación de patrones de diferentes materiales, como vetas de madera o mármol, al utilizar funciones armónicas que se definen en un espacio tridimensional. Se puede obtener las variaciones en la textura de la madera o el mármol sobreponiendo una función de ruido en las variaciones armónicas, como se aprecia en la Figura 2.1. (Foley, et al., 1996)

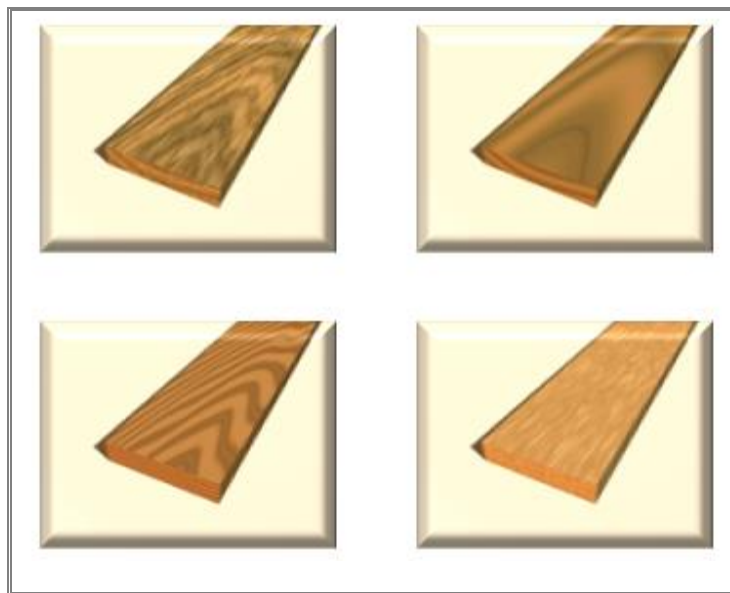


Figura 2.1 Obtención de variaciones en texturas procedurales con la modificación del algoritmo base. (Blender, 2004)

⁴ **Algoritmo:** En matemáticas, ciencias de la computación, y disciplinas relacionadas, un algoritmo es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema.

2.1.1 Breve cronología de texturas procedurales

Desde los primeros días del mapeado de texturas, una variedad de investigadores trabajaron con diferentes modelos de texturas para generar imágenes de texturas en lugar de imágenes escaneadas o pintadas. (Ebert, et al., 2002)

Blinn y Newell en el año de **1976** utilizaron la síntesis de Fourier.

Fu y Lu para **1978** propusieron una gramática sintética basada en la técnica de generación de texturas.

Schacter y Ahuja en **1979** y Schacter en **1980** utilizaron la síntesis de Fourier y modelos estocásticos de diversos tipos de texturas para generar imágenes de simuladores de vuelo.

Fournier, Fussell y Carpenter en el año **1982** además de Haruyama y Barsky en **1984** propusieron utilizar métodos de subdivisión estocásticos para generar texturas, mejor conocidos como fractales.

Otros investigadores desarrollaron modelos estadísticos analizando las propiedades de texturas naturales, y a continuación tratar de reproducir la textura con datos estadísticos.

Cook en el año de **1984** describió el sistema de "árboles de sombra", el cual fue uno de los primeros sistemas en los que fue conveniente generar un procedimiento de texturas durante el renderizado⁵.

Los árboles de sombra permitieron el uso de un modelo diferente de sombreado para cada superficie, así como para las fuentes de luz y de atenuación a través de la atmósfera. Debido a las aportaciones del modelo de sombreado, el modelo hizo posible el uso de texturas para controlar cualquier parte del cálculo de sombreado. Color y transparencia, reflejo de la imagen, el desplazamiento y las texturas sólidas pudieron todas ser implementadas usando el sistema de "árboles de sombra".

Perlin en el año de **1985** describió un completo lenguaje para la generación de texturas procedurales y asentó las bases para la clase más popular de texturas procedurales, es decir, todas aquellas que se basan en el ruido, una generación estocástica de textura primitiva.

Turk, Witkin y Kass en **1991** describieron modelos de textura sintética, inspirados en procesos bioquímicos que producen entre otros efectos patrones de pigmentación en la piel de los animales.

⁵ **Renderizado:** Proceso de generar una imagen desde un modelo. En términos de visualización en una computadora, la "renderización" es un proceso de cálculo complejo desarrollado por la computadora destinada a generar una imagen 2D a partir de una escena 3D.

Sims en el año 1991, describió un muy novedoso sistema de síntesis de textura, en el que las texturas se representan como expresiones automáticamente modificadas y combinadas por un sistema de programación genética. (Ebert, et al., 2002)

Todos los métodos de síntesis de textura mencionados en este tema se les podrían llamar procedurales o de procedimiento.

2.1.2 Ventajas de las texturas procedurales

Las ventajas que presentan las texturas procedurales sobre una imagen de mapa de bits⁶ son:

- La textura procedural es extremadamente compacta. El tamaño de la textura suele ser medidos en kilobytes, mientras que el tamaño de una imagen se suele medirse en megabytes. Esto es especialmente cierto en el caso de texturas sólidas en tres dimensiones.

Nota: Esta ventaja se refiere a la generación en tiempo real de la textura, no hay necesidad de almacenar propiamente la imagen.

- Una representación procedural no tiene una resolución fija. En la mayoría de los casos se puede proporcionar una textura totalmente detallada no importando que tan cerca la tenga el observador.
- Una textura procedural no tiene un tamaño fijo. En otras palabras, es ilimitada en la amplitud y puede cubrir áreas grandes, sin bordes o costuras no deseadas; además de no utilizar un patrón de repetición.
- Una textura procedural puede ser parametrizada, por lo que puede generarse una clase de texturas relacionadas para lo que se necesite.

Muchas de estas ventajas son sólo posibles mejoras para lograr la textura deseada; teniendo en cuenta que se debe hacer un esfuerzo para obtener lo que se pretende. Ya que

⁶ **Mapa de bits:** Estructura de datos que representa una imagen rectangular de píxeles o puntos de color, la cual puede ser visualizada en la computadora.

un mal procedimiento para la obtención de una textura puede sacrificar algún potencial de esas ventajas. (Ebert, et al., 2002)

2.1.3 Desventajas de las texturas procedurales

Las desventajas de las texturas procedurales en comparación con una imagen son las siguientes:

- Una textura procedural puede ser difícil de construir y de depurar. La programación es a menudo compleja, y la programación implícita en la descripción de un dibujo es especialmente difícil en los casos no triviales.
- Una textura procedural puede generar resultados sorprendidos; ya que a menudo es más fácil predecir el resultado al momento en que se escanea o pinta una imagen. Por lo que las texturas procedurales son difíciles de controlar.
- En cuestión de evaluación, puede ser más lento el acceder a una textura procedural que a una imagen almacenada, en el caso de generarla en tiempo real.
- El aliasing⁷ puede ser un problema en la generación de texturas procedurales; llevar a cabo el antialiasing puede ser complicado y es poco probable que sea resuelto de forma automática. (Ebert, et al., 2002)

⁷ **Aliasing:** Artefacto gráfico característico que presenta en pantalla ciertas curvas y líneas inclinadas con un efecto visual tipo "sierra" o "escalón" no deseados.

2.2 Generación de patrones procedurales

La generación de imágenes procedurales es posible porque muchas cosas a nuestro alrededor, tanto naturales como sintéticas, tienen propiedades geométricas, estocásticas (aleatorias) y de simetría que se pueden describirse matemáticamente.

En cuanto a la simetría, como lo describe el profesor Stewart "Los objetos matemáticos más simples son los números, y los más simple patrones de la naturaleza son numéricos. Por ejemplo, las fases de la luna en realizar un ciclo completo de luna nueva a luna llena, se realiza cada veintiocho días. Un año es de trescientos sesenta y cinco días. La gente tiene dos piernas, los gatos tienen cuatro, algunos insectos tienen seis y arácnidos tienen ocho. En casi todas las flores, el número de los pétalos es uno de los números que ocurre en la extraña secuencia de 3, 5, 8, 13, 21, 34, 55, 89, ..., también conocida como la Secuencia de Fibonacci". (Stewart, 1997)

En cuanto a la geometría, está se utiliza para describir con mayor facilidad a la naturaleza, por ejemplo, las hojas de un helecho en términos de modelos recursivos se define como la ocurrencia de una característica de forma a diferentes escalas, también conocido como *fractales*. Estas propiedades pueden verse en todas los escalas, desde las sierras y las formaciones de nubes hasta los detalles de las estructuras celulares en las plantas y los animales.

En cuanto a la aleatoriedad, otra función primitiva muy útil en la generación de texturas procedurales es la utilización de *ruido*. El ruido en sí mismo es sólo un número al azar o una colección de números aleatorios que por sí solos no son muy interesantes.

Sin embargo, si tomamos el ruido y lo utilizamos como una entrada de una función para cambiar la salida, las cosas cambian considerablemente. Para ilustrar mejor esto, imaginemos una función recursiva que genera un cilindro con un cilindro más pequeño dentro de él. Si ahora se añade algo de ruido en la función podemos cambiar la salida y producir algo que se asemeja a un tronco de un árbol como se aprecia en la figura 2.2. (Gudlaugsson, 2006)

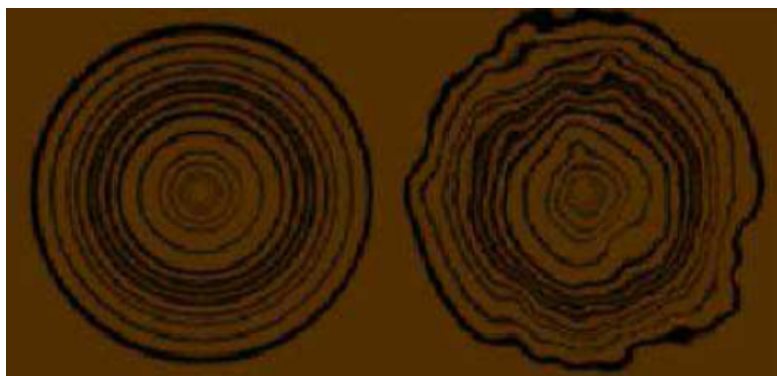


Figura 2.2 Imagen izquierda: función recursiva de cilindros. Imagen derecha: la misma función de cilindros pero añadiendo una función de ruido con lo cual obtenemos la simulación de un tronco de un árbol. (Gudlaugsson, 2006)

2.2.1 Función de ruido

Para la generación de texturas procedurales irregulares, se necesitan de funciones primitivas “irregulares” usualmente llamadas *ruido*. Esto entre otras cosas, es una función de naturaleza estocástica⁸ y rompe con la monotonía de los modelos que de otro modo serían demasiado regulares.

Cuando usamos términos como "aleatorio" y "estocástico", casi siempre quiere decir "aparentemente aleatorio".

La textura primitiva estocástica más evidente es el *ruido blanco*, una fuente de números aleatorios distribuidos de manera uniforme sin correlación alguna entre los siguientes números sucesivos; el ruido blanco puede ser generado por un proceso físico al azar, como el ruido térmico que se produce dentro de muchos sistemas electrónicos analógicos y no es nunca el mismo dos veces.

Muchas personas han utilizado los generadores de números aleatorios para crear sus programas con efectos de imprevisibilidad, por ejemplo para generar movimientos y comportamientos de los objetos con apariencia más natural, o generar texturas. Los generadores de números aleatorios ciertamente tienen sus usos, pero a veces su salida puede ser demasiado dura para un aspecto natural.

Casi cualquier cosa en la naturaleza tiene una naturaleza estocástica: la desigual distribución de césped en un campo, las olas en el mar, los movimientos de una hormiga, el movimiento de las ramas de un árbol, las pautas de mármol, los vientos. Todos estos fenómenos muestran el mismo patrón de grandes y pequeñas variaciones. Una función muy común para generar este tipo de patrones es, como se explicará más adelante, la función de ruido de Perlin; esta se recrea simplemente sumando ruidosos funciones en una variedad de diferentes escalas. Para crear una función de ruido Perlin, se necesitan dos cosas, una función de ruido, y una función de interpolación. (*Ebert, et al., 2002*)

⁸ **Estocástico:** Entendemos el concepto de estocástico como un fenómeno que generalmente tiene asociadas fuentes de variación o valores aleatorios, que se encuentran fuera del control del tomador de decisiones. (*Azarang, et al, 1996*)

2.2.2 Función de ruido de Perlin

Definiciones

Primeramente para entender la generación de la función de ruido de Perlin, se debe recordar lo que es amplitud y frecuencia. En la figura 2.3 se muestra una función periódica o senoidal y una función aleatoria (ruido).

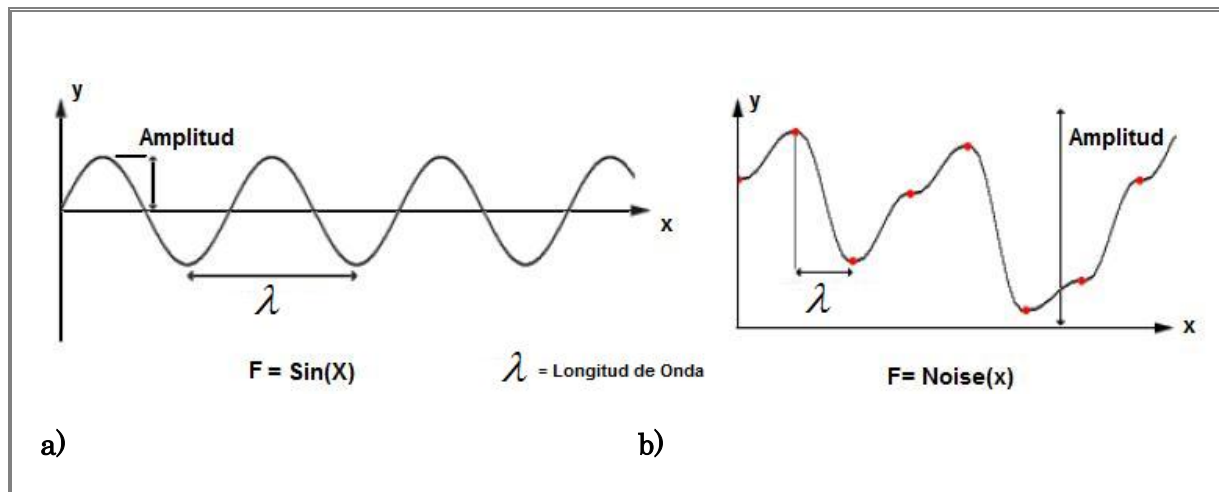


Figura 2.3 a) Función senoidal. b) Funcion de aleatoria (ruido). (Elias, 2003)

En el caso de la función senoidal, la *amplitud* es la distancia que hay entre el eje x y el valor máximo o mínimo de la función (cresta o valle). La *longitud de onda* es la distancia entre crestas, valles o partes idénticas sucesivas. El *periodo* lo podemos definir como el tiempo necesario para la repetición completa de un ciclo. Y el número de ciclos por unidad de tiempo recibe el nombre de *frecuencia*, la frecuencia y el periodo son recíprocos entre si, por lo cual se puede escribir la siguiente fórmula. (Resnick, 1999)

$$frecuencia = \frac{1}{periodo}$$

En la función de ruido los puntos rojos indican los valores aleatorios que se obtuvieron de la función. En este caso, la *amplitud* es la diferencia entre el valor mínimo y máximo de la función. La *longitud de onda* es la distancia desde un punto al siguiente. Y la *frecuencia* se obtiene con la misma fórmula antes menciona.

Creación de la función de ruido de Perlin

Ahora bien, si se trabaja con funciones aleatorias a diferentes frecuencias y amplitudes, y se suman todas, se crea una agradable función de ruido. Esto es a lo que se conoce como la función de ruido de Perlin ejemplificada en la figura 2.4.

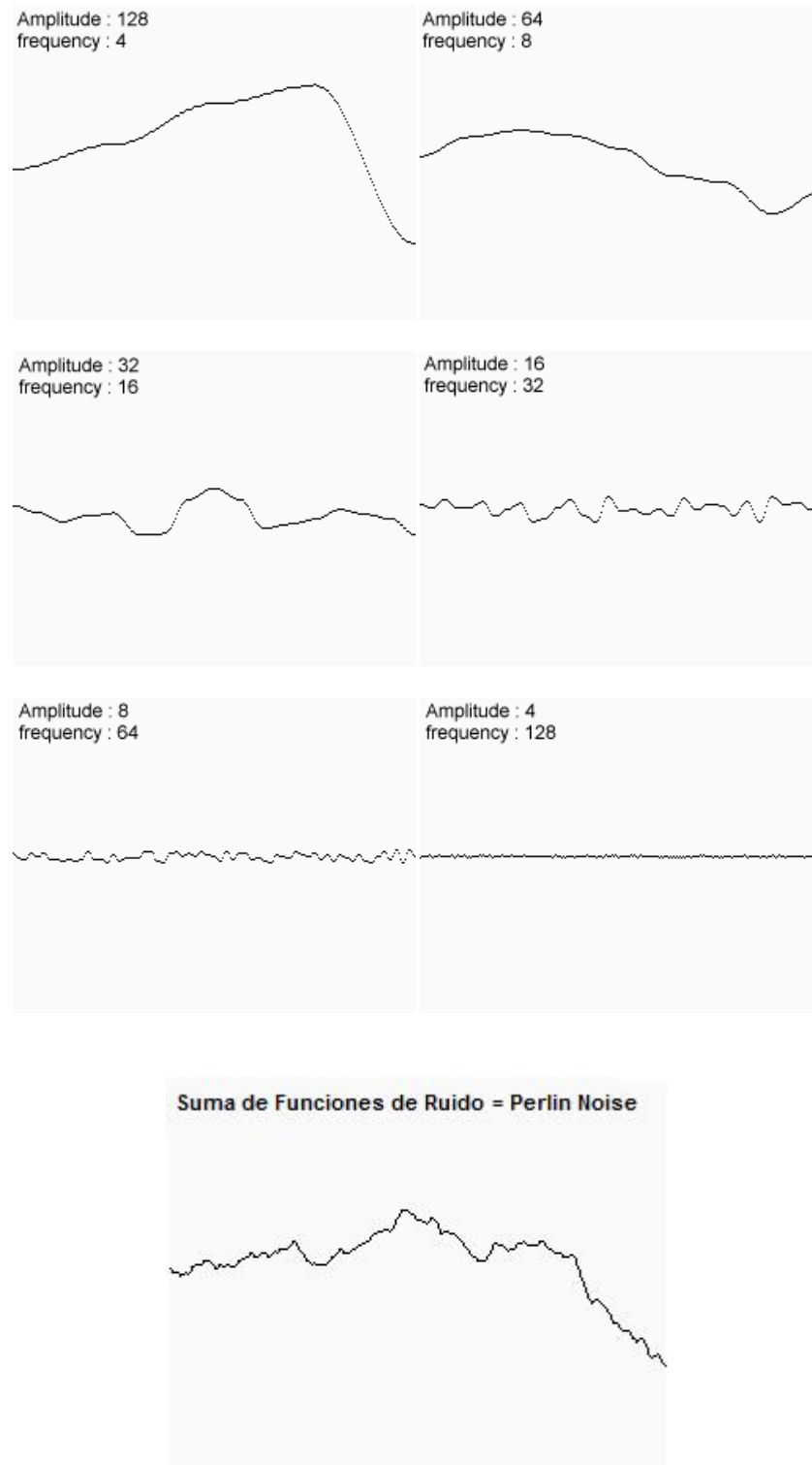


Figura 2.4 Obtención de la función de Ruido de Perlin. (Elias, 2003)

Sin embargo, se pueden crear funciones con ruido Perlin de diferentes características mediante el uso de otras frecuencias y amplitudes en cada paso. Para hacerlo más sencillo, y para evitar la repetición de las palabras de amplitud y frecuencia todo el tiempo, se utiliza un solo número para especificar la amplitud de cada frecuencia. Este valor es conocido como *persistencia*. Y se puede definir como el multiplicador que determina la rapidez con que las amplitudes para cada una de las octavas sucesivas disminuyen en la función de ruido Perlin. En términos matemáticos tenemos:

$$frecuencia = 2^i$$

$$amplitud = persistencia * (octava anterior)$$

En donde i es la i^{a} función de ruido que se añade. Y la amplitud es igual al producto del valor de la persistencia y la octava anterior calculada. Cada una de las sucesivas funciones que se añaden en la función de ruido se le conoce como una octava. La razón de esto es que cada función de ruido es el doble de la frecuencia de la anterior.

Para ilustrar el efecto de la persistencia en la salida del ruido Perlin, se presentan los resultados en la figura 2.5; en donde tenemos la frecuencia de la función, el valor de la amplitud con el factor de persistencia, y como resultado la función del ruido de Perlin. (Perlin, 1999)

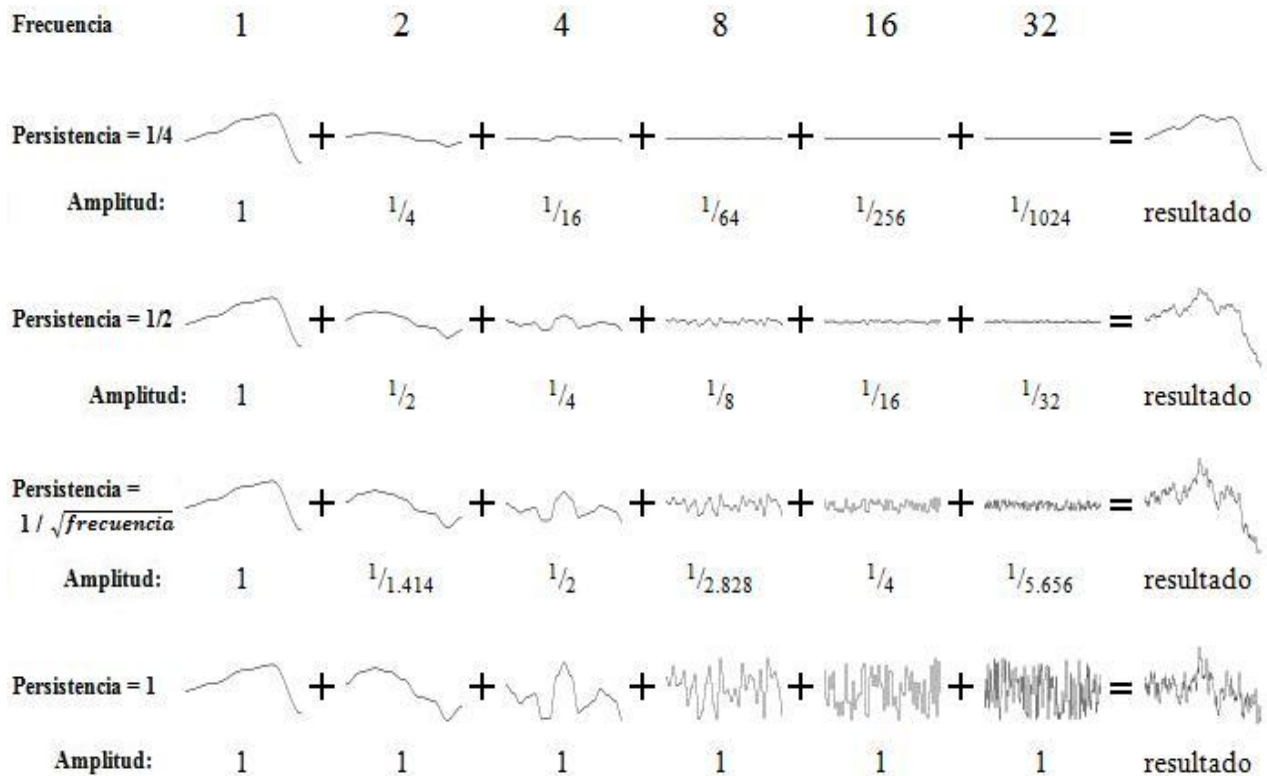


Figura 2.5 Diferentes resultados de la función de Ruido de Perlin con diferentes valores de persistencia. (Elias, 2003)

En la función de ruido de Perlin se aprecian grandes, medianas y pequeñas variaciones. Es por eso que muchos paisajes generados por computadora se hacen con este método.

Utilizando diferentes funciones aleatorias de generación de ruido en 2D, se obtienen los patrones que se muestran en la figura 2.6.

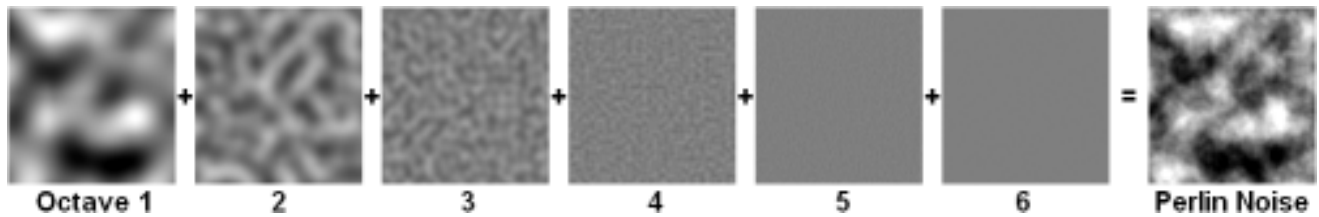


Figura 2.6 Imagen que muestra la combinación de 6 octavas para crear la función de ruido de Perlin 2D. (Perlin, 1999)

Interpolación

Después de haber creado la función de ruido, se tendrán que suavizar los valores que devuelve. Una función estándar para la interpolación se puede escribir de la siguiente manera: se deben tomar 3 valores de entrada (a , b , x), en donde $0 \leq x \leq 1$. La función de interpolación devuelve un valor entre a y b sobre la base del valor x . Es decir, cuando x es igual a 0, devuelve una a , cuando x es 1 devuelve b . Cuando x está entre 0 y 1 devuelve algún valor entre a y b , esta función también es conocida como interpolación lineal. En donde en términos de código quedaría de la siguiente forma:

```
funcion_Interpolacion_Lineal (a, b, x)
return a*(1-x) * b*x
fin de funcion
```

Integrando todo

La parte principal de la función de Perlin es generar un ciclo en donde cada iteración del ciclo añade otra octava de dos veces la frecuencia. Cada iteración pide una función diferente de ruido, denotado por Noise_i . Una vez obtenidos los valores estos deben ser interpolados para obtener una función suave, la cual podemos ocupar por ejemplo en la generación de texturas procedurales.

Se puede jugar tanto como se quiera, probando diferentes persistencias con diferentes frecuencias y diferentes dimensiones. Utilizar incluso una función de Perlin y afectar las propiedades de otra función, etc. (Elias, 2003)

Las siguientes texturas mostradas en la figura 2.7 se realizaron con base en la función de ruido de Perlin:

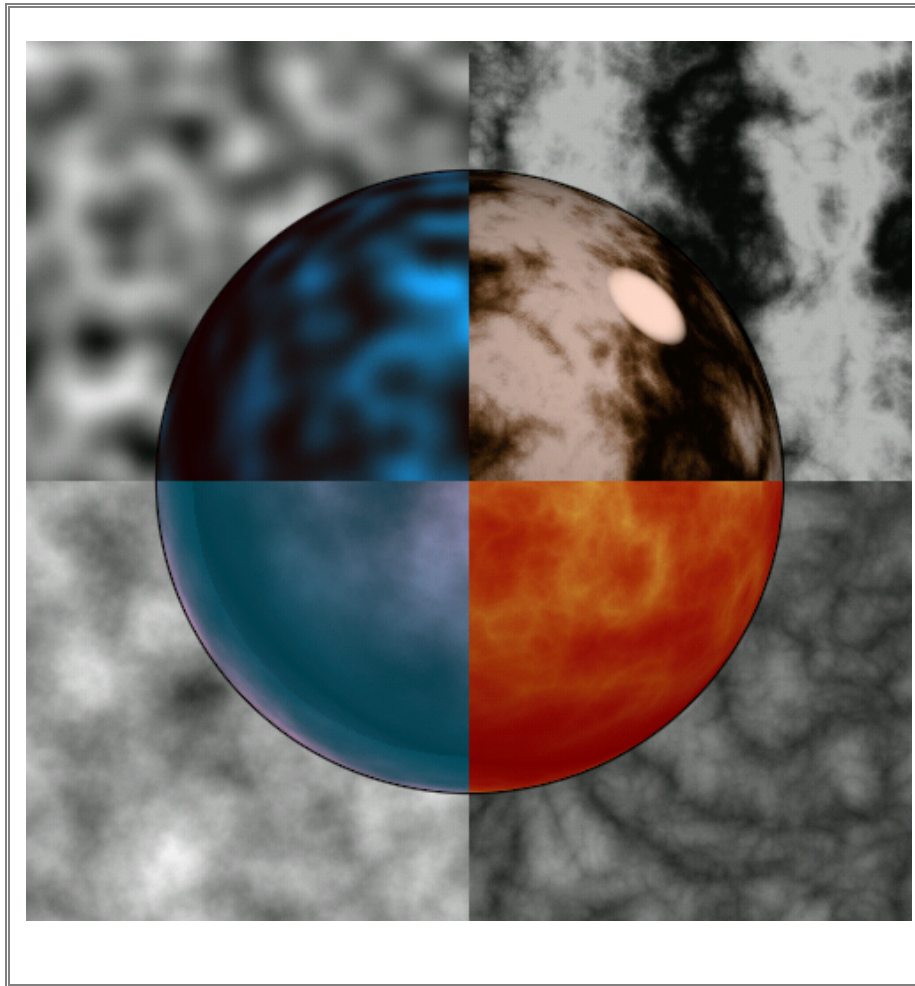


Figura 2.7 Texturas 3D generadas en base a la función de Ruido de Perlin. (Perlin, 1999)

2.2.3 Fractales

Hoy en día los fractales han llamado mucho la atención. Las imágenes que se obtienen con ellos son espectaculares, y se han desarrollado varios métodos para generarlos. El término fractal se refiere a cualquier cosa con gran cantidad de auto-similitud exacta o estadística, por tanto los únicos objetos fractales verdaderos son aquellos generados por procesos infinitamente recursivos.

La mejor forma de ilustrar lo que significa auto similitud es con el ejemplo de copo de nieve de von Koch. A partir de una línea recta con una protuberancia, reemplazamos cada segmento de línea con una figura exactamente igual a la línea original. Si repetimos este proceso un número infinito de veces, se dice que el resultado es auto similar: todo el objeto es similar a una sub-porción de él mismo como se ilustra en la figura 2.8. (Foley, et al., 1996)

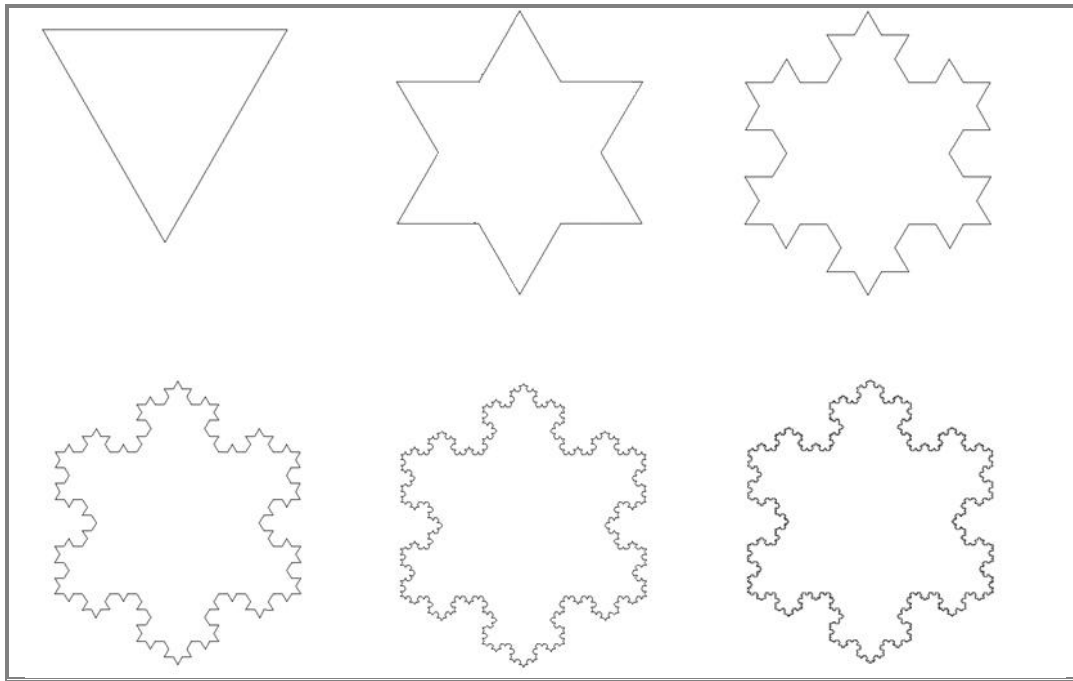


Figura 2.8 Concepto de la generación de fractal con la construcción del copo de nieve de von Koch. (Foley, et al., 1996 pág. 406)

2.2.3.1 Uso de fractales en la generación de texturas

Estos resultados son muy sugerentes para el modelado de formas naturales, ya que muchos objetos naturales parecen exhibir una gran auto-similitud. Las montañas tienen picos, picos más pequeños, rocas y gravas, todo lo cual parece similar; los árboles tienen troncos, ramas y tallos, que también se ven similares; las costas tienen bahías, caletas, estuarios, riberas y zanjas de drenado, que también son de apariencia similar. Si nos fijamos en muchas cosas en la naturaleza, nos daremos cuenta de que son fractales con diferentes niveles de detalle.

Así que el modelado de la auto-similitud a cierta escala parece ser una forma de generar modelos atractivos de fenómenos naturales. (Foley, et al., 1996)

Dado que para la construcción de fractales, se necesitan de varias operaciones repetitivas las cuales se pueden generar en base en algún algoritmo ejecutado por computadora. Logrando además con su construcción una complejidad visual potencialmente ilimitada que se obtiene relativamente de una pequeña cantidad de código. (Ebert, et al., 2002)

Los métodos fractales han probado ser útiles para modelar una amplia variedad de fenómenos naturales. En las aplicaciones de graficación por computadora son muy utilizados para modelar terrenos, nubes, agua, árboles y otras plantas, plumas, piel y distintas texturas de superficie, siendo muy eficientes en elaborar patrones atractivos. (Hearn, et al., 1995)

2.2.4 Conceptos básicos para la generación de texturas procedurales

Es importante mencionar que las texturas procedurales descritas en este apartado no tienen ninguna base en principios físicos o biológicos, más bien están realizadas básicamente por el cálculo de los algoritmos por medio de una computadora con el fin de obtener resultados visuales aceptables. (Ferguson, 2001)

Así que lo más conveniente es empezar a definir algunos de los conceptos básicos para la generación de la mayoría de texturas procedurales:

2.2.4.1 Exponente de Hurst (H)

Este parámetro en términos de matemáticas fractales, está directamente relacionada con la dimensión fractal⁹, lo que da una medida de la rugosidad en la superficie. La relación entre la dimensión fractal (D) y el exponente de Hurst (H) es:

$$D = 2 - H$$

Por lo tanto si tenemos $H=0.50$ entonces $D=1.50$. Un valor de $1 < H \leq 1.50$, resultara en una dimensión fractal más cercana a una línea. Esto daría como resultado una línea más suave con menos picos. Con un rango de $0 < H < 0.50$ arrojaría una dimensión fractal mayor, más puntiaguda. (Heinz-Otto Peitgen, 2004)

La figura 2.9 muestra los resultados para diferentes valores de H.

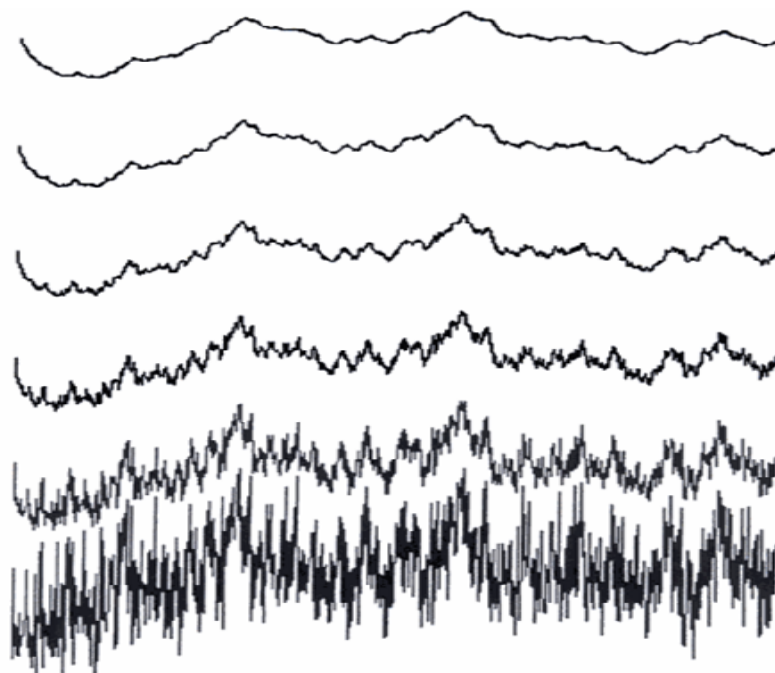


Figura 2.9 Resultados de la variación del parámetro H (exponente de Hurst), variando de 1.0 a 0 con incrementos de 0.2. (Ebert, et al., 2002 pág. 439)

⁹ **Dimensión fractal:** Se define como una medida cuantificable de las características de auto semejanza de las formas fractales. Por ejemplo, la dimensión fractal se ha utilizado para medir la rugosidad de las costas.

Este parámetro es utilizado para saber la afección rugosidad en nuestro patrón de texturas. (Ebert, et al., 2002)

2.2.4.2 Lacunaridad

La noción de “lacunaridad” fue propuesta por Mandelbrot en 1982, como una medida complementaria para determinar estructuras geométricas con la misma dimensión fractal, pero con diferente textura. La lacunaridad, que deriva del vocablo latino “laguna”, es una medida de la distribución de los espacios vacíos en una estructura geométrica. Por lo cual tiene que ver con la distribución del tamaño de los agujeros. Si un fractal tiene grandes lagunas o agujeros, el parámetro de *lacunaridad* es alto; por otro lado, si un fractal es casi invariante, su *lacunaridad es de un valor* bajo, como podemos apreciar en la figura 2.10. (Mandelbrot, 1997)

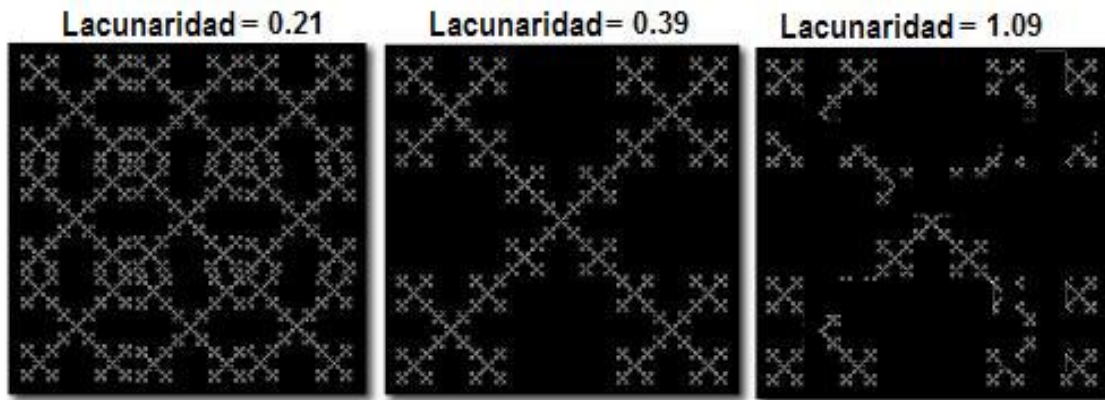


Figura 2.10 Imágenes de fractales en donde se aprecia el concepto de lacunaridad. (Turcotte, 1997 pág. 110)

En la generación de texturas procedurales se utiliza este término, para saber la afección de lagunas o huecos en nuestro patrón de texturas. (Ebert, et al, 2002)

2.2.4.3 Octavas

Los fractales pueden tener potencialmente ilimitados detalles. Pero esos detalles deben tener “límites” para que puedan ser generados por una computadora. De hecho en la naturaleza, los fractales “desaparecen” a cierta escala de nuestra vista, con lo cual se puede definir que tienen una banda limitada. Ese nivel de detalle en la generación de una textura procedural, nos lo proporcionan las octavas.

El valor de la octava nos proporciona el número de veces en que se tiene que hacer la iteración de la función, sumando en cada iteración una textura a escala del patrón original. Un valor grande en la octava agregará detalles más pequeños a la textura.

Sin embargo hay que tener control del número de veces que se va a repetir un ciclo, recordando que también la cantidad de detalle requerido en un punto dado de una imagen, depende de la distancia de la cámara, la resolución de la pantalla, el campo de visión, y otros factores. Además, un exceso de detalle no sólo incrementara el tiempo de cálculo, también puede causar aliasing. (Ebert, et al., 2002)

En la figura 2.11, podemos apreciar el concepto antes mencionado. Teniendo la función original sumada a $\frac{1}{2}$ y $\frac{1}{4}$ de la misma, para obtener una textura con más detalles.

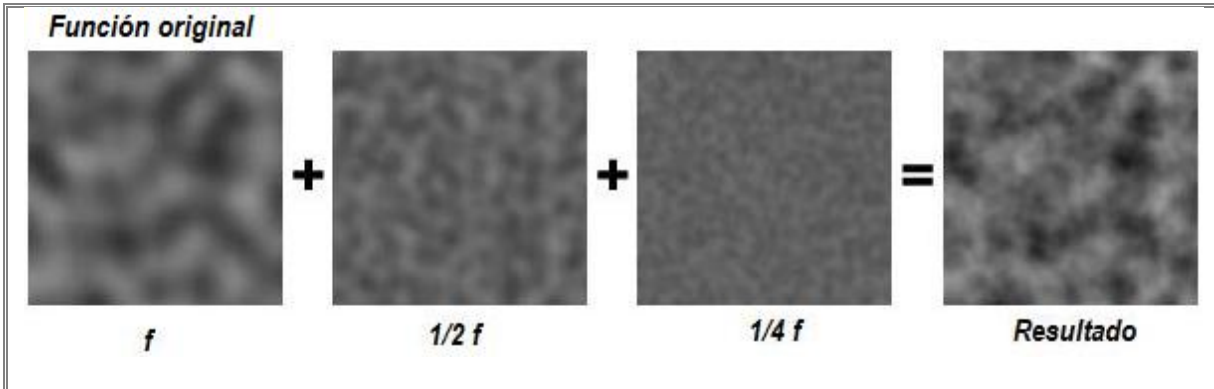


Figura 2.11 Ejemplo del concepto de octavas, teniendo la función f original sumada a $\frac{1}{2}$ y $\frac{1}{4}$ de la misma, dando como resultado una textura con más detalles. (Poissant, 2002)

2.2.4.4 Offset

Parámetro utilizado en modelos procedurales de terreno también conocidos como modelos multifractales, por contener altas y pequeñas variaciones en la superficie de una textura. Dicho parámetro se puede interpretar como un valor de altura para generar la textura de un terreno. (Texture, 2006)

2.2.4.5 Umbral

Se especifica como un valor umbral que se utiliza para determinar si la textura se debe mostrar o no. Si el valor en un punto de la textura procedural es superior al valor umbral, este modifica el valor final de la textura. Si el valor de la textura es inferior al valor umbral, este no altera el valor final de la textura. (Texture, 2006)

2.2.5 Patrones utilizados para generación de texturas procedurales

La construcción de texturas procedurales para este trabajo de tesis, se hizo bajo los parámetros antes mencionados. En primer lugar, se tiene una *función base* (función de ruido de Perlin), la cual es repetida en una variedad de escalas. Se cuenta también con parámetros que controlan la rugosidad (*exponente de Hurst*). Con *octavas* que nos marcarán el número de veces de la iteración en la construcción de la textura agregando más o menos detalles. Por último, tenemos el parámetro de *lagunaridad*, dicha propiedad cambia la frecuencia de la función base en de cada iteración agregando lagunas a la textura final generada. (Ebert, et al., 2002)

2.2.5.1 Función movimiento Browniano fraccional

La función movimiento Browniano fraccional, sus siglas en inglés (fBm), es generada a partir de varios ejemplares de ruido juntos, cada ejemplar tiene una frecuencia y amplitud

diferentes. Las frecuencias y amplitudes de las sucesivas adiciones de ruido están relacionadas por el factor de lacunaridad y persistencia, respectivamente. Se le considera una función "auto-similar" dado que es una copia de sí misma a diferentes escalas, ver figura 2.12.

Esta función tiene un aspecto visual agradable, complejo y natural que imita el patrón de muchas texturas en la naturaleza. (Apodaca, et al., 1999)

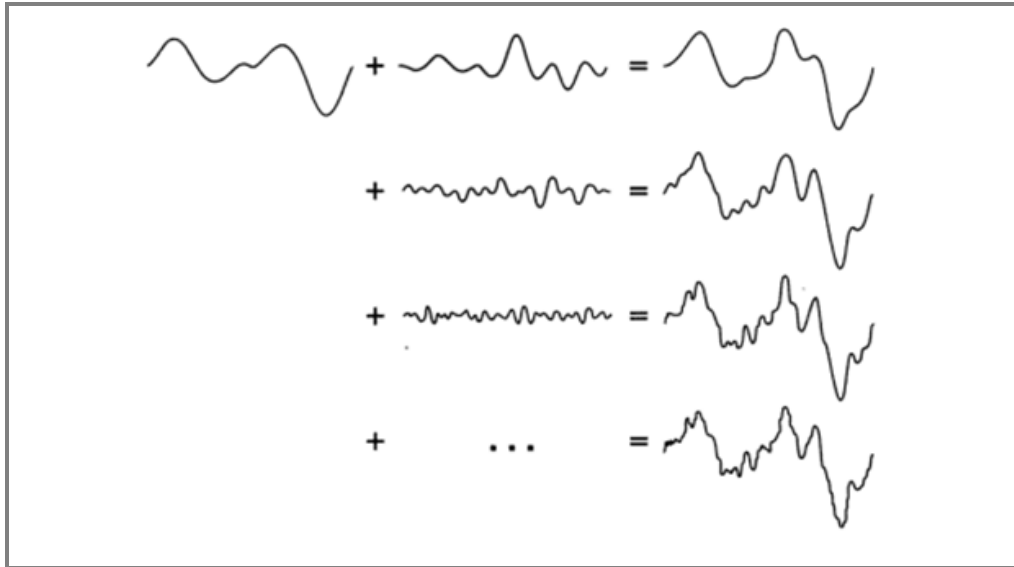


Figura 2.12 Esquema de la función movimiento Browniano fraccional (fBm). (Apodaca, et al., 1999 pág. 252)

A continuación se presenta el pseudocódigo de generación de dicha función a fin de comprender los parámetros necesarios para la generación de esta textura. (Ebert, et al., 2002 pág. 437)

```

/*
 * Procedural fBm evaluated at "point".
 *
 * Parameters:
 * "H" is the fractal increment parameter
 * "lacunarity" is the gap between successive frequencies
 * "octaves" is the number of frequencies in the fBm
 */
double fBm( Vector point, double H, double lacunarity, double octaves )
{
    double value, remainder, Noise();
    int i;
    value = 0.0;
    /* inner loop of fractal construction */
    for (i=0; i<octaves; i++) {
        value += Noise( point ) * pow( lacunarity, -H*i );
        point *= lacunarity;
    }
    remainder = octaves - (int)octaves;
    if ( remainder ) /* add in "octaves" remainder */
        /* 'i' and spatial freq. are preset in loop above */
        value += remainder * Noise3( point ) * pow( lacunarity, -H*i );
    return value;
}

```

En la figura 2.13 se muestran dos cubos texturizados con la función movimiento Browniano fraccional (fBm).

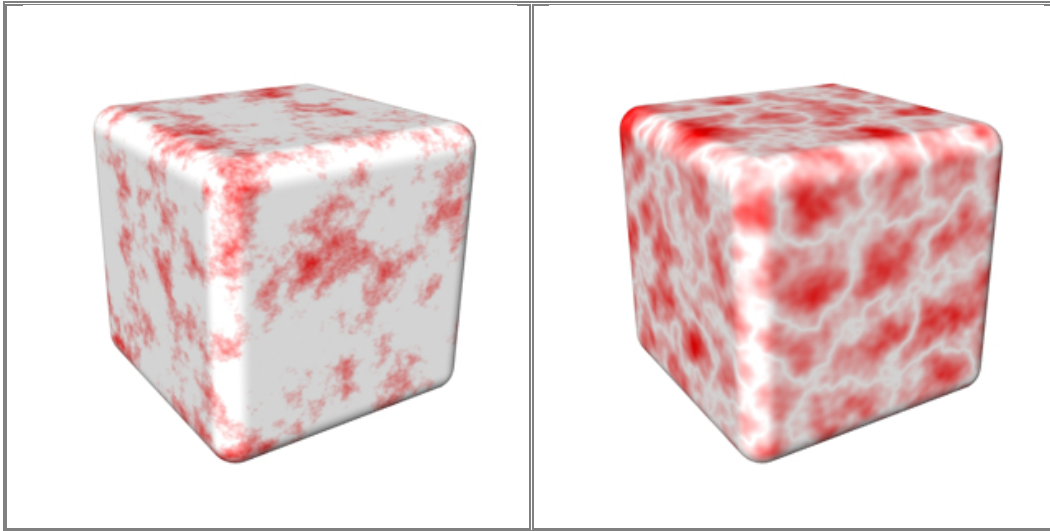


Figura 2.13 Cubos texturizados a base de la función movimiento Browniano fraccional (fBm). (*Texture, 2006*).

2.2.5.2 Función de turbulencia

Un familiar de la función fBm es la función de Perlin llamada turbulencia. La única diferencia entre la turbulencia y la de fBm es el valor absoluto. El resultado es de una apariencia más "ondulante". (*Apodaca, et al., 1999*)

Esta textura combina capas de ruido fractal con diferentes frecuencias para crear complejos patrones detallados e interesantes. Estableciendo una mayor gama de efectos en términos de alteración de los detalles. (*Van, 2004*)

A continuación se muestra el pseudocódigo de la función de turbulencia para la generación de texturas. (*Ebert, et al., 2002* pág. 86) Para crear este tipo de textura, dado que la función tiene componentes positivos y negativos, solo toma su valor absoluto.

```
float
turbulence(point Q)
{
    float value = 0;
    for (f = MINFREQ; f < MAXFREQ; f *= 2)
        value += abs(snoise(Q * f))/f;
    return value;
}
```

En la figura 2.14 se muestran dos cubos texturizados con la función de turbulencia.

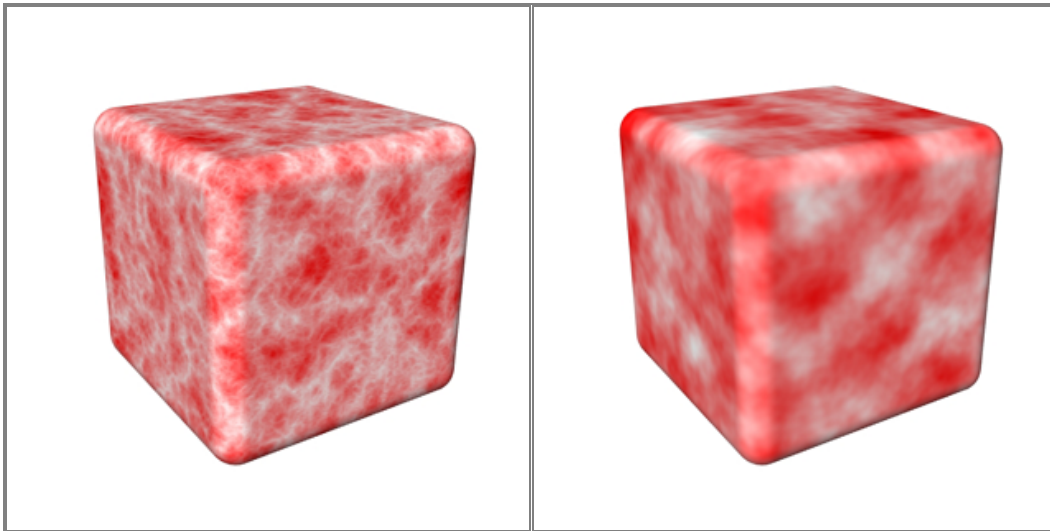


Figura 2.14 Cubos texturizados a base de la función Turbulencia. (Texture, 2006).

2.2.5.3 Función de terreno heterogéneo

La función de terreno heterogéneo es conocida como un multifractal. Un multifractal puede ser definido como un fractal que contiene una multiplicidad de medidas, es decir que está formado heterogéneamente y no tienen los mismos “valores” de medidas en todas partes. Así una imagen o textura multifractal podrá tener variaciones a lo largo de la textura, por ejemplo en su dimensión fractal. (Ebert, et al., 2002)

En la generación de texturas procedurales la función de terreno heterogéneo generará una apariencia heterogénea en la imagen final obtenida; controlando la cantidad de detalles añadidos de acuerdo a un valor de "offset". Cuando el parámetro offset sea igual a cero la función será heterogénea en extremo, y conforme el valor del offset vaya aumentando la función va ir cambiando de multifractal a monofractal, acercándose a una textura de superficie plana. Dicha función es muy útil para la creación de paisajes (sobre todo planetas), ya que simula la forma en que la tierra tiende a ser más plana y suave en las zonas bajas, y áspera y desigual en el aumento de altura. (Van, 2004)

El pseudocódigo para la generación de la función procedural de terreno heterogéneo, se muestra a continuación. (Ebert, et al., 2002 pág. 500)

```
/*
 * Heterogeneous procedural terrain function: stats by altitude method.
 * Evaluated at "point"; returns value stored in "value".
 *
 * Parameters:
 * "H" determines the fractal dimension of the roughest areas
 * "lacunarity" is the gap between successive frequencies
 * "octaves" is the number of frequencies in the fBm
```

```

* "offset" raises the terrain from "sea level" */
double
Hetero_Terrain( Vector point,
                double H, double lacunarity, double octaves, double offset )
{
    double value, increment, frequency, remainder, Noise3();
    int i;
    static int first = TRUE;
    static double *exponent_array;

    /* precompute and store spectral weights, for efficiency */
    if ( first ) {
        /* seize required memory for exponent_array */
        exponent_array =(double *)malloc( (octaves+1) * sizeof(double) );

        frequency = 1.0;
        for (i=0; i<=octaves; i++) {
            /* compute weight for each frequency */
            exponent_array[i] = pow( frequency, -H );
            frequency *= lacunarity;
        }
        first = FALSE;
    }

    /* first unscaled octave of function; later octaves are scaled */
    value = offset + Noise3( point ); point.x *= lacunarity;
    point.y *= lacunarity; point.z *= lacunarity;

    /* spectral construction inner loop, where the fractal is built */
    for (l=1; l<octaves; l++) {
        /* obtain displaced noise value */
        increment = Noise3( point ) + offset;

        /* scale amplitude appropriately for this frequency */
        increment *= exponent_array[l];

        /* scale increment by current "altitude" of function */
        increment *= value;

        /* add increment to "value" */
        value += increment;

        /* raise spatial frequency */
        point.x *= lacunarity;
        point.y *= lacunarity;
        point.z *= lacunarity;
    } /* for */

    /* take care of remainder in "octaves" */
    remainder = octaves - (int)octaves;
    if ( remainder ) {
        /* "i" and spatial freq. are preset in loop above */
        /* note that the main loop code is made shorter here */
        /* you may want to make that loop more like this */
        increment = (Noise3( point ) + offset) * exponent_array[l];
        value += remainder * increment * value;
    }
    return( value );
} /* Hetero_Terrain() */

```

En la figura 2.15 se muestran un cubo texturizado con la función de terreno heterogéneo.

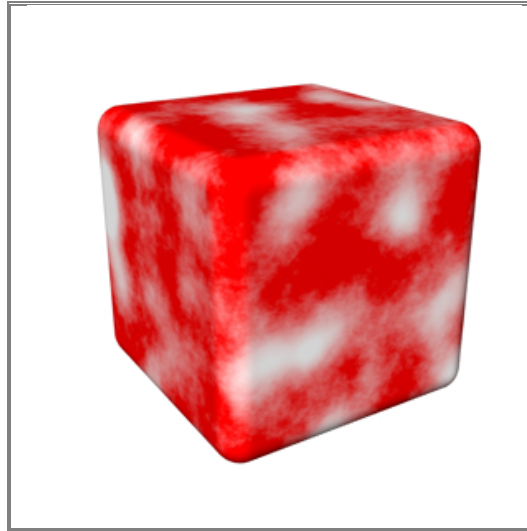


Figura 2.15 Cubo texturizado con la función de terreno heterogéneo. (Texture, 2006).

2.3 Aliasing

Aliasing es un término del ámbito de procesamiento de señales. Se refiere a una variedad de defectos desagradables en la reconstrucción de una señal digital como consecuencia de una limitada toma de muestreo, como se aprecia en la figura 2.16.

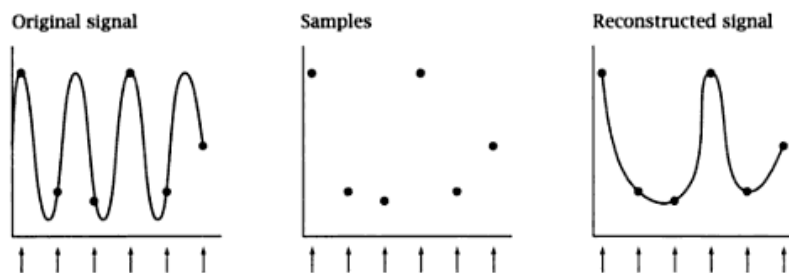


Figura 2.16 Muestro de una señal, en donde la reconstrucción de la señal presenta problemas de aliasing.

La cantidad de información en la señal original se le llama ancho de banda. La cantidad de información que puede ser capturado por las muestras depende de la frecuencia de muestreo (número de puntos de muestreo por unidad de distancia).

En la graficación por computadora, aliasing se refiere a una variedad de alteraciones y artefactos desagradable en una imagen que resulta de la inadecuada toma del muestreo. Que desafortunadamente las condiciones para la correcta toma de muestras y la reconstrucción no son siempre fáciles de cumplir, y cuando no se cumplen, se produce aliasing.

El aliasing visualmente se observará en una imagen con curvas y líneas inclinadas en una pantalla, pero que debido a la resolución de ésta, resulta incapaz de representar la curva como tal, y por tanto dichas curvas se muestran en la pantalla con el efecto visual tipo "sierra" o "escalón" no deseados.

Para evitar el aliasing es necesario contar con un proceso el cual nos ayude a minimizarlo, a este proceso se le conoce como antialiasing. (Ebert, et al., 2002)

2.3.1 Antialiasing

Para mejorar la distorsión de la información como consecuencia del muestreo se deben aplicar métodos de creación antialiasing que compensen el proceso del muestreo. Ya que en todos los casos, el sistema de muestreo determina la calidad potencial de la imagen que se está captando y los medios que habrá que emplear para corregir los defectos.

Con el fin de evitar la pérdida de información, se necesita establecer la frecuencia de muestreo como mínimo del doble de la frecuencia más alta que presenta la señal, a la cual se denomina frecuencia del muestreo de nyquist: f_s .

$$f_s = 2f_{máx}$$

Otra manera de expresar esto es que el intervalo de muestreo no debe ser mayor que la mitad del intervalo del ciclo (intervalo de muestreo de nyquist).

$$\Delta x_s = \frac{\Delta x_{ciclo}}{2}$$

Donde $\Delta x_{ciclo} = 1 / f_{máx}$. (Hearn, et al., 1995 págs. 178-179)

Desafortunadamente, siempre hay un límite práctico en la resolución de una imagen debido al espacio de memoria o a las limitaciones de pantalla, y la tasa de muestreo de una imagen es proporcional a su resolución. Por lo cual es imposible que una imagen muestre demasiado pequeños detalles para ser visible en determinadas resoluciones.

Hay otra razón por la cual el aumento de la tasa de muestreo no es nunca una solución completa para el problema de aliasing. Algunas señales tienen ancho de banda ilimitado, por lo que es difícil obtener una frecuencia máxima. Los cambios bruscos en la señal tienen componentes de alta frecuencia de manera arbitraria. No importa cuán grande sea la resolución de la imagen, el aumento de la tasa de muestreo a un valor finito no puede eliminar el aliasing de muestreo. Esta es la razón de porque líneas inclinadas son dentadas incluso en los muestra de mayor resolución.

En ocasiones el aliasing no siempre puede resolverse mediante el aumento de la tasa de muestreo, estamos obligados a encontrar la manera de eliminar las altas frecuencias de

la señal antes del muestreo. La técnica se denomina filtro paso-bajas, ya que pasa la información de baja frecuencia mientras que elimina la de mayor frecuencia. El efecto visual de un filtrado paso-bajas es difuminar la imagen. El reto consiste en borrar la imagen lo menos posible, al mismo tiempo de atenuar las frecuencias altas no deseadas. (Ebert, et al., 2002)

2.4 Atributos de las superficies

La elección de los atributos visuales es esencial para una representación exitosa de una superficie. En la construcción de las texturas procedurales utilicé algunos de estos atributos para obtener la apariencia deseada.

2.4.1 Color

Utilizamos el color por estética, para establecer un ambiente o estado de ánimo por cuestiones de realismo, como realce, para identificar la relación entre áreas.

Muchos colores son producidos como resultado de una combinación entre dos o más colores primarios¹⁰, o a través de añadiduras o mezclas.

Desde una perspectiva de la ciencia, el color es la longitud de onda dominante reflejada por un objeto de color. Es decir, el color es el producto de las longitudes de onda que son reflejadas o absorbidas por la superficie de un objeto, que al ser captadas a través de nuestros ojos y que luego son transmitidas al cerebro, quien procesa para darnos una apariencia de un cierto rango de color. (Capilla, 2002)

2.4.2 Uso del color

El color se utiliza en las texturas de dos maneras. En primer lugar, porque simplemente el objeto está texturizado particularmente de ese color en la realidad. Por ejemplo, se puede crear una textura de color azul brillante para cubrir la piel de un ser humano pero la piel humana en la realidad no es azul. Por lo cual se tendrá que utilizar tonos parecidos a la piel. Así que se trata de buscar semejanza en colores con el objetivo de tener realismo.

El segundo uso del color es más subjetivo y artístico. Este entra en juego cuando se tiene la libertad artística para decidir sobre los colores utilizados para determinadas sensaciones. Este uso artístico del color también puede aplicarse a grupos de colores en la vida real, sutilmente alterando las percepciones de las personas hacia las cosas, incluso si el conjunto de colores que ha utilizado en las texturas se han creado simplemente como una recreación de cómo ese objeto se parece en la realidad. Un ejemplo de esto sería si tal vez se

¹⁰ **Colores primarios:** Son aquellos colores que no pueden obtenerse mediante la mezcla de ningún otro por lo que se consideran únicos. Los tres colores que cumplen con esta característica son: amarillo, rojo y azul. Mezclando pigmentos de éstos colores pueden obtenerse todos los demás colores.

quiere texturizar a un personaje maligno, donde es más probable el uso de colores de tonos oscuros, tales como: azules, púrpura o incluso pardos o negros para indicar una sombra más oscura de personalidad.

Por otro lado, si se quisiera obtener un personaje más angelical, se trabajaría más con una serie de colores deslumbrantes, brillantes, como el blanco y oro. Este tipo de uso del color es muy subjetivo, y puede ayudar a transmitir ideas o percepciones al usuario. (Van, 2004)

2.4.3 Matiz, Saturación y Brillo

Cuando se trabaja con el color, es vital comprender la forma de manipularlo para que lo podamos ajustar a nuestras necesidades. Así cuando se trabaja con imágenes en color, se tiene que tener una buena comprensión de los distintos componentes a fin de que se puedan manipular según sea necesario. (Van, 2004)

Estos componentes se puede dividir en: matiz, saturación y brillo.

Matiz

El matiz es una de las principales propiedades de un color, que se describe con nombres tales como: rojo, amarillo, azul, etc. En términos de graficación por computadora se conoce al matiz básicamente como el color de los píxeles de una imagen.

Así cuando se trabaja con diferentes imágenes que se desean mezclar, se adaptan los colores de las imágenes para llegar dar la tonalidad de la mezcla hecha. Se puede además hacer ajustes a las imágenes con lo cual pueden cambiar drásticamente los colores de la imagen, dándole el control sobre la apariencia de la misma. (Van, 2004)

Saturación

Se conoce como saturación a la cantidad de intensidad de color. Es decir la pureza del color. En mayor o menor grado de mezcla de colores las cuales modifican a un color fundamental, hace que un color esté más o menos saturado. Hablando en términos de graficación por computadora se sabe que la saturación de una imagen es la cantidad de color que contiene cada píxel y la intensidad de ese color. Si se quita toda la saturación de una imagen, esta se convierte en una imagen en escala de grises. Cuanto más alto sea el contenido gris, menor será la saturación. Es por eso que se debe utilizar correctamente la saturación ya que un problema muy común es la sobresaturación¹¹ de texturas, presentando un efecto desagradable en nuestra imagen. (Van, 2004)

Brillo

Es el grado de claridad y oscuridad de un color. También se lo conoce como el atributo de la percepción visual en el que una fuente parece ser reflejo de la luz o radiación. Más fácil de entender con el ejemplo de la diferencia entre una imagen luminosa y una imagen oscura. (Van, 2004)

¹¹ **Sobresaturación:** Imágenes que utilizan niveles de saturación extremos, es decir, imagen con un resultado de colores irreales o tonalidades exageradas.

2.5 Espacio de color en software grafico

Dado que los seres humanos tenemos tres tipos de sensores (conos de la retina) activos para detectar los altos niveles de iluminación. Las señales de estos tres tipos de sensores determinan la respuesta de color del observador. Por esta razón, el color natural es un fenómeno tridimensional. Para describir cuantitativamente el color se necesita usar un bien definido sistema de coordenadas en un espacio tridimensional. En los gráficos por computadora se suele utilizar los colores *rojo*, *verde* y *azul* (RGB, por sus siglas en inglés) para proporcionar tal sistema de coordenadas. Sin embargo, son infinitamente muchos de esos sistemas de coordenadas que podrían aplicarse al espacio, y ninguno de ellos es intrínsecamente superior a cualquier otro sistema. Por lo tanto, cuando tratamos con los colores, existen otros formatos o espacios de color, como HSV, LUV y CMY, que se definen con tres sistemas de coordenadas llamados ejes. (Shirley, et al., 2005)

2.5.1 Modelo de color RGB

El modelo de colores RGB emplea un sistema de coordenadas cartesianas. El cual se puede representar con un cubo unitario que se define con los ejes de R, G y B, como lo ilustra la figura 2.17. El origen representa el negro y el vértice de las coordenadas (1, 1, 1) es el blanco. Los vértices del cubo en los ejes representan los colores primarios y los vértices restantes representan el color complementario para cada uno de los colores primarios.

El esquema de color RGB es un modelo aditivo, es decir, se suman las intensidades de los colores primarios para producir otros colores. Cada punto de color en las fronteras del cubo se puede representar como el conjunto de tres coordenadas (R,G,B), donde a estas coordenadas se les asignan los valores en el rango de 0 a 1. (Hearn, et al., 1995)

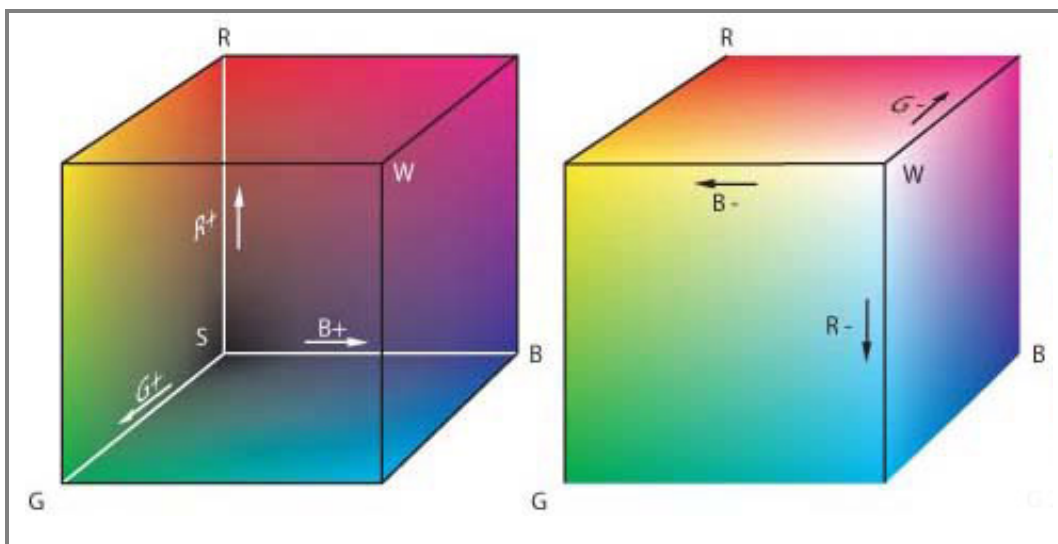


Figura 2.17 Modelo de color RGB, el cual define los colores con un proceso aditivo en el cubo unitario. (Hearn, et al., 1995 pág. 603)

2.5.2 Modelo de color HSV

En vez de un conjunto de colores primarios, el modelo HSV utiliza descripciones de color que tienen una aplicación más intuitiva por parte del usuario. Para dar una especificación de color, un usuario selecciona un color espectral y las cantidades de blanco y negro que se deben agregar para obtener diferentes sombras, tintes y tonos. Los parámetros de color en este modelo son matiz (H), saturación (S) y valor (V).

La representación tridimensional del modelo HSV se deriva del cubo RGB. Si se imagina que se ve el cubo a lo largo de la diagonal del vértice blanco al origen (negro), con un contorno hexagonal. La frontera del hexágono representa los diversos matices y se utiliza como la parte superior del hexácono HSV de la figura 2.18. En el hexácono, se mide la saturación a lo largo de un eje horizontal y el valor se da a lo largo de un eje vertical a través del centro del hexácono. (Hearn, et al., 1995)

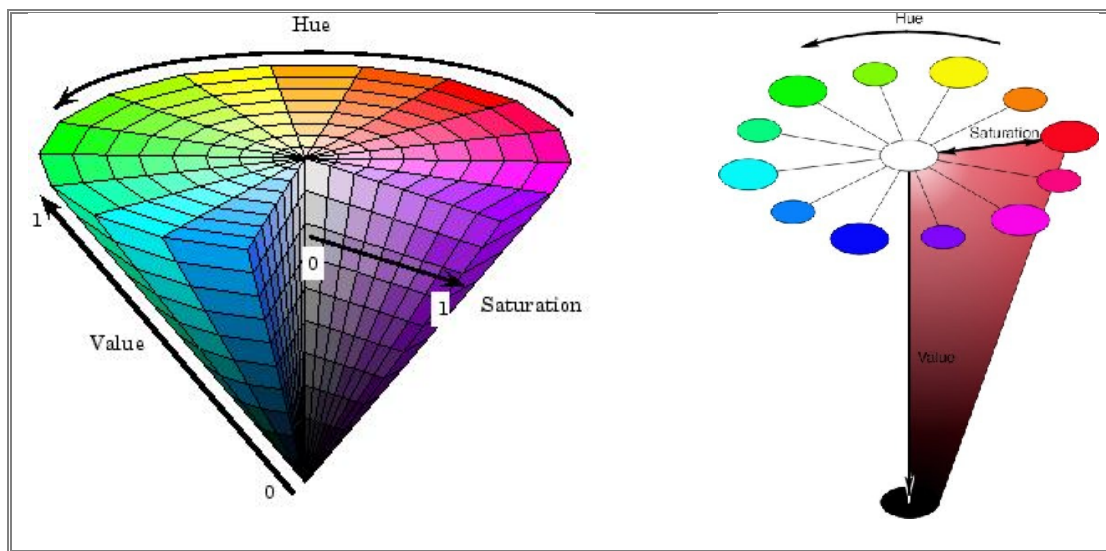


Figura 2.18 Modelo HSV representado por un hexácono. (Hearn, et al., 1995 pág. 607)

El matiz (H) se representa con respecto del eje vertical, en el intervalo 0° en el rojo a 360° . Los vértices en el hexágono están separados en intervalos de 60° . El amarillo está en 60° , el verde a 120° y el cian opuesto al rojo en $H=180^\circ$. Los colores complementarios están a una separación de 180° , ver figura 2.19 para comprender mejor el concepto.



Figura 2.19 Imagen en donde se visualiza el matiz del modelo de color HSV. (Hearn, et al., 1995 pág. 607)

La saturación (S) varía de 0 a 1 y se representa en este modelo como la razón de pureza de un matiz seleccionado en su pureza máxima en $S=1$. Se dice que un matiz seleccionado tiene una pureza de un cuarto con el valor de $S=0.25$. Con $S=0$, tenemos la escala gris.

El valor (V) varía de 0 en el pico del hexágono a 1 en la parte superior. El pico representa el negro. Los colores tienen su intensidad máxima en la parte superior del hexágono. Cuando $V=1$ y $S=1$, tenemos matices “puros”. El blanco es el punto en $V=1$ y $S=0$. En la figura 2.20 podemos visualizar el efecto que produce el cambio de saturación y valor. (Hearn, et al., 1995)

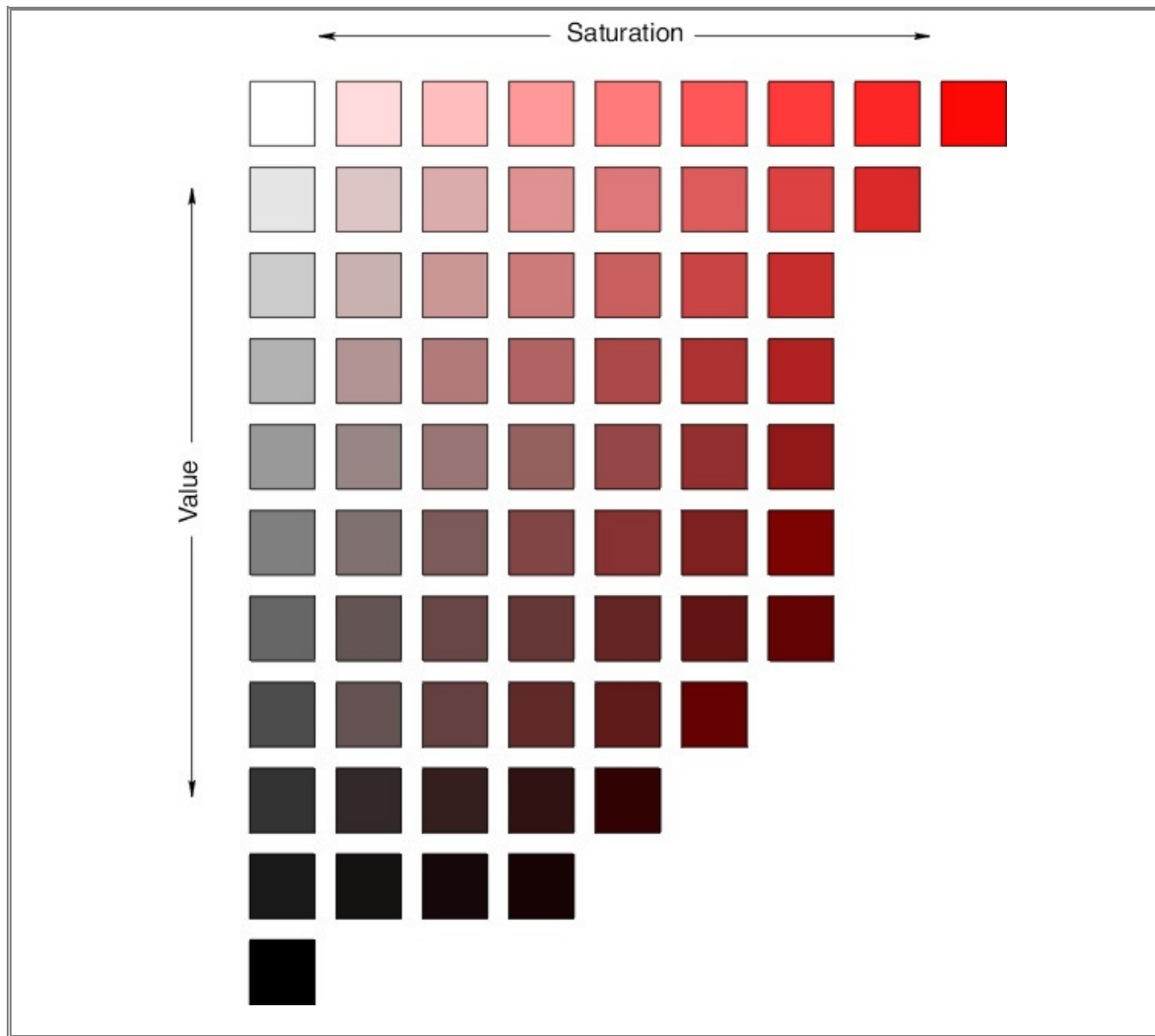


Figura 2.20 Resultado de variaciones de saturación y valor (brillo) del modelo HSV. (Capilla, 2002 pág. 114)

El ojo humano puede distinguir alrededor de 128 matices distintos y más o menos 130 tintes diferentes (niveles de saturación). Para cada uno de éstos, se puede detectar un número de sombras (especificaciones de valores), dependiendo del matiz que se selecciona. (Hearn, et al., 1995)

2.5.3 Conversión entre modelos HSV y RGB

La conversión permite a los programadores "pensar" con HSV ya que dicho modelo está más orientado a entender los matices, tonos y brillo al usuario, mientras que con el modelo RGB se dan las instrucciones a la máquina con la finalidad de ser visualizadas en un monitor de color.

Para determinar las operaciones necesarias en esta transformación, primero consideramos como se puede derivar el hexácono HSV del cubo RGB. La diagonal de este cubo del origen (negro) al blanco corresponde a un área de corte transversal hexagonal del hexácono. En cualquier corte transversal, todos los lados del hexágono y todas las líneas radiales desde el eje V hasta cualquier vértice tienen el valor V. Para cualquier conjunto de valores RGB, V equivale al valor máximo en este conjunto. El punto HSV que corresponde al conjunto de valores RGB se encuentra en el corte transversal hexagonal en el valor V. De este modo el parámetro S se determina como la distancia relativa de este punto desde el eje V. El parámetro H se determina al calcular la posición relativa del punto entre cada sextante del hexágono (ver figura 2.21). (Foley, et al., 1996 págs. 464-471)

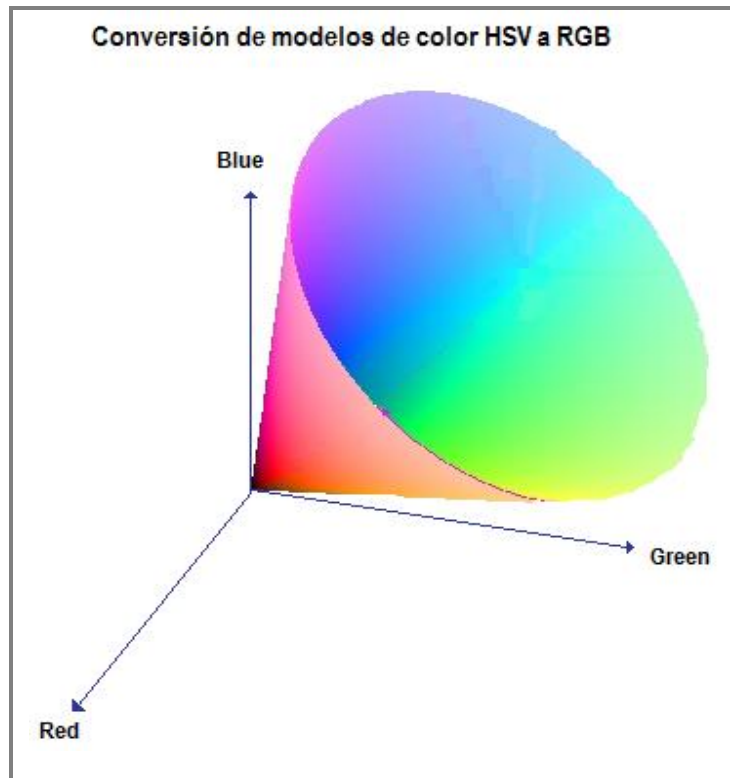


Figura 2.21 Imagen representativa de la conversión de modelos de color HSV y RGB. (Foley, et al., 1996 pág. 470)

Existen varios algoritmos para realizar esta conversión. Dado que no es el objetivo principal de este trabajo, no se profundizara más en este tema.

3. SÍNTESIS DE TEXTURAS PROCEDURALES PARA UN SIMULADOR VIRTUAL DE CIRUGÍA

En este capítulo se explicará la programación, desarrollo, generación e integración de la síntesis de texturas procedurales, para el simulador de entrenamiento de resección transuretral de próstata. Las cuales están divididas principalmente en:

- textura del tejido de la uretra (tejido con vascularidades)
- textura del tejido interno de la próstata (apariencia amarillo-pardo)
- textura del tejido de la cápsula de la próstata (apariencia rugosa)

Se realizó primero el estudio de las imágenes médicas reales, del procedimiento RTU, obtenidas de un video médico (ver capítulo 1, sección 1.5). Posteriormente con ayuda de funciones primitivas de generación de texturas procedurales, se simuló la apariencia visual de las imágenes reales.

3.1 Funciones básicas para la generación de texturas procedurales

Se decidió utilizar la generación de texturas procedurales porque nos ayudan a mejorar y obtener las texturas que se necesita para el simulador de cirugía, tomando en cuanto los siguientes puntos:

- El ojo humano es muy perceptivo de los artefactos repetitivos que pueden ser causados por un mapeo 2D en una superficie 3D. Ya que la mayoría de los objetos en las escenas digitales no son puramente planos. Asignar intrínsecamente una textura 2D a una superficie en 3D puede ser una tarea difícil, dando lugar a distorsiones y costuras visibles. (*Jagnow, 2004*)
- Una técnica para lograr una variación espacial de las texturas de los materiales en tres dimensiones sin lugar a distorsiones o costuras visibles, es generarlas a partir de una textura sólida, ofreciendo así una representación más natural. (*Jagnow, 2004*)
- Las texturas sólidas son vitales para mantener una apariencia constante realista y detallada, ya que los objetos “aparecerán” esculpidos en una sustancia sólida. (*Ebert, et al., 2002*)

El uso de métodos procedurales para la generación de texturas en este trabajo de tesis, nos ayudará a lograr una apariencia y aspecto visual que se requieren, logrando con ello obtener una textura continua en la cual se podrá modelar la glándula prostática; teniendo además la ventaja de contar con una textura que nos dará la apariencia de profundidad al realizar los cortes sobre el tejido, apareciendo las condiciones visuales de las texturas que se encuentran en su interior, que al final el usuario lo visualizará como se aprecia en el procedimiento real.

Se tendrá además la posibilidad de contar con las funciones procedurales base, las cuales nos ayudaran a extender el resultado a imagen más grandes, logrando con ello generar un modelo de síntesis de textura útil, visto y programable para el simulador virtual de cirugía.

3.1.1 Funciones procedurales básicas

El desarrollo de los algoritmos y las funciones básicas para la generación de texturas, se tomaron principalmente del libro *Texturing and Modeling a Procedural Approach*. (*Ebert, et al., 2002*)

Obteniendo las siguientes funciones procedurales que se explicarán y mostrarán a continuación.

3.1.1.1 Función de ruido

Para generar texturas procedurales irregulares lo primero que necesitamos es contar con una función primitiva irregular normalmente llamada *ruido*.

Las propiedades de una función de ruido ideal son:

- Ser una función pseudo-repetible en sus entradas.
- Tener un rango conocido, que normalmente vaya de -1 a 1.
- Tener una banda limitada, con una frecuencia máxima de aproximadamente 1.
- No presentar periodicidad evidente o patrones regulares. Tales funciones pseudo-aleatorias son siempre periódicas, pero la periodicidad puede ser muy larga y por lo tanto no es visible.
- Ser una función estacionaria, es decir, su carácter estadístico debe ser de traslación invariante. (Ebert, et al., 2002 pág. 68)

A continuación se presenta parte del código fuente de la función de ruido de Perlin¹², útil en la generación de texturas procedurales, el cual visualmente agrega a la textura una apariencia más compleja.

```
/* coherent noise function 2 dimension */
/* (copyright Ken Perlin) */

float noise2(float vec[2])
{
    int bx0, bx1, by0, by1, b00, b10, b01, b11;
    float rx0, rx1, ry0, ry1, *q, sx, sy, a, b, t, u, v;
    register i, j;

    if (start) {
        start = 0;
        init();
    }

    setup(0, bx0,bx1, rx0,rx1);
    setup(1, by0,by1, ry0,ry1);

    i = p[ bx0 ];
    j = p[ bx1 ];

    b00 = p[ i + by0 ];
    b10 = p[ j + by0 ];
    b01 = p[ i + by1 ];
    b11 = p[ j + by1 ];

    sx = s_curve(rx0);
    sy = s_curve(ry0);

#define at2(rx,ry) ( rx * q[0] + ry * q[1] )
```

¹² **Código fuente de la función de ruido de Perlin:** Para consultar el código completo se puede visitar la siguiente referencia web. (*Noise Perlin, 1997*)

```
q = g2[ b00 ] ; u = at2(rx0,ry0);
q = g2[ b10 ] ; v = at2(rx1,ry0);
a = lerp(sx, u, v);

q = g2[ b01 ] ; u = at2(rx0,ry1);
q = g2[ b11 ] ; v = at2(rx1,ry1);
b = lerp(sx, u, v);

return lerp(sy, a, b);
}
```

La función de ruido de Perlin se genera de manera uniforme con gradientes de unidad distribuida. Su método es la generación de vectores con componentes entre -1 y 1. Estos vectores se encuentran dentro del cubo que delimita el ámbito de la unidad. Si algún vector cuya longitud es superior a 1 es descartado.

La rutina anterior genera el valor del ruido de gradiente para un solo punto con coordenadas en (x, y). Con la combinación de ocho valores, se realiza una interpolación suavizada trilineal, con el cual se obtiene el valor de ruido de gradiente. Las interpolaciones lineales son controlados por una función de escalón suavizado de las partes fraccionales de las coordenadas de entrada.

En la figura 3.1 podemos apreciar algunos resultados de la implementación de la función de ruido de Perlin con diferentes escalas en las coordenadas de entrada.

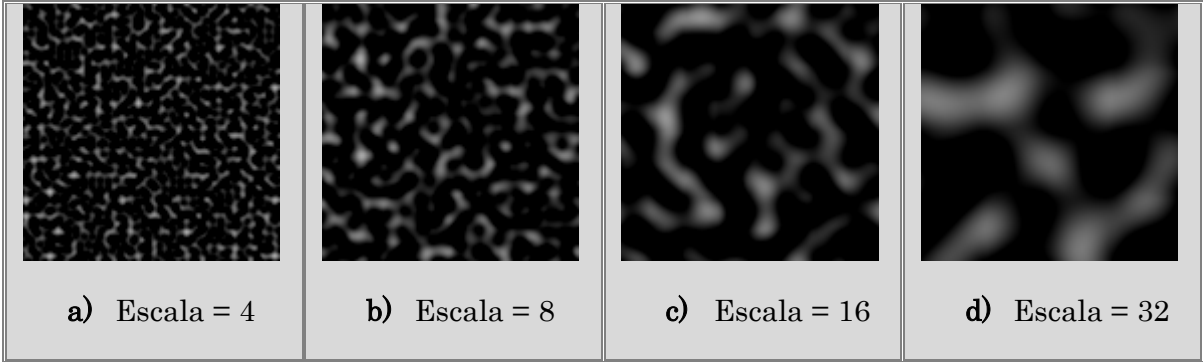


Figura 3.1 Texturas 2D generadas por computadora con la implementación de la función de ruido de Perlin a diferentes escalas.

Como se ha mencionado con anterioridad, la función de ruido nos agrega en el resultado de la generación de texturas, una apariencia más compleja en la misma, la cual en muchas ocasiones es más parecida a los materiales que encontramos en la naturaleza. En la siguiente figura 3.2, se muestra una imagen uniforme (a), teniendo en comparación las imágenes (b) y (c) a las cuales se les ha agregado la función de ruido obteniendo una apariencia más compleja y útil para el simulado de una textura.

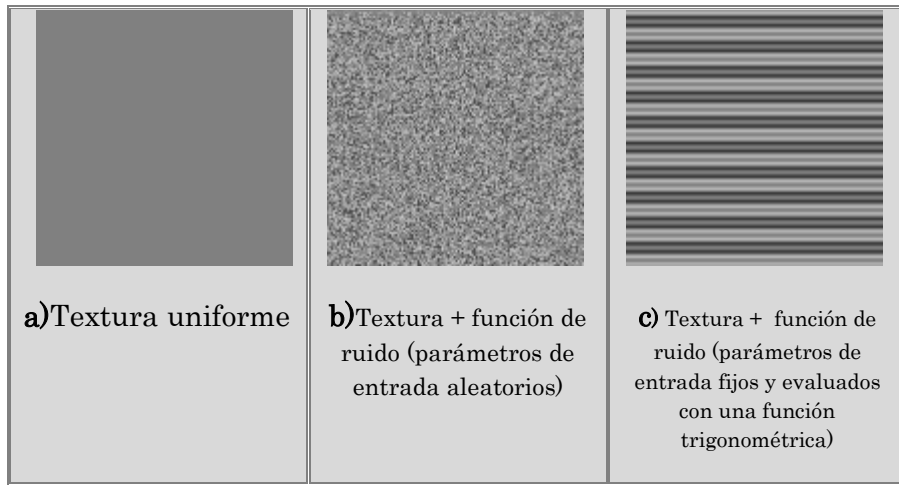


Figura 3.2 Texturas 2D generadas por computadora. a) Textura uniforme. b) Textura uniforme con ruido aleatorio c) Textura uniforme con un valor fijo en la entrada de la función ruido.

3.1.1.2 Función movimiento Browniano fraccional

Esta función se desarrollo en base al pseudocódigo mostrado en el capítulo anterior, teniendo los siguientes parámetros de entrada:

Funcion fBm (xmax, ymax, escala, h, lacunaridad, num_octavas)

- **xmax, ymax** → Determinan el tamaño de la textura a generar en base del valor asignado en las coordenadas (xmax, ymax).
- **Escala** → Parámetro con el valor de la escala a la cual se desarrollara la textura, entre más pequeño sea el valor de este parámetro, aparecerán más pequeños detalles sobre la textura.
- **H** → Exponente de Hurst, proporciona el grado de rugosidad en la textura.
- **Lacunaridad** → Parámetro que agrega espacios o huecos en la textura generada. Entre mayor sea el valor de lacunaridad, la textura presentara más espacios vacíos.
- **Octavas** → Indica el número de iteraciones a realizar en la rutina, en donde al agregar un valor grande, aumentará el nivel de detalle en la textura. Solo hay que tener cuidado de no generar aliasing.

Después de probar la rutina y generar una base de texturas, cambiando uno a uno cada parámetro en el algoritmo, se pueden visualizar algunos de los resultados en la figura 3.3.

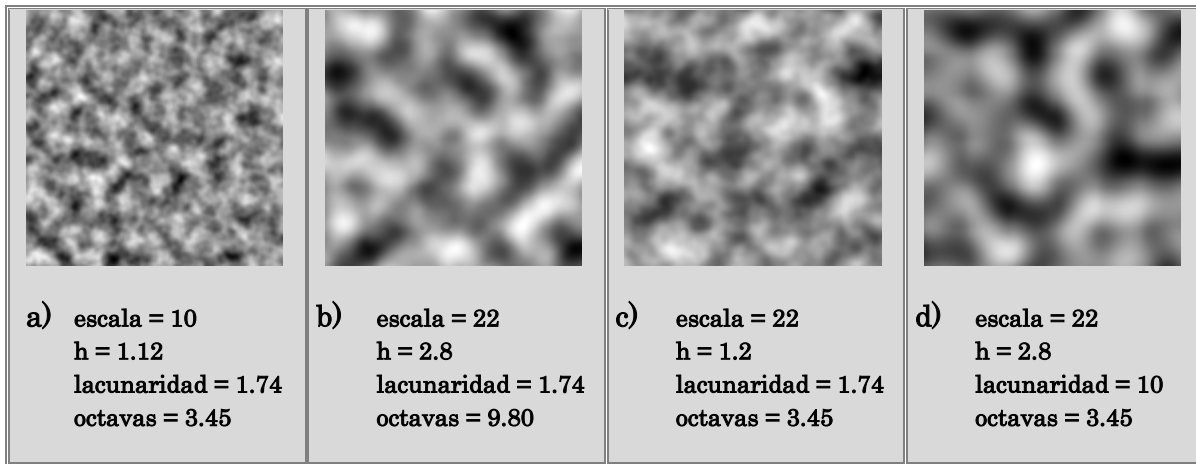


Figura 3.3 Texturas 2D generadas con la función fbm.

Con dichos resultados se podrá trabajar más adelante para obtener las texturas que se plantearon en el objetivo de este trabajo de tesis.

3.1.1.3 Función terreno heterogéneo

La función de terreno heterogéneo, se describe como una función muy útil en la generación de terrenos, utilizado principalmente en ambientes virtuales en 3D. Creando un ambiente de altas montañas, acantilados y zonas de superficies planas, tal como se comporta un terreno.

En este caso se realizó la implementación del algoritmo en 2D, obteniendo una apariencia de textura que nos fue muy útil como base para la generación de la textura del simulador.

Los parámetros de entrada de esta función son:

Funcion Terreno_heterogeneo (xmax, ymax, h, lacunaridad, offset, num_octavas)

- **xmax, ymax** → Determinan el tamaño de la textura a generar.
- **H** → Exponente de Hurst, proporciona el grado de rugosidad en la textura.
- **Lacunaridad** → Parámetro que agrega espacios o huecos en la textura generada. Entre mayor sea el valor de lacunaridad, la textura presentara más espacios vacíos.
- **Offset** → Parámetro que indica la elevación del terreno.
- **Octavas** → Indica el número de iteraciones que debe realizar la rutina.

Después de generar una base de imágenes con el algoritmo de esta función se puede visualizar algunos de los resultados en la figura 3.4, los cuales se obtuvieron al cambiar uno a uno los parámetros antes mencionados.

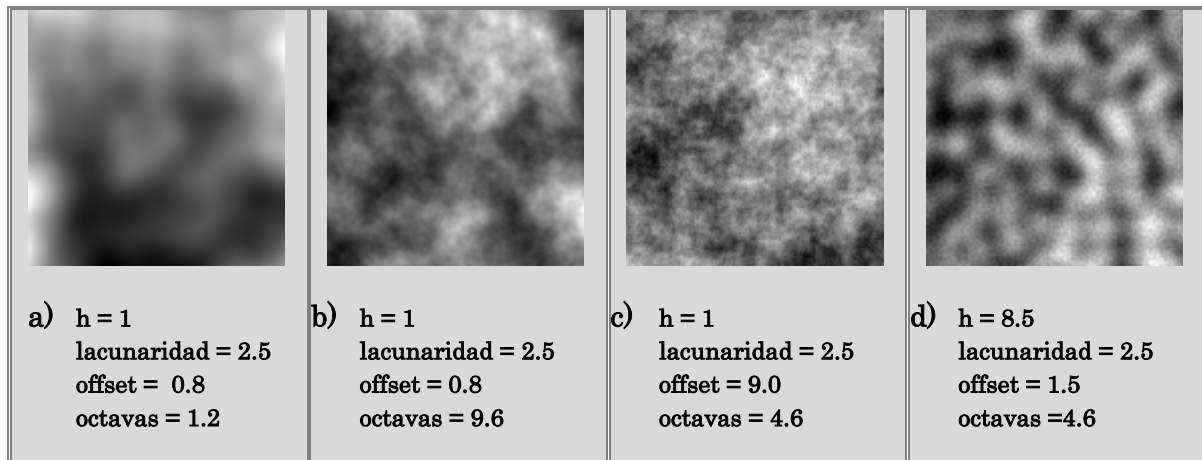


Figura 3.4 Texturas 2D generadas con la función terreno heterogéneo.

3.1.1.4 Función turbulencia

Esta función se implementó en base al pseudocódigo mostrado en el capítulo anterior. Dicha función proporcionó una textura con vascularidades, la cual resulta muy útil para la generación del tejido con apariencia vascular.

Los parámetros de entrada para esta función, son:

Función turbulencia (xmax, ymax, escala, octava_inicial, num_octavas)

- **xmax, ymax** → Determinan el tamaño de la textura a generar.
- **Escala** → Escala a la cual se desarrollará la textura, entre más pequeño sea el valor de este parámetro se obtendrá más detalles en la textura final generada.
- **Octava_inicial** → Parámetro de inicio del valor de las octavas. Debe ser un valor entero menor que el valor del parámetro de número de octavas.
- **Octavas** → Indica el número de iteraciones que debe realizar la rutina.

Algunas de las imágenes generadas con esta función pueden apreciarse en la figura 3.5, las cuales se generaron modificando uno a uno los valores de sus parámetros de entrada.

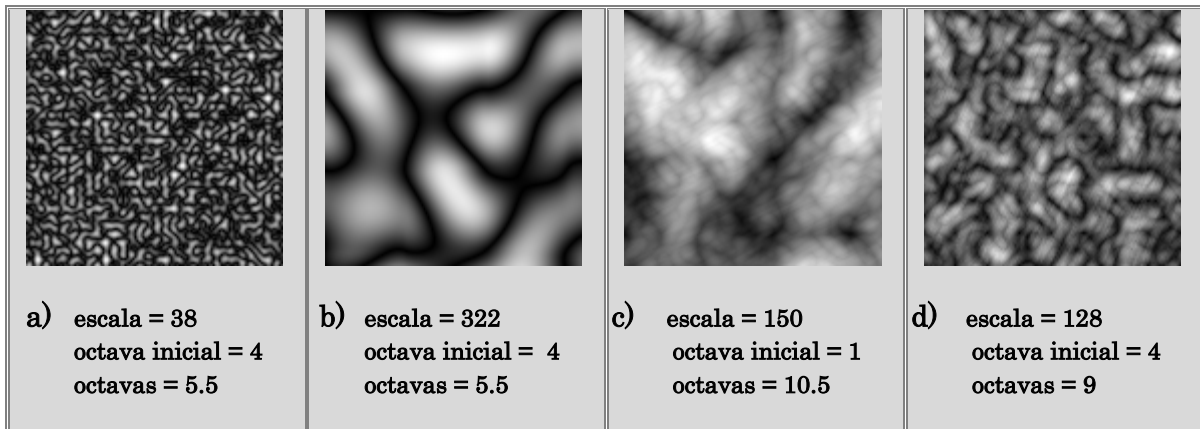


Figura 3.5 Texturas 2D generadas con la función turbulencia.

3.1.1.5 Función turbulencia modificada (apariencia de pliegues)

La implementación de esta función, se llevó a cabo con el mismo algoritmo de la función de turbulencia comentada en el punto anterior. La diferencia radica en una multiplicación de un valor constante por cada valor de la coordenada “y”, operación que se realiza dentro del algoritmo de la función, como se aprecia en el siguiente código:

```
Function Turbulencia_modificada = tex_turb(xmax, ymax, scale, initoctave,
octaves)

    fx = xmax/scale;    %64
    fy = ymax/scale;    %64
    [p, g1, g2, g3] = init_noise;

    for x=1:xmax
        for y=1:ymax
            point(1) = fx*(x/xmax);
            point(2) = fy*(y/ymax)* 0.1; %% Multiplicación de valor
                                     %% constante
            Turbulencia_mod = turb(point, octaves, initoctave, p, g2, R);
            T(x,y) = turbulencia_mod;
        end
    end
end
```

Los parámetros de esta función son:

Funcion turbulencia_modificada (xmax, ymax, escala, octava_inicial, num_octavas)

- **xmax, ymax** → Determinan el tamaño de la textura a generar.

- **Escala** → Escala a la cual se desarrollara la textura, entre más pequeño sea el valor de este parámetro se obtendrá más detalles en la textura final generada.
- **Octava_inicial** → Parámetro de inicio del valor de las octavas. Debe ser un valor entero menor que el valor del parámetro de número de octavas.
- **Octavas** → Indica el número de iteraciones que debe realizar la rutina.

Los parámetros de esta función son los mismos que los valores requeridos en la función de turbulencia. Los cambios de la función “turbulencia modificada”, están propiamente dentro del algoritmo. En la figura 3.6 se aprecian algunos de las texturas obtenidas con esta función.

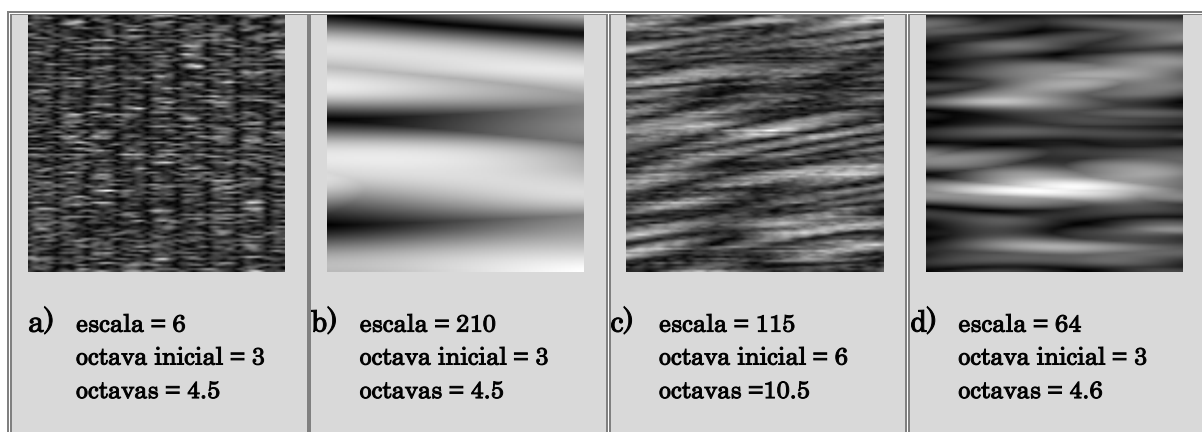


Figura 3.6 Texturas 2D generadas con la función turbulencia modificada.

3.2 Uso de los canales del modelo HSV

La importancia del modelo de color HSV para la generación de texturas, radica en que bajo este modelo, la percepción del color, las tonalidades y la cantidad de brillo se hace de una forma más intuitiva, es decir, es más fácil identificar el matiz, el grado de pureza del color y la cantidad de brillo o contraste que se desea sobre la textura.

La descomposición de una imagen en este modelo de color, se logra obtener una imagen en tres capas en escala de grises, una para el matiz, una para la saturación y una para el valor o brillo.

Con la ayuda de la herramienta ImageJ¹³, se trabajo para la descomposición de las imágenes médicas al modelo de color HSV, lo cual fue muy útil para tener un inicio de partida en la generación de las texturas.

Los pasos a seguir sobre el programa ImageJ son:

¹³ **ImageJ:** Programa desarrollado en Java de dominio público de procesamiento de imágenes. Puede mostrar, editar, analizar, procesar, guardar e imprimir 8-bits, 16-bit y 32-bit de imágenes. Puede leer diferentes formatos de imagen como: TIFF, GIF, JPEG, BMP, DICOM, FITS. Es compatible con las pilas (stack). Distribuciones descargables están disponibles para Windows, Mac OS, Mac OS X y Linux.

- Cargar la imagen de textura real.
- Utilizar la opción **HSV Stack** sobre la imagen cargada.
- Guardar los resultados obtenidos de las tres capas en escala de grises.

En la figura 3.7 se muestran el programa ImageJ con una imagen real en RGB descompuesta en sus tres canales del modelo HSV. Estos mismos pasos se realizaron a cada una de las tres imágenes médicas reales. Para así obtener las tres capas en escala de grises de cada imagen.

Una vez obtenidos los resultados de las capas de cada imagen en escala de grises, se analizaron para lograr obtener una imagen sintética lo más parecido a estas. Para finalmente invertir los pasos, es decir contar con tres imágenes en escala de grises para cada canal o capa del modelo HSV y convertir la imagen en una sola a color con el modelo RGB.

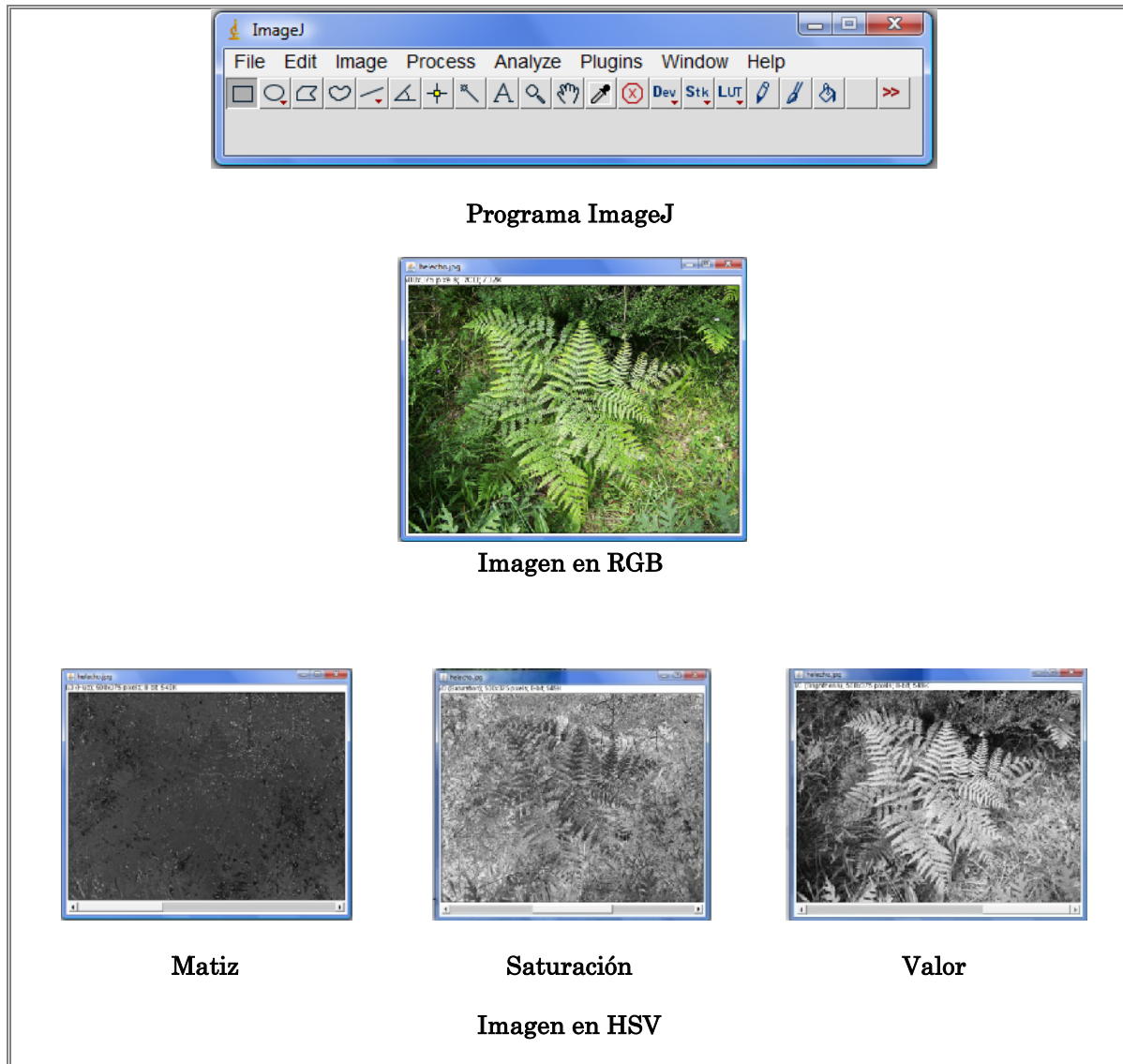


Figura 3.7 Con ayuda de la herramienta ImageJ se logra la conversión de una imagen a color RGB al modelo de color HSV. Con el modelo HSV se logra obtener tres capas en escalas de grises de la imagen a color correspondientes a cada canal de matiz, saturación y brillo (valor).

3.3 Generación de texturas procedurales de tejidos de la próstata

La generación de las texturas procedurales se logra con la implementación y combinación de las funciones básicas: *terreno heterogéneo*, *turbulencia*, *turbulencia modificada* y *movimiento Browniano fraccional (fBm)*.

En este apartado se encuentra una breve descripción del desarrollo y generación de cada una de las texturas a sintetizar, haciendo uso de diagramas de flujo e imágenes gráficas que ayudaran a interpretar mejor el proceso de síntesis de cada textura.

La generación de cada textura se baso principalmente en los siguientes pasos: en primera instancia con el análisis de los resultados obtenidos en la descomposición de las imágenes médicas reales del modelo de color HSV, posteriormente con el uso de las funciones procedurales básicas se trato de asemejar las imágenes en la escala de grises de dicho modelo, para finalmente realizar la conversión al modelo RGB de la textura sintética.

A continuación se presentan una a una la generación de cada textura.

3.3.1 Textura del tejido de la uretra (vascularidades)

Se inicia la generación de esta textura con la descomposición de la imagen médica real. Teniendo las tres siguiente imágenes a sintetizar, mostradas en el figura 3.8.

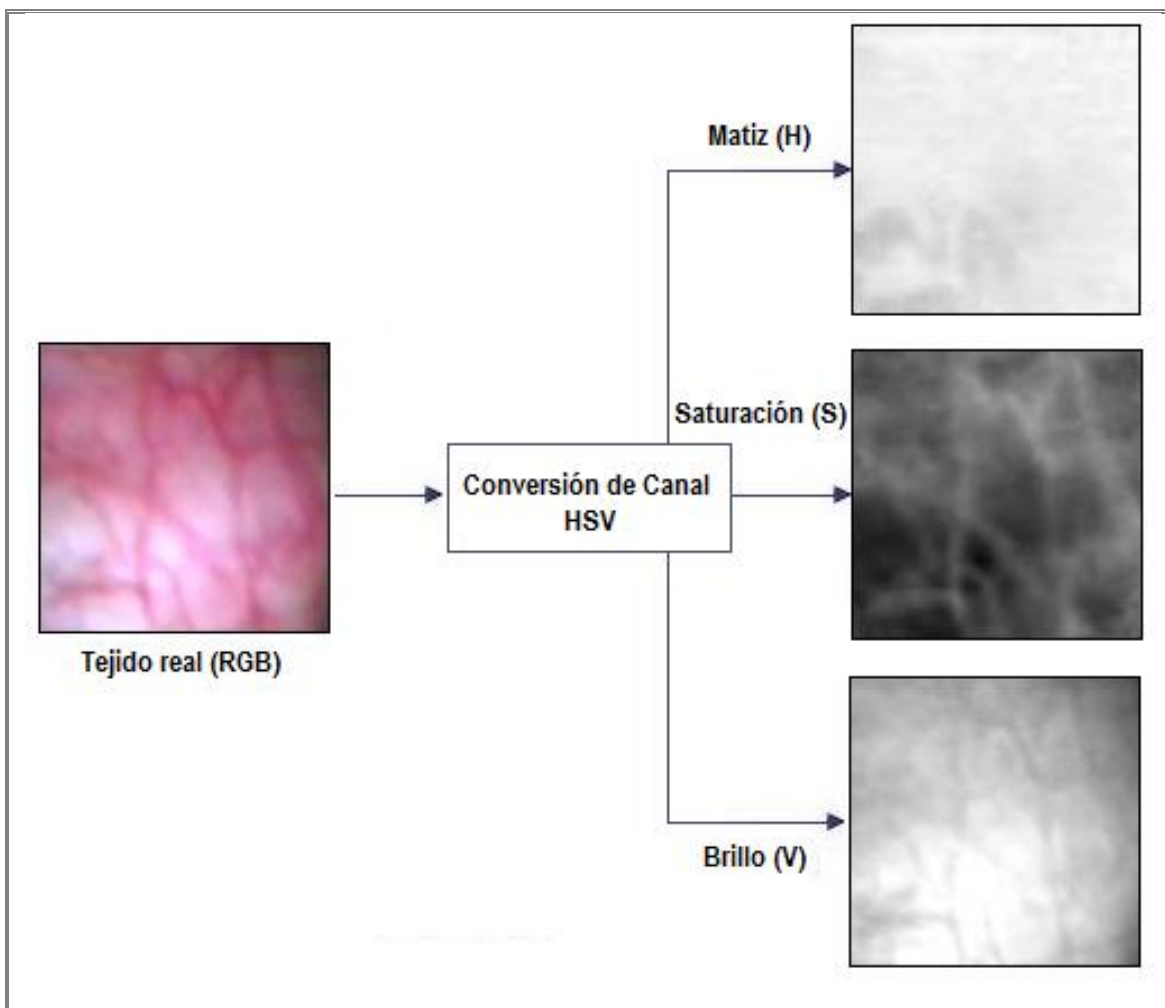


Figura 3.8 Descomposición de la imagen de tejido real de vascularidades al modelo de color HSV.

Una vez analizadas las tres imágenes en escala de grises del modelo HSV de la imagen real, se procede a desarrollar la síntesis de esta textura con el uso de las tres funciones procedurales básicas: *terreno heterogéneo*, *turbulencia* y *fBm*.

Cabe mencionar que la selección de dichas funciones procedurales, se llevo a cabo mediante el análisis de los resultados previos obtenidos con los algoritmos de cada función, siendo estas las que más cumplen con las características del tejido real que se desea sintetizar, así como el uso de los valores de sus parámetros con los cuales obtenemos dichos resultados. Por ejemplo, la función de turbulencia nos proporciona las características de venas o vascularidades dentro de la textura, la cual indudablemente nos será útil para darle dicha apariencia a nuestra textura sintética.

Para profundizar más en el desarrollo y generación de la textura vascular, se presenta el siguiente diagrama de flujo, en el cual se encontrará la explicación de la integración de las tres funciones procedurales básicas, para obtener la textura final deseada.

3.3.1.1 Diagrama de flujo para la generación de textura con vascularidades

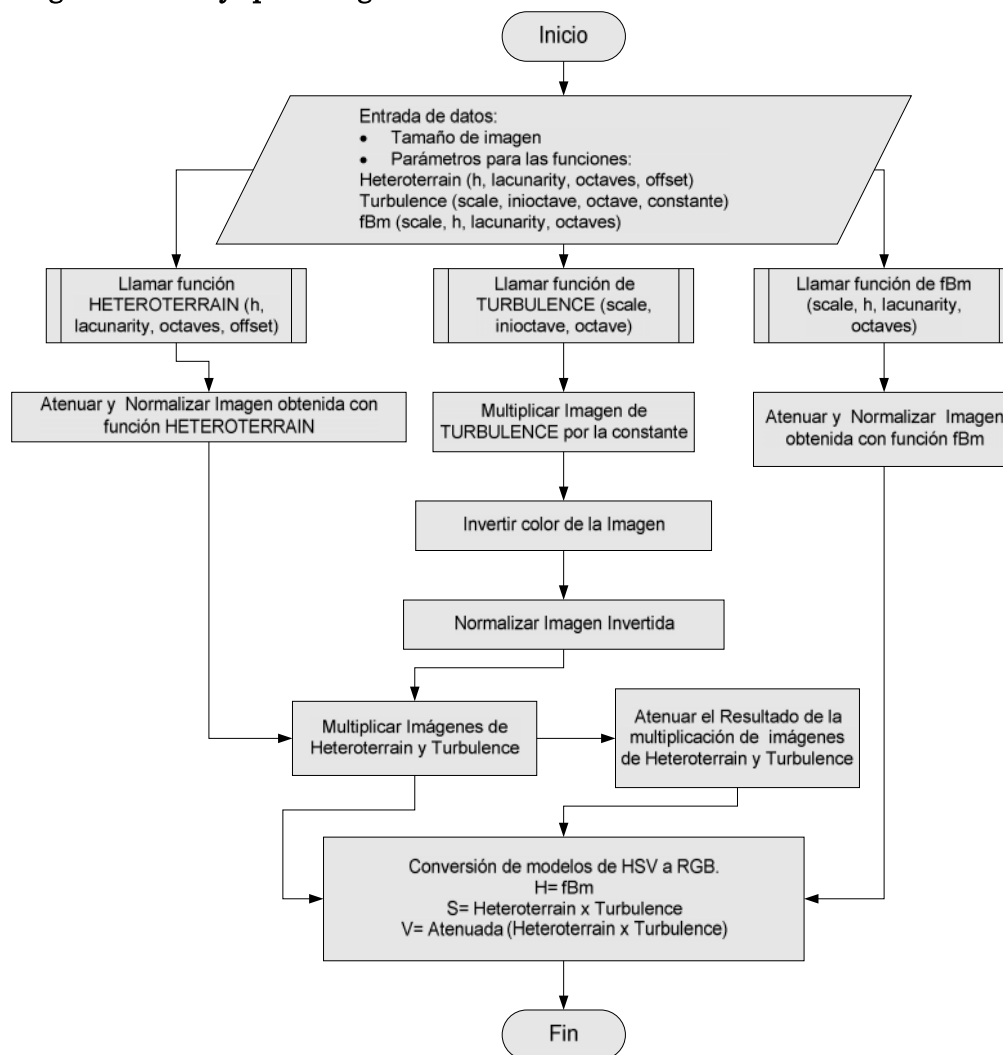


Diagrama Flujo

En este algoritmo se recibirán los siguientes parámetros de entrada:

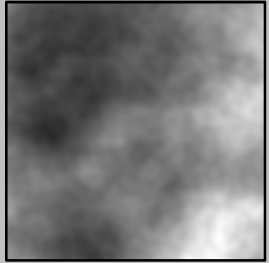
- **xmax, ymax** → Estos parámetros nos determinaran el tamaño de la textura a generar.
- **Parámetros de la función terreno heterogéneo** → (h, lacunaridad, offset, octavas).
- **Parámetros de la función turbulencia** → (escala, octava_inicial, octavas, valor constante).
- **Parámetros de la función fBm.** → (escala, h, lacunaridad, octavas).

Los parámetros de cada función nos darán la posibilidad de manipular los valores, lo que nos dará la posibilidad de visualizar diferentes resultados, teniendo la viabilidad de elegir aquel que sea el más convincente para usar en el simulador.

Continuando con el proceso mostrado en el diagrama de flujo, se procede a la ejecución de las tres funciones básicas. Es importante mencionar que las operaciones posteriores se realizaron con la finalidad de obtener las características lo mas similares de las texturas en escala de grises del modelo de color HSV de la imagen real. Las cuales se explican a continuación.

Se obtiene la textura 2D de la función *terreno heterogéneo*, los valores de los parámetros se manejaron de acuerdo a los resultados previos analizados de dicha función. Posteriormente se aplica a la textura obtenida una atenuación y una normalización. Con la atenuación logramos agregar más brillo a la textura generada, y con la normalización logramos establecer los valores dentro del rango de la escala de los grises, es decir valores intermedios entre 0 y 1. Esta función agregara a la textura final una zona en donde se concentraran las venas con tonalidades más rojas, lo cual dará la apariencia de congestión e inflamación de algunas venas. Sus parámetros y la visualización de la función la podemos observar en la siguiente tabla.

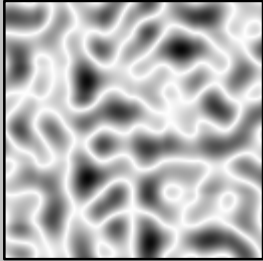
Función terreno heterogéneo

Parámetros *	Textura
H=1.2 Lacunaridad=1.9 Offset=1.5 Número octavas=4.63	

* *Nota: De aquí en adelante, los valores de todos los parámetros fueron elegidos revisando las bases de imágenes generadas automáticamente con las funciones procedurales básicas. Siendo estos los que más se acercaban a los requerimientos visuales de cada textura.*

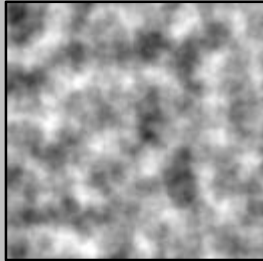
Continuando con el desarrollo de la síntesis, se procede a generar la textura 2D de la función *turbulencia*. Una vez creada la textura producto de la función, se le debe multiplicar un valor constante, logrando con ello un cambio en el color final de la textura a generar. Después será necesario invertir la textura, esto es con la finalidad de lograr que las venas sean las que adquieran las tonalidades de “rojo”. Para finalmente normalizar la función final logrando con ello que los valores radiquen en la escala de los grises, pudiendo visualizar el resultado final con sus respectivos parámetros en la siguiente tabla.

Función turbulencia

Parámetros *	Textura
<p>Escala=128 Octava inicial=4 Octava final=4.87 Factor=3.24</p>	

La última función a generar para el desarrollo de la textura con vascularidades, es la función conocida como *movimiento Browniano fraccional (fBm)*. La cual debe ser atenuada. La atenuación es un aspecto muy importante, ya que en este caso se logrará obtener una textura ideal para ser colocada en el canal H (Matiz) del modelo HSV. Para poder contar con las diferentes tonalidades (blanco, rosa, rojo) que hay dentro de la textura final deseada. En la siguiente tabla se muestra el resultado de la función con sus respectivos parámetros.

Función fBm

Parámetros	Textura
<p>Escala=16 H=1.12 Lacunaridad=2 Número octavas=3.52</p>	 <p>Nota: Se modifico el contraste de la imagen con fines de impresión de la tesis, ya que la visualización de esta textura es un color blanco con pequeñas manchas en gris claro, con lo cual dificulta su apreciación visual.</p>

Para conocer más los efectos que se logran con estas funciones sobre la textura final generada, se recomienda revisar los resultados obtenidos en el siguiente capítulo de este mismo trabajo de tesis, ya que se podrá encontrar una explicación más detallada de la visualización final de la textura.

Ahora bien continuando con la explicación del algoritmo y una vez obtenidos los resultados de las funciones básicas, se procede a realizar un producto entre el resultado de la función terreno heterogéneo y la función turbulencia, las cuales podemos visualizar en la figura 3.9. Logrando con ello tener una imagen con vascularidades combinada con una zona en donde se concentraran las venas más inflamadas; que al ser convertida al modelo RGB, tendrá el efecto de contar con una zona de venas más intensa en color rojo. El resultado de este producto se colocará en el canal S (Saturación) del modelo HSV.

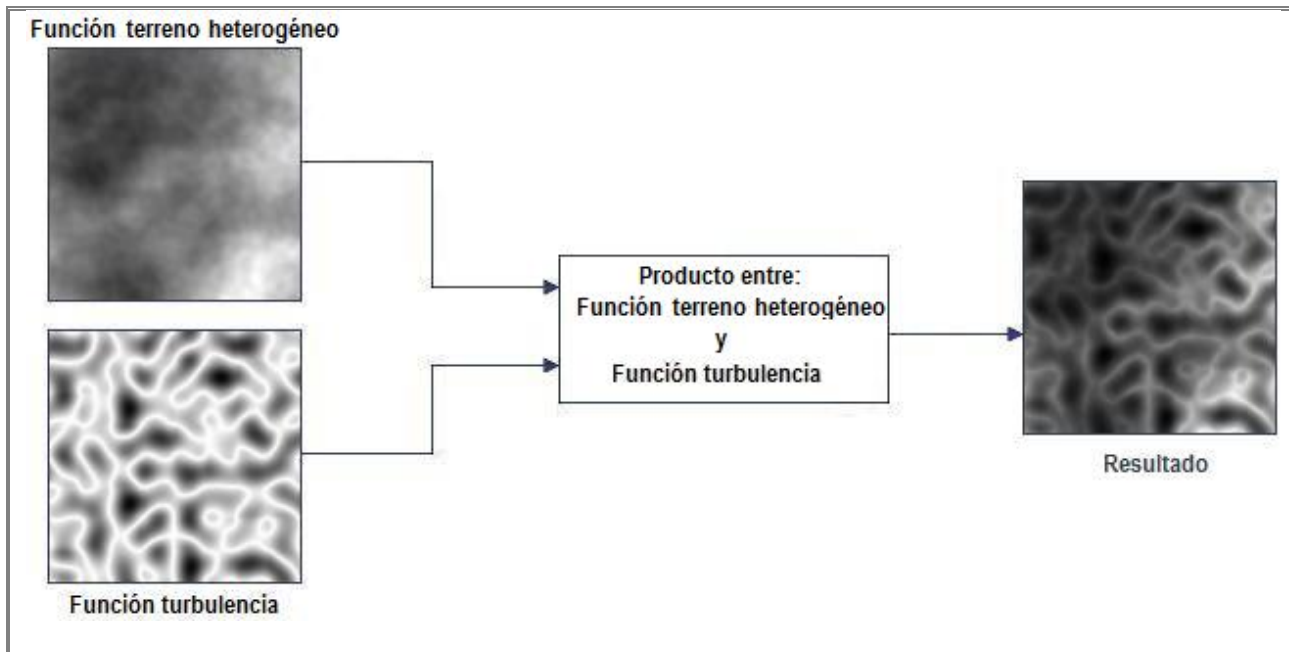


Figura 3.9 Resultado del producto de la función terreno heterogéneo y la función turbulencia, para la generación de la textura de vascularidades.

La última textura para ser colocada en el canal V (Valor) del modelo de color HSV, la obtendremos de la función que resultó del producto de las funciones terreno heterogéneo y turbulencia, vista anteriormente, aplicándole a está una atenuación para así contar con una imagen en color blanco con detalles de vascularidades en tonos muy suaves en color gris claro, como lo podemos apreciar en la figura 3.10.

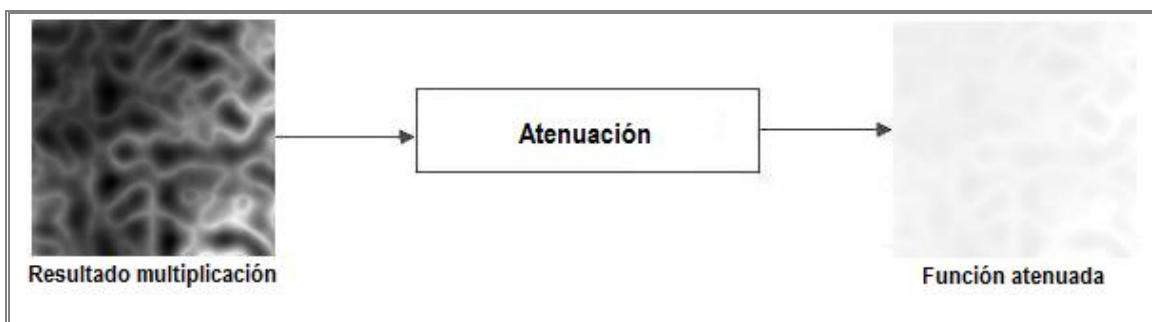


Figura 3.10 Resultado de la atenuación de la función que resultó del producto de funciones básicas: turbulencia y terreno heterogéneo.

Finalmente, contando con las tres texturas necesarias para realizar la conversión de modelos de color HSV a RGB, se procede a realizar lo siguiente. En el canal de matiz (H), se asignará el resultado de la función fBm. En el canal de saturación (S), el resultado del producto de la función terreno heterogéneo y turbulencia. En el canal de valor (V), el resultado de la atenuación del producto de funciones terreno heterogéneo y turbulencia. En la figura 3.11 se aprecia el resultado gráfico de la conversión final de modelos HSV a RGB, apareciendo también la imagen real del tejido para su comparación con la imagen generada por computadora.

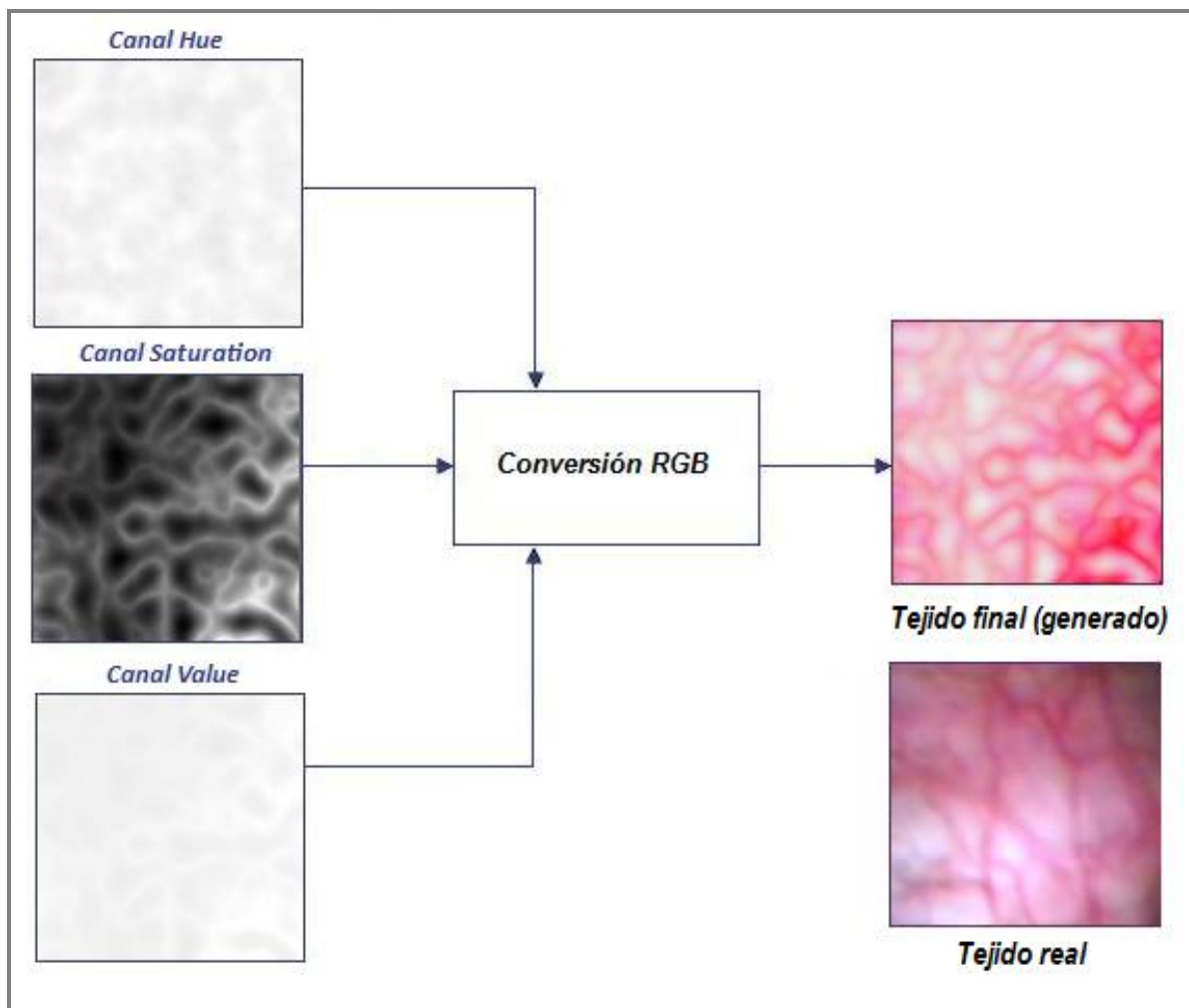


Figura 3.11 Conversión de modelos HSV a RGB, obteniendo el resultado final de la textura sintética de vascularidades.

3.3.2 Textura del tejido interno de la próstata (apariencia amarillo – pardo)

Para la generación de esta textura, se iniciará por analizar sus correspondientes imágenes en escala de grises de sus tres canales sobre el modelo HSV de la imagen real, las cuales podemos visualizar en la figura 3.12.

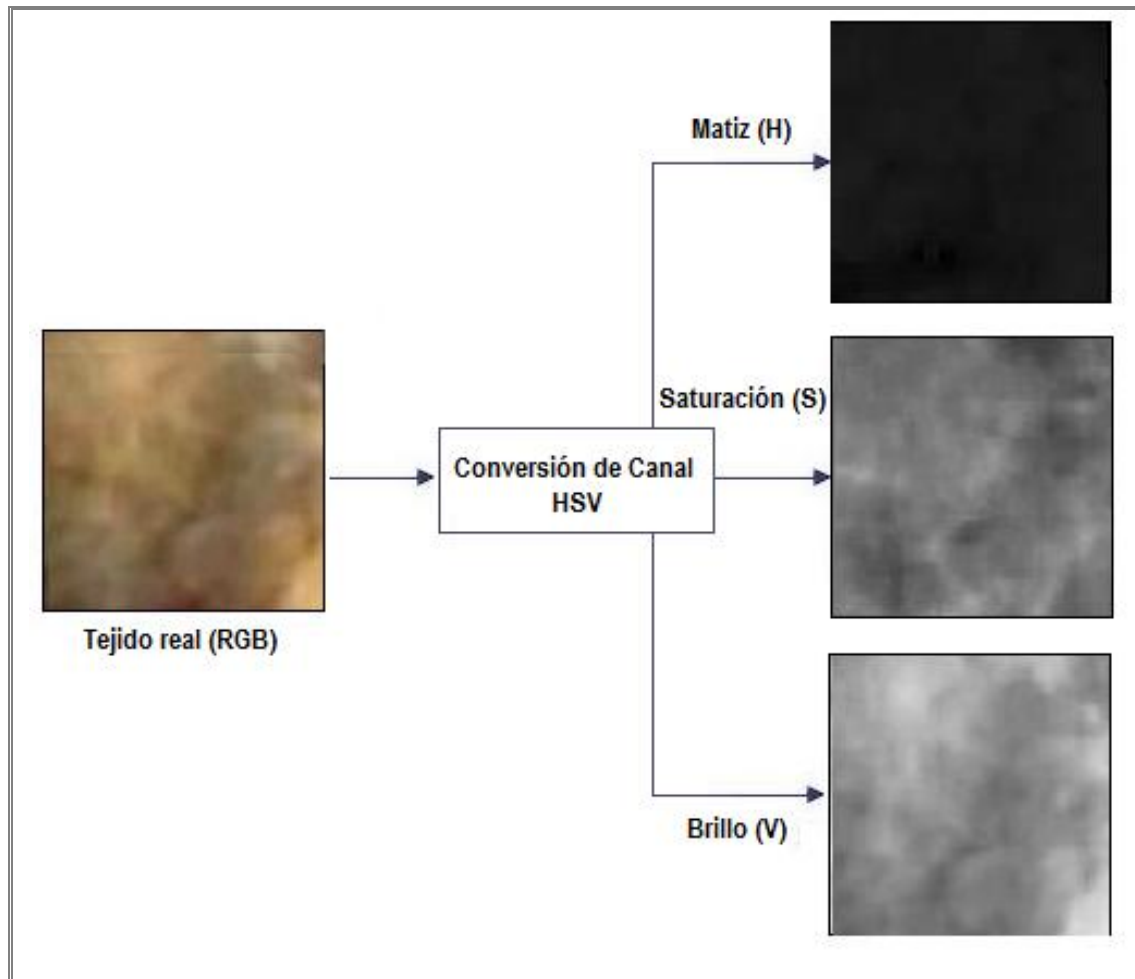


Figura 3.12 Descomposición de la imagen de tejido real interno de la próstata (apariencia amarillo – pardo) al modelo de color HSV.

Al igual que la textura anterior, se trabajó en la obtención de tres texturas con uso de las funciones procedurales básicas; consiguiendo con ellas la apariencia lo más parecido de las tres capas del modelo de color HSV de la imagen real y así obtener la textura final deseada.

La síntesis de esta textura se generó con el uso de las siguientes funciones básicas: *turbulencia* y *fBm*. Es importante mencionar que los valores de los parámetros que aparecen en dichas funciones, fueron resultado del análisis de las bases de imágenes que se trabajaron para cada función procedural, eligiendo aquella textura que cumpliera con los requerimientos visuales lo más parecido a los detalles que presentan las imágenes de textura real.

A continuación se presenta el siguiente diagrama de flujo, con una serie de imágenes con las cuales se podrá entender mejor el desarrollo y generación de la textura de tejido interno de la próstata.

3.3.2.1 Diagrama de flujo para la generación de textura (aparición amarillo – pardo)

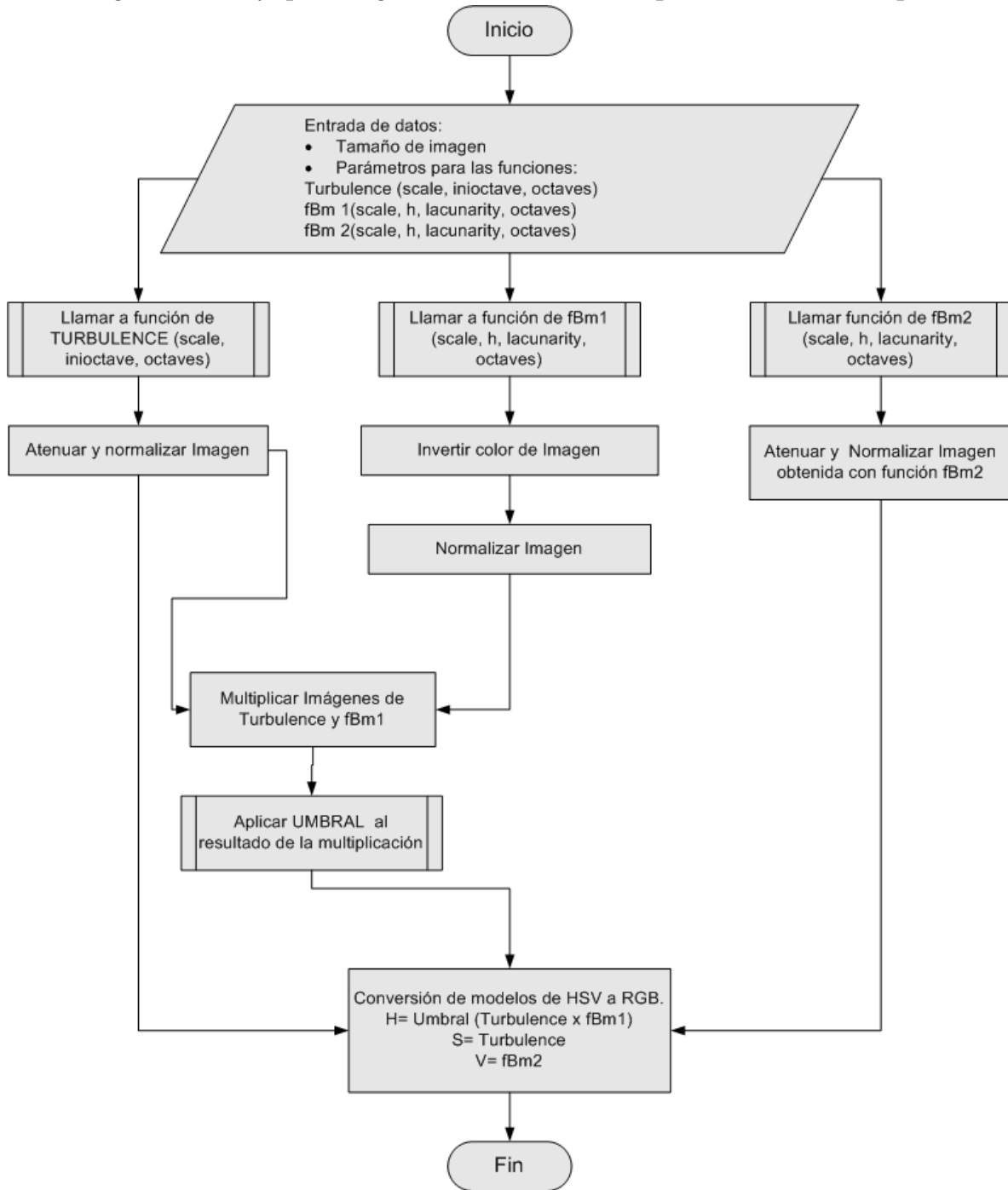


Diagrama Flujo → Tejido Amarillo

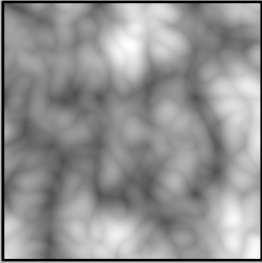
En este algoritmo se recibirán los siguientes parámetros de entrada:

- **xmax, ymax** → Estos parámetros nos determinaran el tamaño de la textura a generar.
- **Parámetros de la función turbulencia** → (escala, octava_inicial, octavas).
- **Parámetros de la función fBm1.** → (escala, h, lacunaridad, octavas).
- **Parámetros de la función fBm2.** → (escala, h, lacunaridad, octavas).

Los parámetros de cada función nos darán la posibilidad de manipular los valores, lo que nos dará la posibilidad de visualizar diferentes resultados. Siendo por eso, que la función de *movimiento Browniano fraccional (fBm)*, sea utilizada doblemente dentro del algoritmo. Obteniendo con ello resultados diferentes entre sí, capaces de agregar los efectos deseados sobre la textura final a generar.

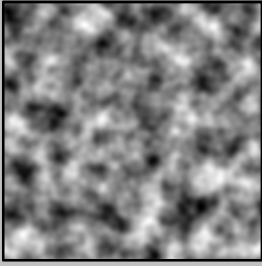
Se procede a obtener la primera textura en 2D utilizando la función procedural básica: *turbulencia*, la cual nos dará la apariencia “base” de la generación de la textura final del tejido. Con ella tenemos una apariencia de esponjosidad muy parecida a las condiciones que se aprecian en la imagen real. Una vez generada, es necesario atenuarla y normalizarla, para obtener una imagen con más brillo gracias a la atenuación, encontrándose en la escala de los grises gracias a la normalización, es decir en el rango de valores de 0 a 1. El resultado de dicha función lo podemos apreciar en la siguiente tabla. Cabe mencionar que dicha textura se colocará en la canal de saturación (S) del modelo HSV.

Función turbulencia

Parámetros	Textura
Escala=96 Octava inicial=2 Número octavas=4.5	


La segunda textura 2D a generar se obtiene con el uso de la función procedural *fBm*. Una vez obtenida la textura, esta debe invertir sus colores (blanco/negro) y debe normalizarse, para ser utilizada más adelante en la generación de la textura interna de la próstata. El uso de esta función es muy importante, ya que con ella se podrá llevar a cabo un producto, para lograr añadir los diferentes tonos de color a la textura final, en este caso un color predominantemente en “amarillo”, con algunos tonos en blanco y rojo, con lo cual se logra una simulación visual lo más parecida de la imagen real. El resultado de la función y los valores de sus parámetros se muestran en la siguiente tabla.

Función fBm1

Parámetros	Textura
Escala=16 H=0.5 Lacunaridad=2 Número octavas=2.52	

La última función procedural de que haremos uso en la generación de esta textura, es nuevamente la función *fBm* evaluada con diferentes valores en sus parámetros de entrada a diferencia de la función anterior. Dicha función será atenuada y normalizada para contar con una textura en color blanco con algunas manchas en gris claro. Esta función nos agregara parte de su estructura en el textura final generada, cabe mencionar que el resultado se colocará en el canal de valor (V) del modelo en HSV. El resultado de la función y los parámetros que sirvieron para su generación se pueden visualizar en la siguiente tabla.

Función fBm2

Parámetros	Textura
Escala=16 H=1.12 Lacunaridad=2 Número octavas=3.52	

Para continuar con la generación de esta textura, y como se comentó con anterioridad, se realizará un producto entre el resultado de la función turbulencia y la función de fBm1. Logrando obtener una imagen con los detalles de ambas funciones, la cual nos dará los efectos visuales deseados sobre todo en la tonalidad del color, siendo muy similares a los que se aprecian en la imagen del tejido real. En la figura 3.13 se puede observar el resultado del producto de dichas funciones.

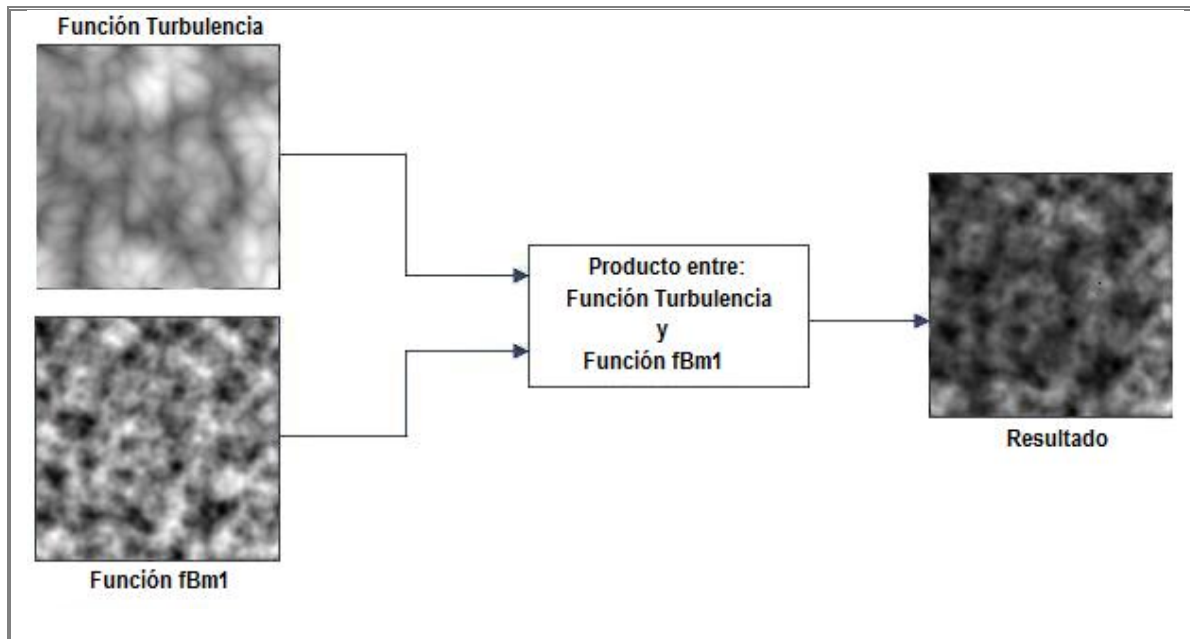


Figura 3.13 Resultado del producto de la función turbulencia y la función fBm1, para textura (tejido colaginoso interno de la próstata).

En la figura 3.14, podemos apreciar el resultado de la textura obtenida en la figura anterior al ser pasada por un bloque de umbral, el cual nos ayudará a obtener el color final de nuestra textura sintética, para el color predominantemente amarillo. Dicha textura será colocada en el canal de matiz (H) del modelo de color HSV.

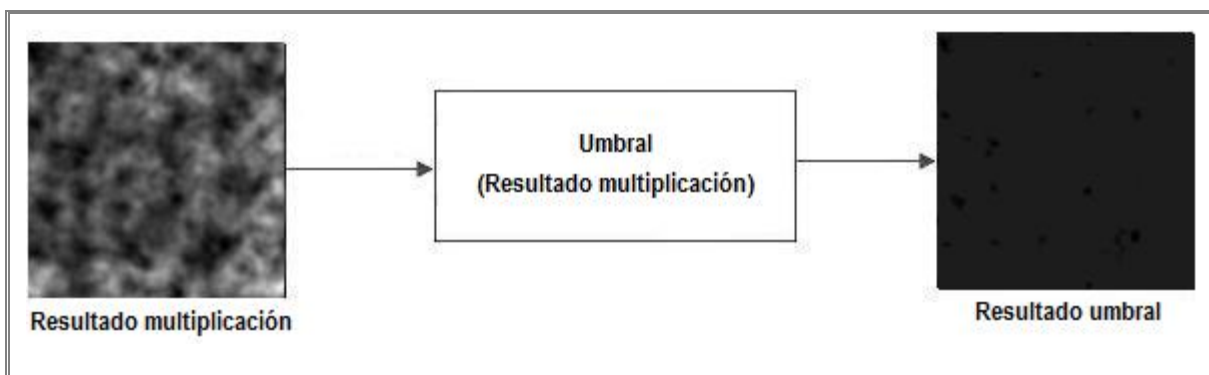


Figura 3.14 Resultado al aplicar el umbral al resultado de la multiplicación de las funciones turbulencia y fBm1 para obtener el color de la textura final del tejido interno de la próstata.

Finalmente se realizará la conversión de modelos HSV a RGB de la siguiente forma: en el canal de matiz (H) se asignó la textura que resulto despues de pasarla por el bloque del umbral. En el canal de saturación (S) se asignó la textura obtenida de la funcion de turbulencia y en el canal de valor (V) se asignó la textura obtenida de la función fBm2, los resultados graficos de la conversion al modelo de color RGB pueden ser apreciados en la figura 3.15.

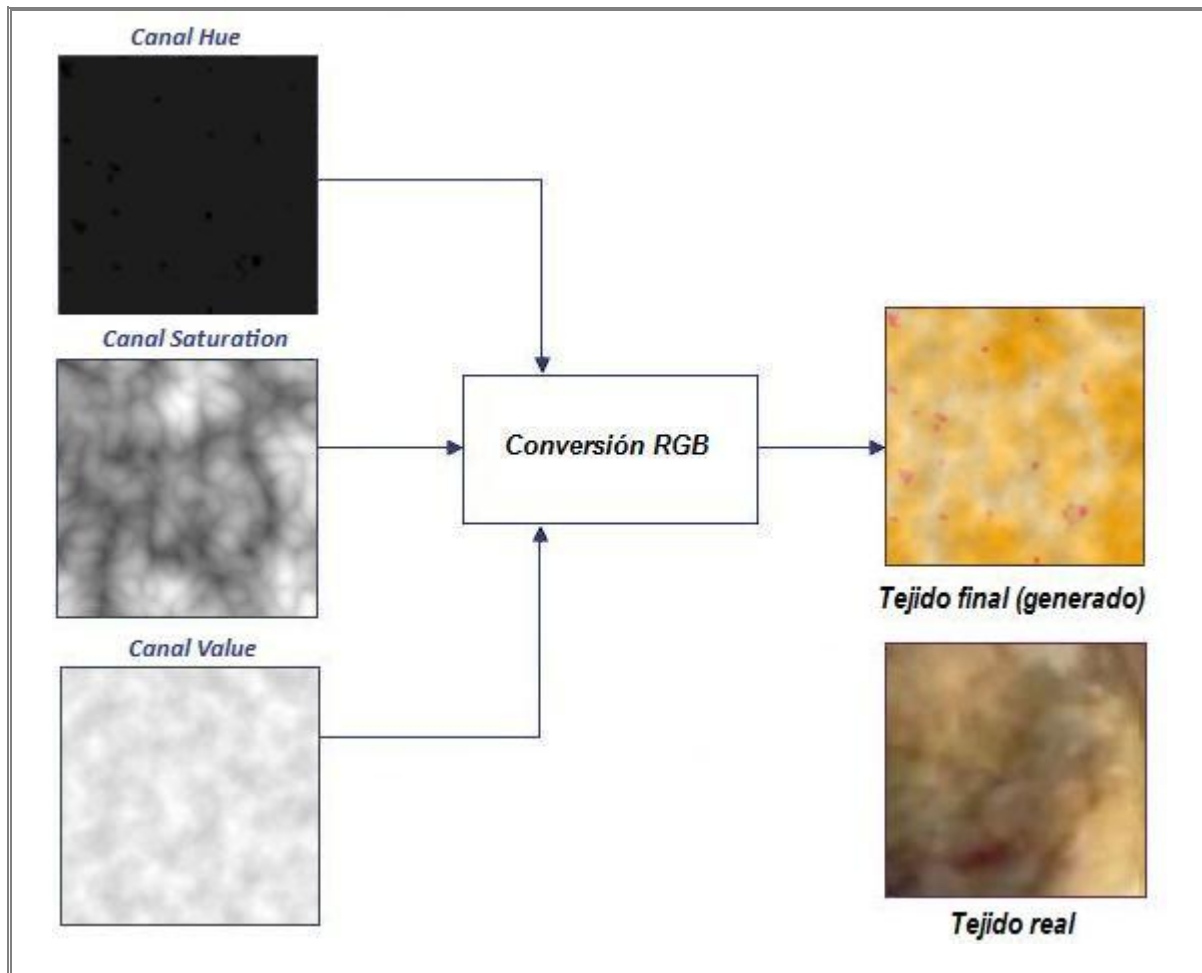


Figura 3.15 Conversión de modelos HSV a RGB, obteniendo una textura sintética (tejido interno de la próstata).

Para conocer más efectos que se logran al modificar los valores de los parámetros de las funciones básicas, de la textura final generada, se recomienda revisar los resultados obtenidos en el siguiente capítulo de este mismo trabajo de tesis, ya que se podrá encontrar una explicación más detallada de la visualización final de la textura de tejido interno de la próstata.

3.3.3 Textura del tejido de la cápsula de la próstata (apariencia rugosa)

Para simular la textura del tejido fibroso de la capsula, se procedió a realizar los mismo pasos que en la generación de las anteriores texturas.

Se inicia con el estudio de las tres capas de textura obtenidas de la imagen real con el modelo de color HSV (ver figura 3.16), con el propósito de obtener tres imágenes sintéticas generadas por computadora muy parecidas.

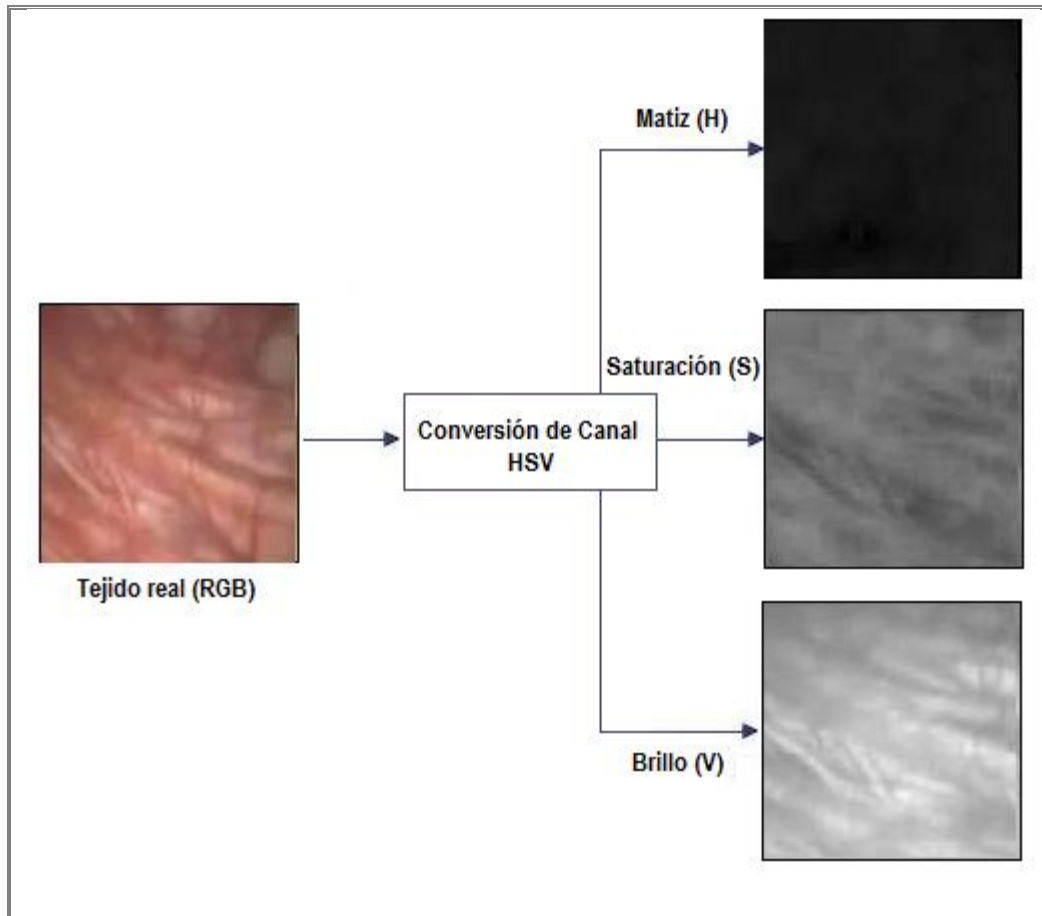


Figura 3.16 Descomposición de la imagen de tejido real de la cápsula de la próstata (apariencia rugosa) al modelo de color HSV.

Esta textura se generó con la función básica: *turbulencia modificada*. Se decidió utilizar dicha función dado que nos proporciona la apariencia de pliegues o ondulaciones, muy parecidas al tejido real que queremos simular.

A continuación se explica en el siguiente diagrama de flujo, el desarrollo y la generación de esta textura, con la cual tendremos la posibilidad de obtener cuatro tonalidades; esto nos da la facilidad de contar con una variedad de texturas aplicables a la cápsula del simulador, que nos servirán para realizar algunas de las pruebas. De la misma forma se cuentan con algunas ilustraciones, las cuales facilitan la explicación del desarrollo de esta textura.

3.3.3.1 Diagrama de flujo para la generación de textura del tejido cápsula

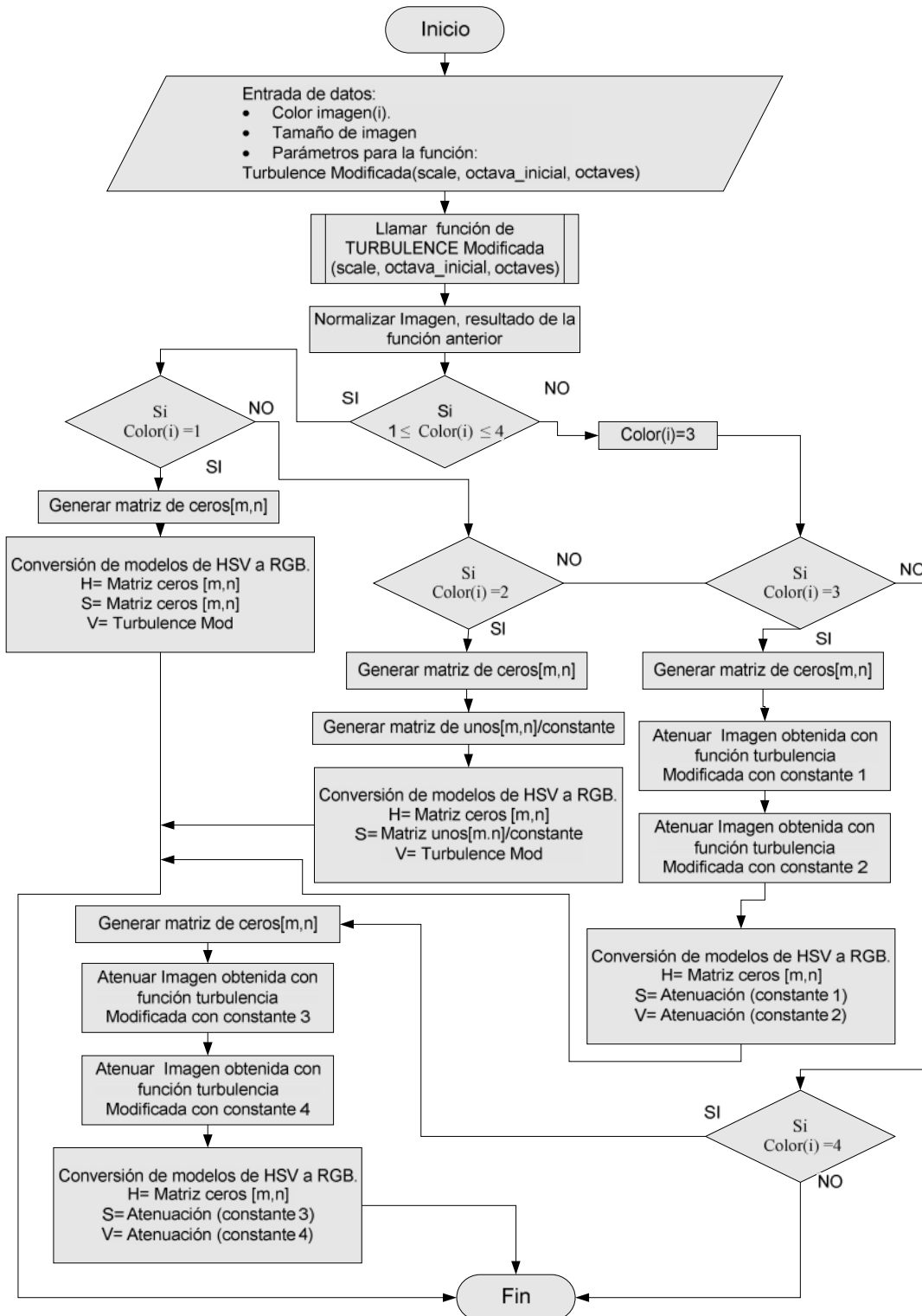


Diagrama Flujo → Tejido Rugoso

En este algoritmo se recibirán los siguientes parámetros de entrada:

- **xmax, ymax** → Parámetros que determinan el tamaño de la textura a generar.
- **Parámetros de la función turbulencia** → (escala, octava_inicial, octavas).
Los parámetros de la función nos darán la posibilidad de manipular los valores, lo que nos dará la posibilidad de visualizar diferentes resultados.

Dentro de la generación de esta textura, tendremos la posibilidad de elegir el valor del parámetro color, pudiendo obtener lo siguiente:

- **Color** → Este algoritmo nos da la posibilidad de generar cuatro texturas de matices diferentes. Teniendo a elegir entre un valor entero que vaya del rango de uno a cuatro.
 - Color = 1 → Textura generada en escala de grises.
 - Color = 2 → Textura generada en color negro con rosa pálido.
 - Color = 3 → Textura generada en color blanco con una tonalidad encendida en rojo.
 - Color = 4 → Textura generada en color blanco con una tonalidad en rosa pálido.

En la figura 3.17, se puede apreciar las diferentes tonalidades, dependiendo del valor que se elija para el color.

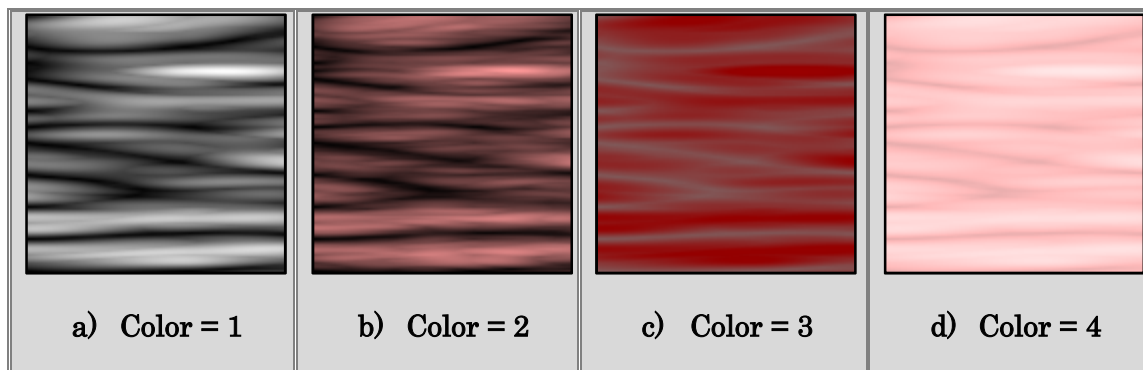
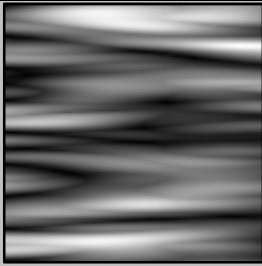


Figura 3.17 Resultados de las diferentes tonalidades en la generación de textura de la cápsula dependiendo del valor que se elija para el color.

Una vez contando con los valores de los parámetros, se generara la textura 2D utilizando la función de *turbulencia modificada*, siendo la única función a utilizar para la generación de esta textura, ya que cuenta con las características visuales de pliegues y rugosidades muy parecidas a la imagen real de la textura de la cápsula. Una vez obtenida la textura se deberá normalizar, la cual será utilizada más adelante para asignarla en alguno de los canales del modelo HSV y obtener el color deseado. En la siguiente tabla se aprecia el resultado de dicha función con los valores utilizados para su generación.

Función turbulencia modificada

Parámetros	Textura
Escala=64 Octava inicial=3 Número octava=4.5	

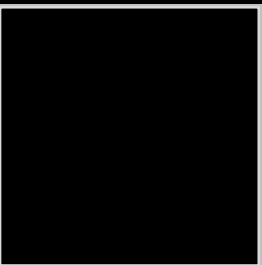
Como se mencionó anteriormente, este algoritmo tienen la capacidad de generar cuatro texturas de tejido de la cápsula con un matiz diferente para cada una. Esta diferencia radica al momento de realizar la conversión de modelos HSV a RGB, utilizando en todas las conversiones la textura generada con la función básica *turbulencia modificada*.

3.3.3.2 Conversión de modelos HSV a RGB para textura del tejido de la cápsula

En esta sección se explicará brevemente la conversión de los modelos HSV a RGB, de las cuatro posibilidades de selección del parámetro color.

Si el valor del parámetro *color* es igual a *uno*, se deberá colocar una imagen uniforme de color negro en los canales de matiz (H) y saturación (S), colocando finalmente el resultado de la función *turbulencia modificada* en el canal (V). Al realizar la conversión de modelos la textura conservara sus características en escala de grises. En la figura 3.18 se puede observar la explicación grafica de la conversión.

Textura uniforme en color negro

Parámetros	Textura
Matriz Ceros	

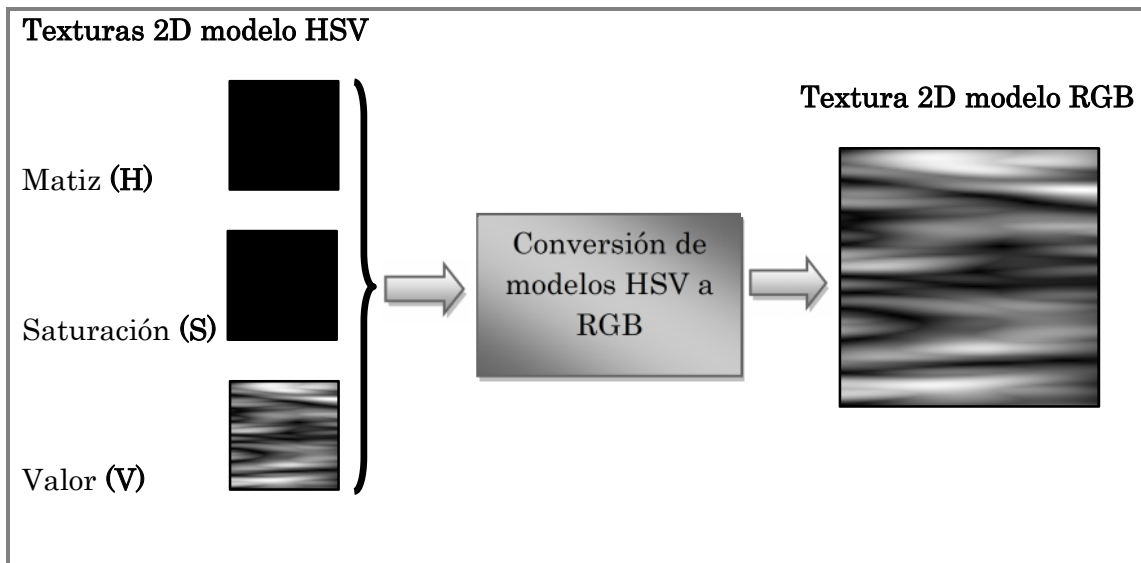
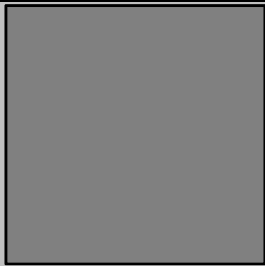


Figura 3.18 Resultado de la conversión de modelos HSV a RGB del tejido de la cápsula en escala de grises.

Cuando el valor del parámetro *color* sea igual a *dos*, se deberá colocar en el canal de matiz (H) una textura uniforme de color negro, en el canal de saturación (S) se colocará una textura uniforme de color gris, para finalmente en el canal de valor (V) se colocara la textura generada de la función procedural *turbulencia modificada*. Al realizar la conversión de modelos obtendremos una textura con apariencia rugosa de un color rosa con negro la cual podemos apreciar en la figura 3.19.

Imagen uniforme en color gris

Parámetros	Textura
(Matriz Unitaria)/ Factor	

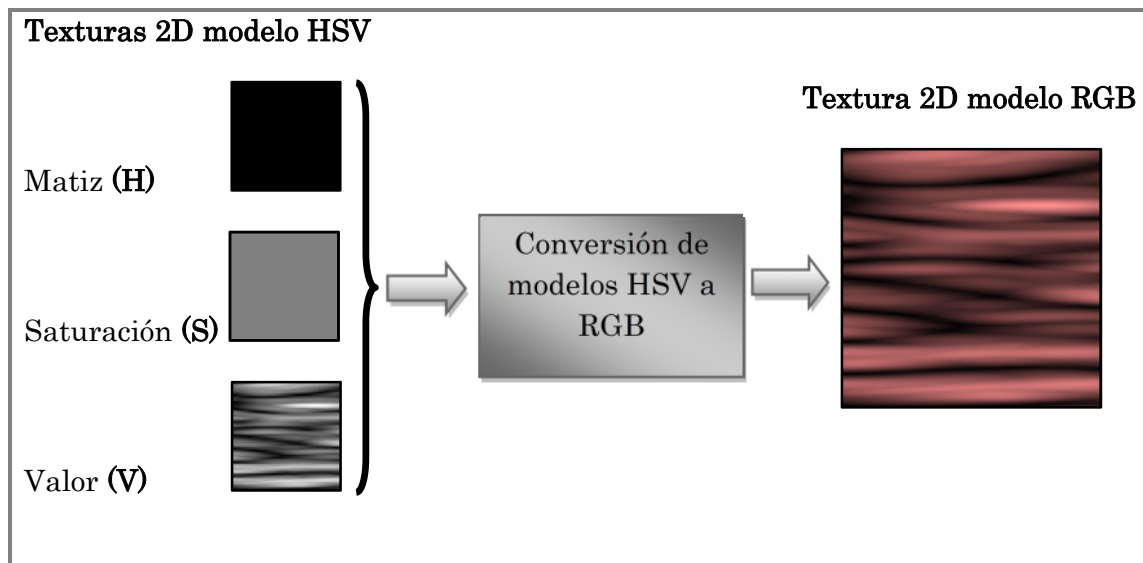


Figura 3.19 Resultado de la conversión de modelos HSV a RGB del tejido de la cápsula en color negro con rosa.

Si el valor del parámetro *color* es igual a *tres*, obtendremos una textura encendida con una apariencia de pliegues. Para generar dicha textura se tendrá que colocar en el canal de matiz (H) una textura uniforme de color negro. Para el canal de saturación (S) se colocará el resultado de la atenuación de la textura generada por la función de *turbulencia modificada*. Para finalmente en el canal de valor (V) se colocará nuevamente un segundo resultado de la atenuación de la misma textura obtenida de la función procedural básica *turbulencia modificada*. Siendo las atenuaciones diferentes entre sí. La atenuación se logra sumando una constante positiva a la textura generada de la función básica, logrando con ello dar una apariencia con más brillo en el resultado de la textura. En la figura 3.20 se muestra la conversión y el resultado de la textura final simulada.

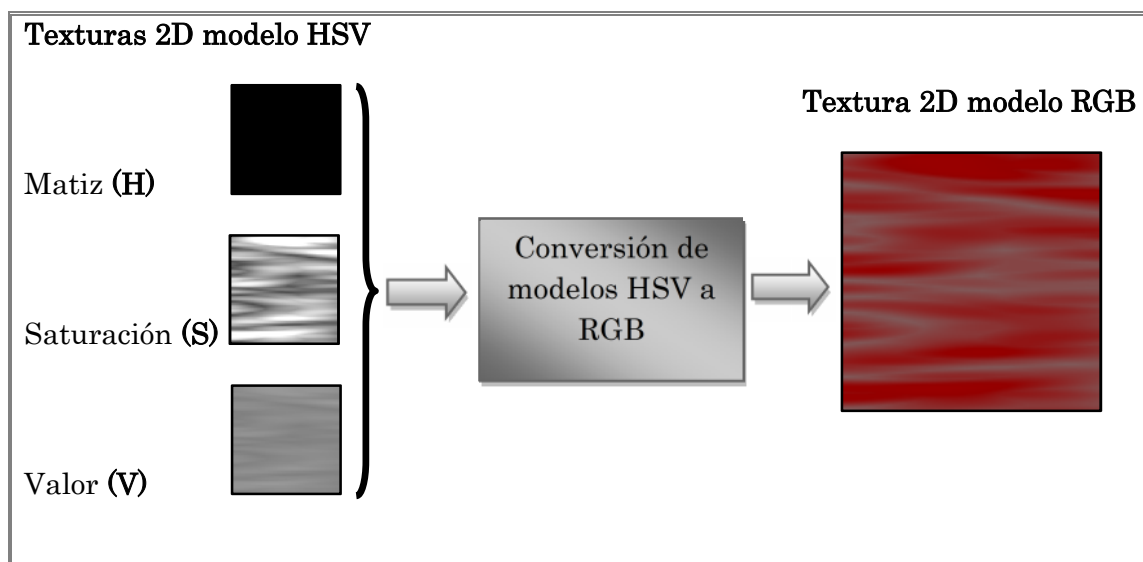


Figura 3.20 Resultado de la conversión de modelos HSV a RGB del tejido de la cápsula en color rojo pálido.

Finalmente si el valor del parámetro *color* es igual a *cuatro*, obtendremos una textura que visualmente se parecerá más a la textura del tejido de la cápsula obtenida de las imágenes medicas reales.

Para generar esta textura se necesito colocar en el modelo HSV en sus tres canales las siguientes texturas: en el canal de matiz (H) una textura uniforme de color negro, en la canal de saturación (S) el resultado de la atenuación de la función procedural básica, y en el canal de valor (V) el resultado de una segunda atenuación de la misma función procedural básica: *turbulencia modificada*. Al igual que el resultado anterior las atenuaciones entre sí son diferentes, al realizar las atenuaciones logramos con ello obtener una apariencia muy similar a los detalles que se aprecia en la imagen médica real. Las atenuaciones se realizan sumando un valor decimal positivo al resultado de la función básica. En la figura 3.21 se aprecia la conversión al modelo de color RGB, así como una muestra de la textura real para su comparación.

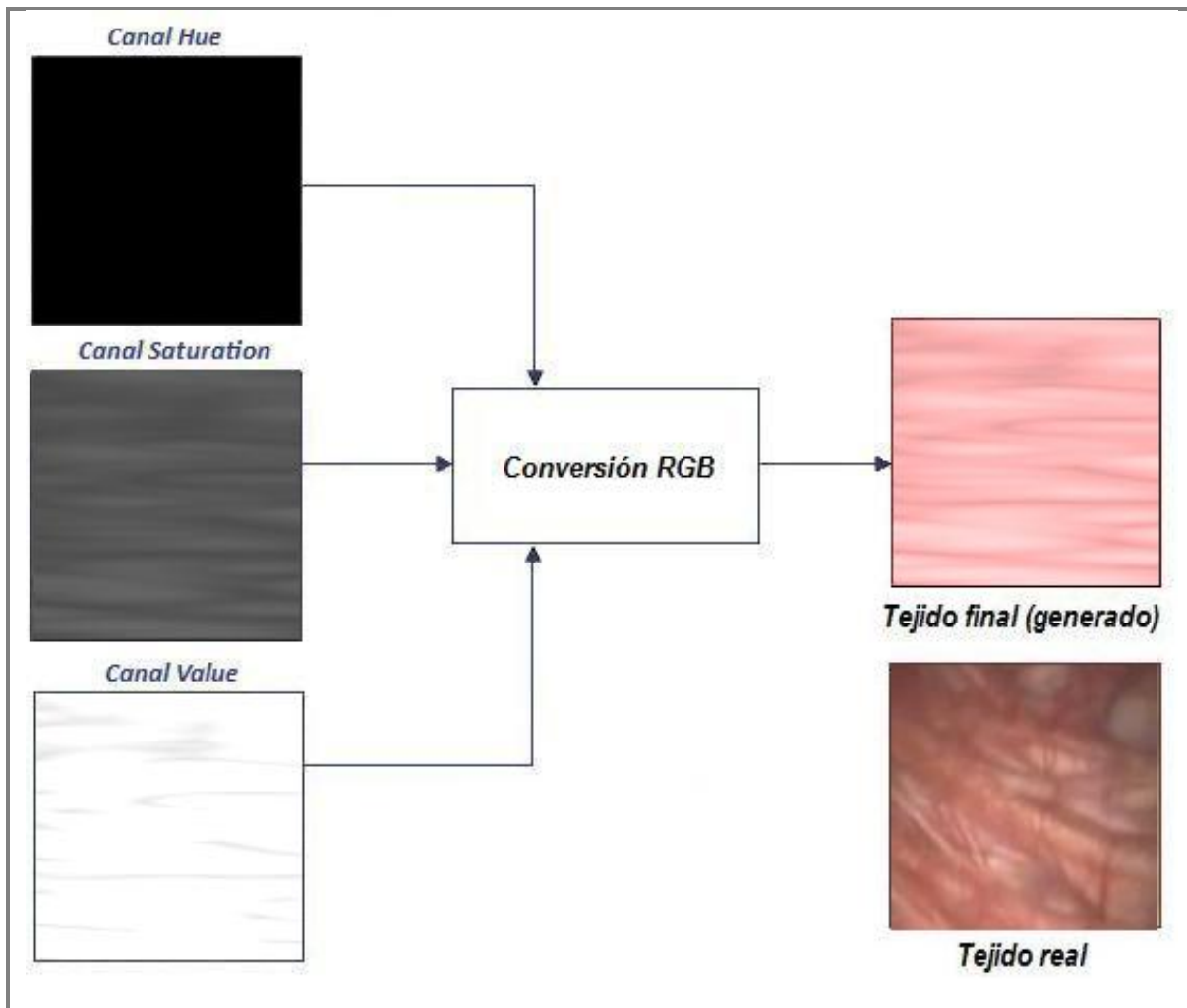


Figura 3.21 Resultado de la conversión de modelos HSV a RGB, obteniendo una imagen de textura final para la cápsula de la próstata.

3.4 Generación de textura volumétrica procedural de la próstata

Una textura sólida, llamada también volumétrica o textura 3D, es aquella que se encuentra definida en los tres planos coordenados (x,y,z). Estas son muy útiles en la representación de materiales con variaciones que requieren vistas desde su estructura interna, logrando con ello una apariencia más natural. Siendo casi imposible el mismo resultado si se contara con una textura en 2D, ya que se obtendrían resultados visualmente muy bajos restando valor a la apariencia del objeto simulado. (Jagnow, 2004)

Los materiales más comunes simulados con texturas sólidas son: la madera, el granito y el mármol. Al utilizar este tipo de texturas nos permiten revisar el interior del objeto simulado, manteniendo una constante en su apariencia visual y con ello asegurar una apariencia más realista. En la figura 3.22 podemos ver la ejemplificación visual de las diferencias de un mapeado con uso de texturas 2D y 3D sobre una superficie en 3D, notando las irregularidades y la discontinuidad del material en el mapeo con la textura 2D, haciendo la comparación con el uso de una textura en 3D la cual nos arroja mejores resultados y la posibilidad de apreciar la estructura y detalles del material desde su interior.

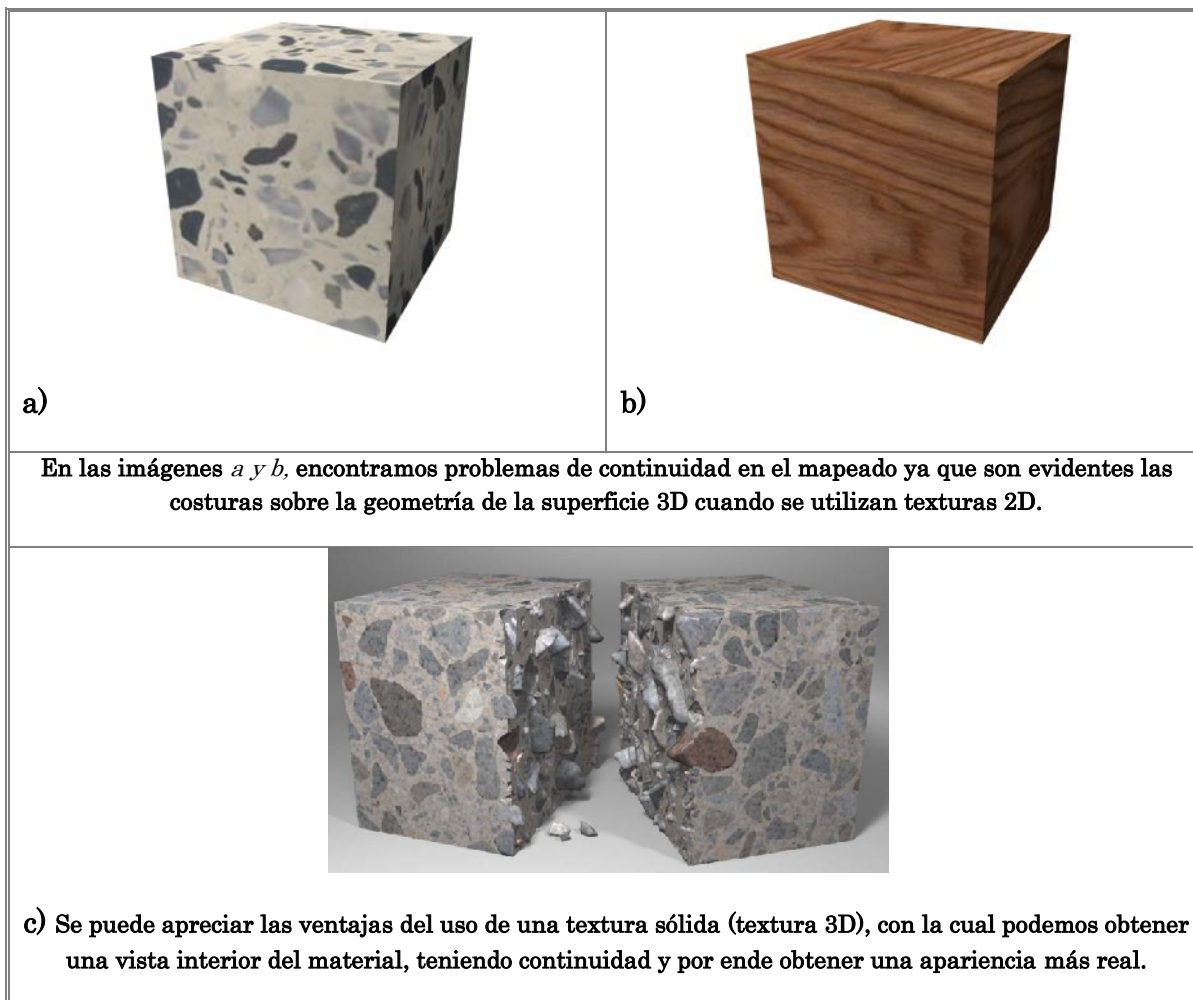


Figura 3.22 Diferencias entre un mapeo 2D y 3D sobre una superficie sólida en 3D. (Jagnow, 2004)

Teniendo como antecedentes las ventajas que ofrecen las texturas 3D, se considera la necesidad de obtener una textura volumétrica para ser utilizada en el simulador virtual de cirugía de próstata, ya que está englobará todas las características visuales que se necesitan para el simulador, es decir, se contará con una continuidad del tejido, con efectos sólidos, permitiendo al usuario llevar a cabo cortes y el efecto de cauterización sobre el tejido al momento de usar el resectoscopio, obteniendo resultados que no perderán la continuidad del material teniendo la posibilidad de realizar vistas del modelo desde su interior sin que con ello se pierda una apariencia visual real.

Para lograr esto se extenderán los resultados ya obtenidos y probados de las funciones procedurales en 2D, explicadas anteriormente en este mismo capítulo. Serán ahora generadas en tres dimensiones logrando con ello la obtención de una textura volumétrica unificada que contendrá las tres texturas: textura del tejido de la uretra (vascularidades), textura del tejido interno de la próstata (amarillo-pardo) y textura del tejido de la capsula de la próstata (apariencia rugosa); con las cuales se formarán los contornos de la glándula prostática.

3.4.1 Procesos para la generación de la textura volumétrica

La generación de la textura 3D de la próstata se puede explicar principalmente con los siguientes pasos a realizar:

- Las funciones básicas 2D de *turbulencia*, *turbulencia modificada*, *movimiento Browniano fraccional (fBm)* y *terreno heterogéneo*, así como los valores de cada parámetro de estas, se seguirán utilizando de la misma forma que se utilizaron en la generación de texturas 2D. La diferencia radicará en la extensión de la coordenada “z” para la generación de la textura sólida.
- Al agregar el parámetro z en los algoritmos, se lograra obtener una profundidad en la textura que conceptualmente nos formara un cubo (x,y,z) del material. En el cual podremos “esculpir” los contornos de la próstata, obteniendo con ello el modelo 3D de la glándula prostática.
- Para la integración de las tres texturas generadas del tejido de la próstata, se utilizara una pila de imágenes de contornos de la glándula en blanco y negro; desarrollando un algoritmo capaz de agregar las texturas correspondientes en la región de la glándula, con lo cual se logrará el objetivo final: generar una textura volumétrica que integre las tres texturas de los tejidos prostáticos con las características y efectos del procedimiento RTU.

3.4.2 Extensión de los algoritmos generadores de textura 2D a un volumen en 3D

Como bien se ha mencionado, el desarrollo y generación de la textura volumétrica, estará dado por la ampliación de los resultados obtenidos en la generación de texturas en 2D. Esta ampliación estará dada con la integración de un parámetro extra llamado “z”, con el cual obtendremos una profundidad en la textura, logrando con ello finalmente una textura sólida en tres dimensiones.

3.4.2.1 Agregando parámetro “z” a los algoritmos generadores de textura

Al agregar el parámetro extra de la coordenada “z”, se logra repetir “z” veces la textura 2D; con lo cual se pudo obtener una pila de imágenes con coordenadas en (x,y,z), tomando en cuenta que el valor de las coordenadas debe ser el mismo, lo que conceptualmente nos formara un cubo en el espacio de tres dimensiones. Gráficamente podemos observar en la figura 3.23 la generación de los cubos de texturas del tejido de la próstata. Cabe mencionar que los valores de los demás parámetros son los mismos que se utilizaron en la generación de las imágenes de texturas 2D.

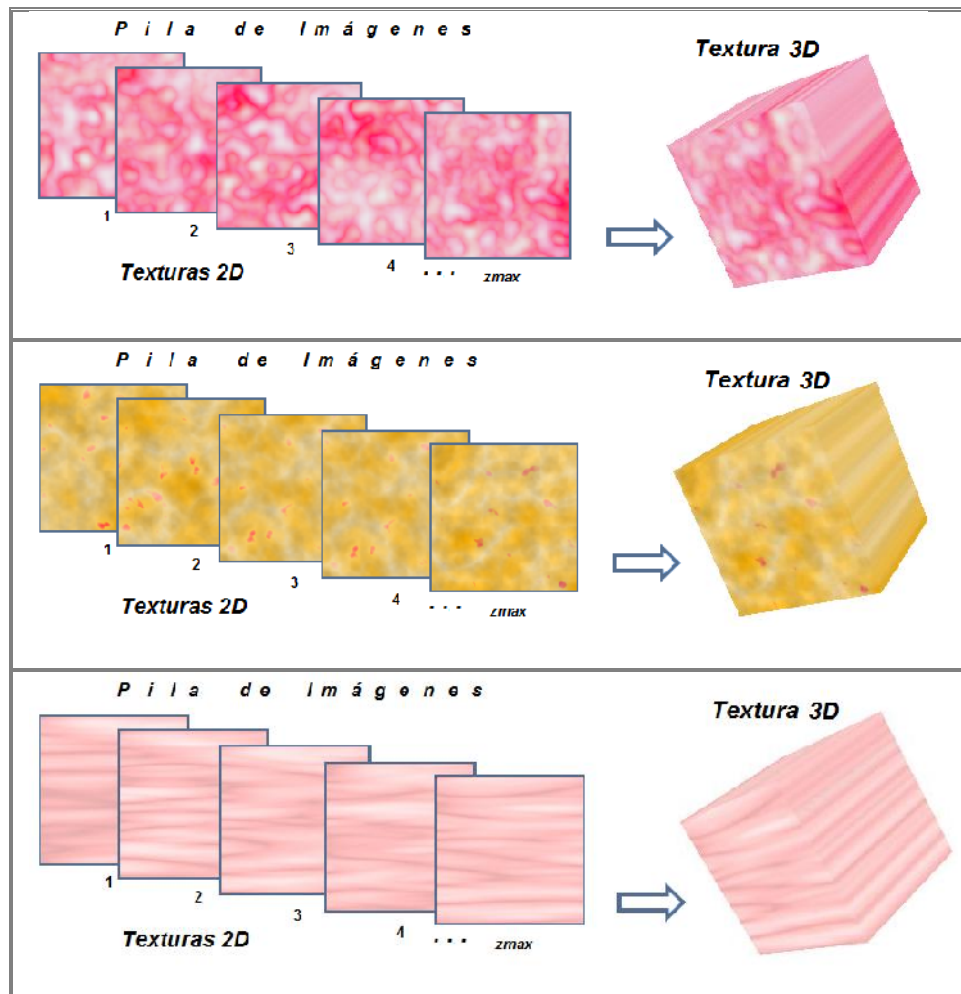


Figura 3.23 Cubos 3D generados con las tres texturas del tejido de la próstata (vascularidades, apariencia amarilla y apariencia rugosa), estos resultados fueron obtenidos con la extensión de los resultados previos en 2D.

3.4.2.2 Rotación de textura 3D del tejido de la cápsula

Aquí es importante discutir el aspecto de orientación de la apariencia rugosa, ya que el mapeado sobre el modelo necesita ubicarse en otra posición para lograr el efecto deseado.

Es por eso que se recurre a una rotación de los ejes (x,y,z) del cubo generado del tejido rugoso de la próstata para cambiar la orientación del aspecto rugosidad del lado correcto del modelo.

En la figura 3.24 podemos observar la rotación y el resultado de la apariencia de la textura.

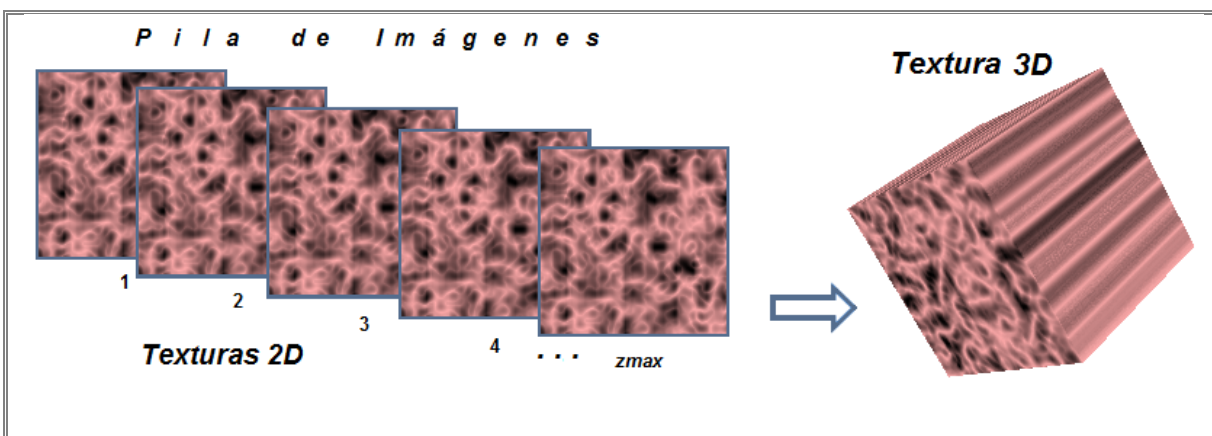


Figura 3.24 Rotación de textura de tejido de la próstata con apariencia rugosa.

3.4.2.3 Integración de las tres texturas 3D con imágenes previas de contorno de próstata

Una vez obtenidos los cubos de cada textura de tejido y realizado la rotación de la textura con apariencia rugosa, se procede a la integración de todo.

En primera instancia se necesita de una pila de imágenes segmentadas de la próstata. Las imágenes representan cortes de la glándula enfocándose principalmente en las regiones de nuestro interés, las cuales finalmente nos definirán el modelo de la próstata final. Dichas imágenes fueron parte del desarrollo del trabajo de tesis del Ing. Valdez Soriano: Generación de modelos de mallas de tetraedros para simuladores quirúrgicos. (Valdéz, 2008)

Ahora bien el algoritmo para la integración de las texturas con las imágenes proporcionadas, se desarrollo para realizar los siguientes pasos:

- 1) Cargar imagen (i) de la pila de imágenes segmentadas de la próstata, en donde *i* nos marcará el numero de imagen dentro de la pila.
Se cuenta con una pila de imágenes de segmentos de próstata las cuales generalmente coinciden con la pila generada de texturas 3D de los tejidos, logrando con ello la integración por segmentos; generando a su vez una tercera pila con las imágenes de textura finales, estas últimas son las que finalmente se utilizaran en el mapeo del simulador.
- 2) Escalar la imagen del segmento de próstata, al tamaño de la textura con los parámetros (xmax,ymax). *Se realiza una escalación sobre la imagen "cargada" de la*

próstata, con la finalidad de lograr el producto uno a uno, entre la textura con la imagen, para evitar distorsiones o resultados no deseados. En la figura 3.25 se muestra gráficamente los pasos 1 y 2 antes mencionados.

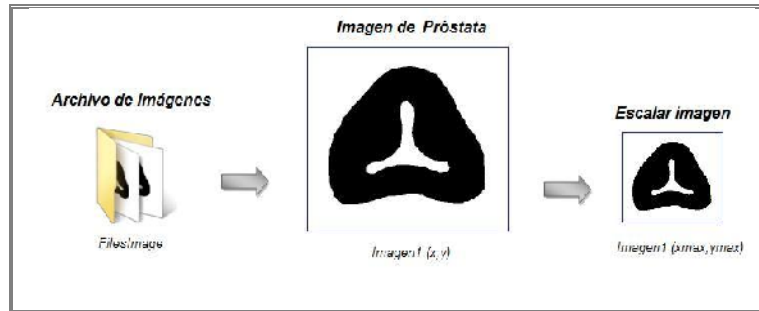


Figura 3.25 Lectura de una imagen de segmento de la próstata, la cual posteriormente se escala al tamaño de la textura generada.

3) Separar contornos exterior e interior. *Se procede a separar los contornos de la próstata con la finalidad de contar con la forma necesaria para proceder a la integración de texturas.*

4) Al contorno exterior se le aplica la operación morfológica de erosión¹⁴. Al contorno interior se le aplica la operación morfológica de dilatación¹⁵. *Se recurrió a las operaciones morfológicas de erosión y dilatación para contar con una franja o transición entre texturas, logrando con ello la continuidad necesaria sobre el modelo virtual.* En la figura 3.26 se visualizan los pasos 3 y 4 previamente comentados.

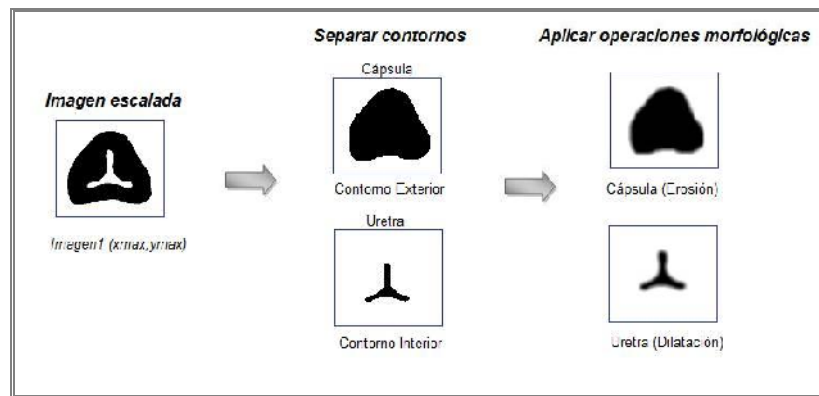


Figura 3.26 Separación de contornos de la imagen de próstata, a las cuales posteriormente se les aplicaran las operaciones morfológicas de erosión y dilatación.

¹⁴ **Erosión:** Operación morfológica que se base en la reducción del nivel de los píxeles del entorno de un objeto.

¹⁵ **Dilatación:** Operación contraria a la erosión. Se basa en el aumento del nivel de los valores de los píxeles en el entorno de los objetos presentes en la imagen, obteniendo como resultado una expansión en los mismos.

- 5) Integrar la textura de tejido de la uretra con vascularidades y la textura de tejido interno de la próstata (amarillo-pardo) con el contorno interno (uretra) de la siguiente manera:

$$\begin{aligned} \textit{Texturizado1} &= A \times B \\ \textit{Texturizado2} &= \bar{A} \times C \\ \textit{Imagen_uretra_texturizada} &= \textit{Texturizado1} + \textit{Texturizado2} \end{aligned}$$

En donde:

$$\begin{aligned} A &= \textit{Imagen_Contorno_Uretra(dilatada)} \\ B &= \textit{Textura_Interna_Próstata(amarillo - pardo)} \\ C &= \textit{Textura_Uretra(vascularidades)} \end{aligned}$$

En la figura 3.27, se puede apreciar la composición y mezcla del contorno “interno dilatado” con las texturas de tejido con apariencia vascular y apariencia en amarillo-pardo. Para finalmente obtener el primer bloque del volumen sólido con el producto de las dos texturas antes mencionadas.

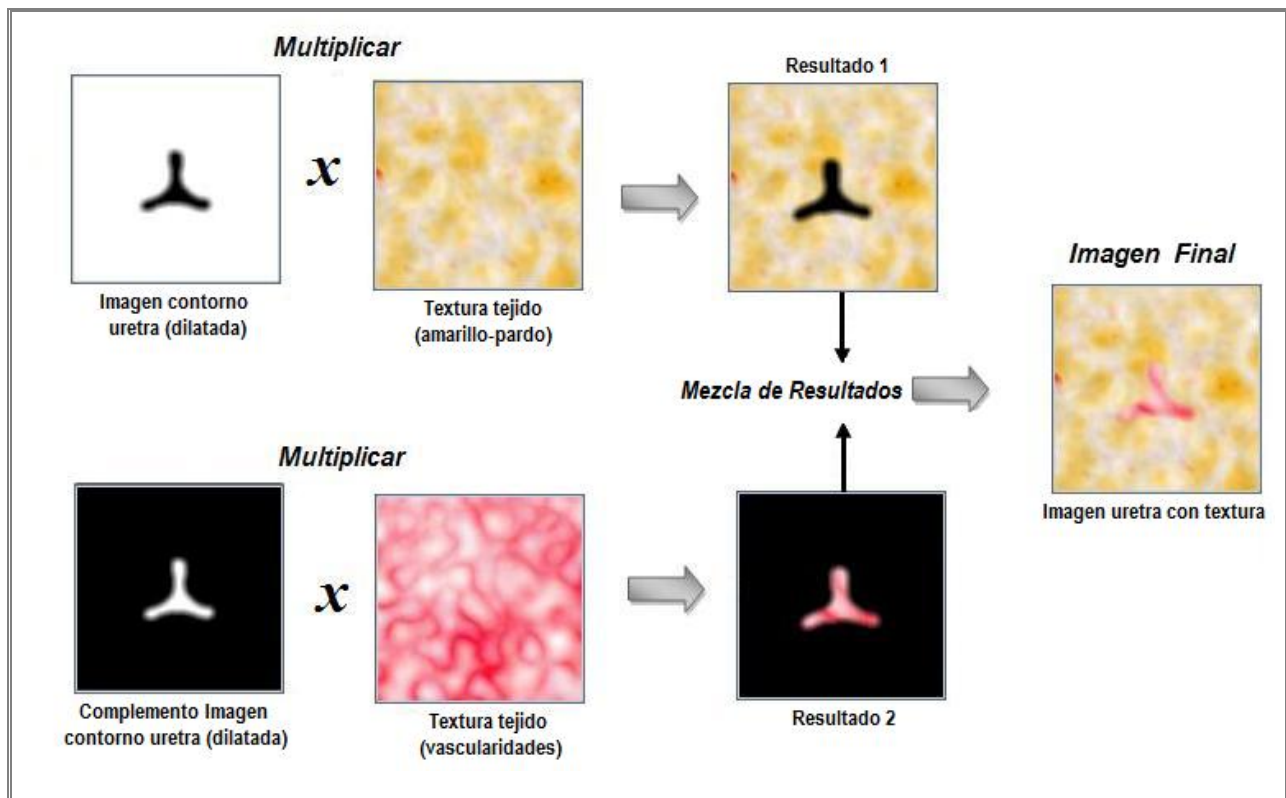


Figura 3.27 Imagen de uretra texturizada. La texturización se logra con la mezcla de la textura de apariencia vascular y la textura de apariencia amarillo-pardo.

- 6) Finalmente se debe integrar y combinar la imagen anterior obtenida (mostrada en la figura 3.27) con la tercera textura (tejido de la cápsula de la próstata, apariencia rugosa), realizando el mismo proceso de la mezcla anterior, el cual quedará de la siguiente manera:

$$\begin{aligned} \textit{Texturizado3} &= D \times E \\ \textit{Texturizado4} &= \bar{D} \times F \\ \textit{Imagen_Próstata_Final} &= \textit{Texturizado3} + \textit{Texturizado4} \end{aligned}$$

En donde:

$$\begin{aligned} D &= \textit{Imagen_Contorno_Próstata(erosionada)} \\ E &= \textit{Textura_cápsula_próstata (rugosa)} \\ F &= \textit{Imagen_uretra_texturizada} \end{aligned}$$

En la figura 3.28, se puede visualizar el resultado de la imagen final generada, la cual integra la forma de la glándula prostática en conjunto con las tres texturas simuladas. Esto se logró con el producto del contorno “exterior erosionado” integrado con la textura con apariencia “rugosa” y el resultado del producto de las dos texturas anteriores (apariencia vascular y apariencia amarillo-pardo).

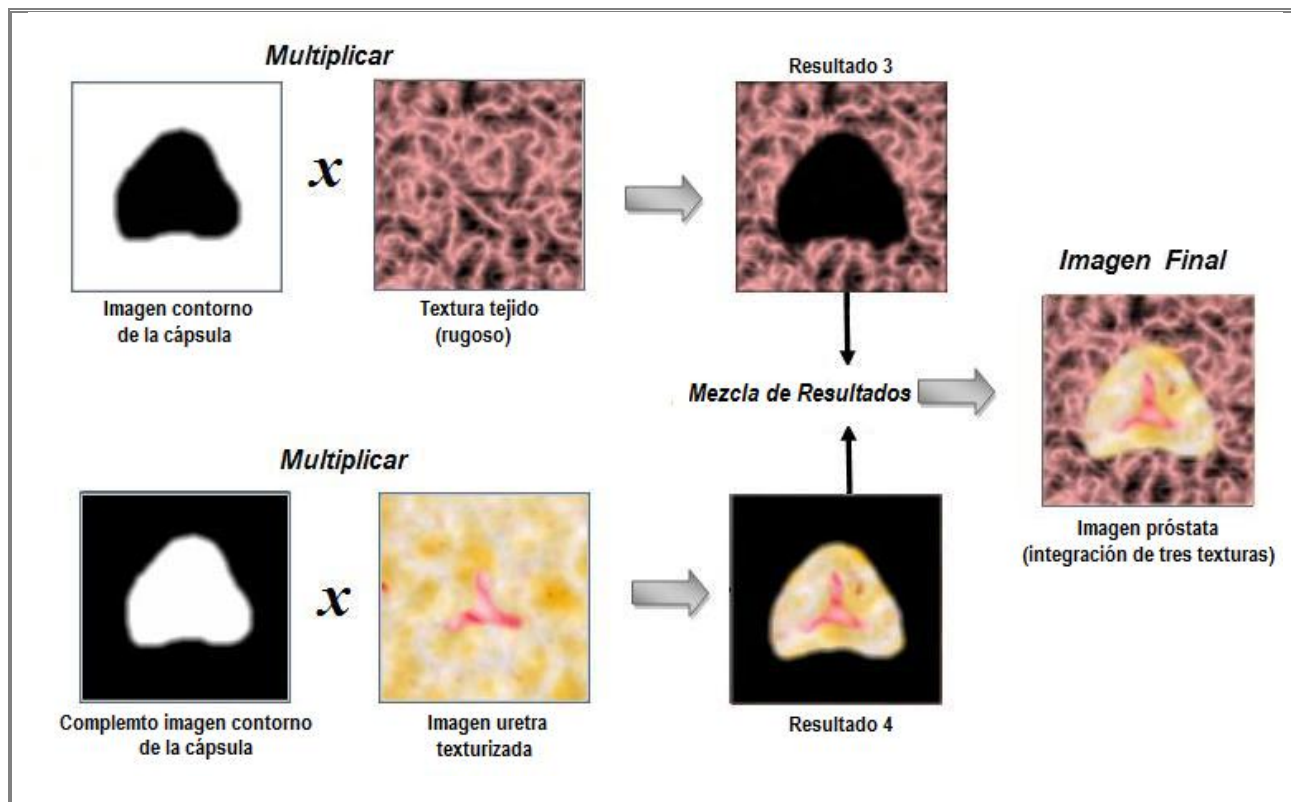


Figura 3.28 Imagen de próstata final texturizada. Se pueden apreciar las tres texturas simuladas, las cuales dan la forma de la glándula prostática.

3.4.2.4 Obtención de la pila: textura volumétrica final de la glándula prostática

Una vez entendido el proceso de integración de texturas y contornos de las imágenes segmentadas de la próstata. Se procederá a generar una pila de imágenes finales de la próstata en conjunto con las tres texturas simuladas. Esto es debido a que dentro del algoritmo, se realiza un ciclo completo de mezcla y combinación hasta terminar con el bloque de texturas simuladas, es decir, hasta llegar a la textura (z_{max}), en donde z_{max} es el valor final del cubo simulado. Por lo cual es importante que los parámetros (x,y,z) cuenten con el mismo valor para asegurar la correcta composición y generación del cubo de texturas.

La idea de este método es ir esculpiendo la figura de la próstata dentro de cada segmento hasta formar un cubo de textura en 3D con la forma de la glándula prostática, obteniendo con ello finalmente la textura volumétrica aplicable al simulador virtual de cirugía. La visualización de la pila de imágenes y el bloque final generado lo podemos visualizar en la figura 3.29.

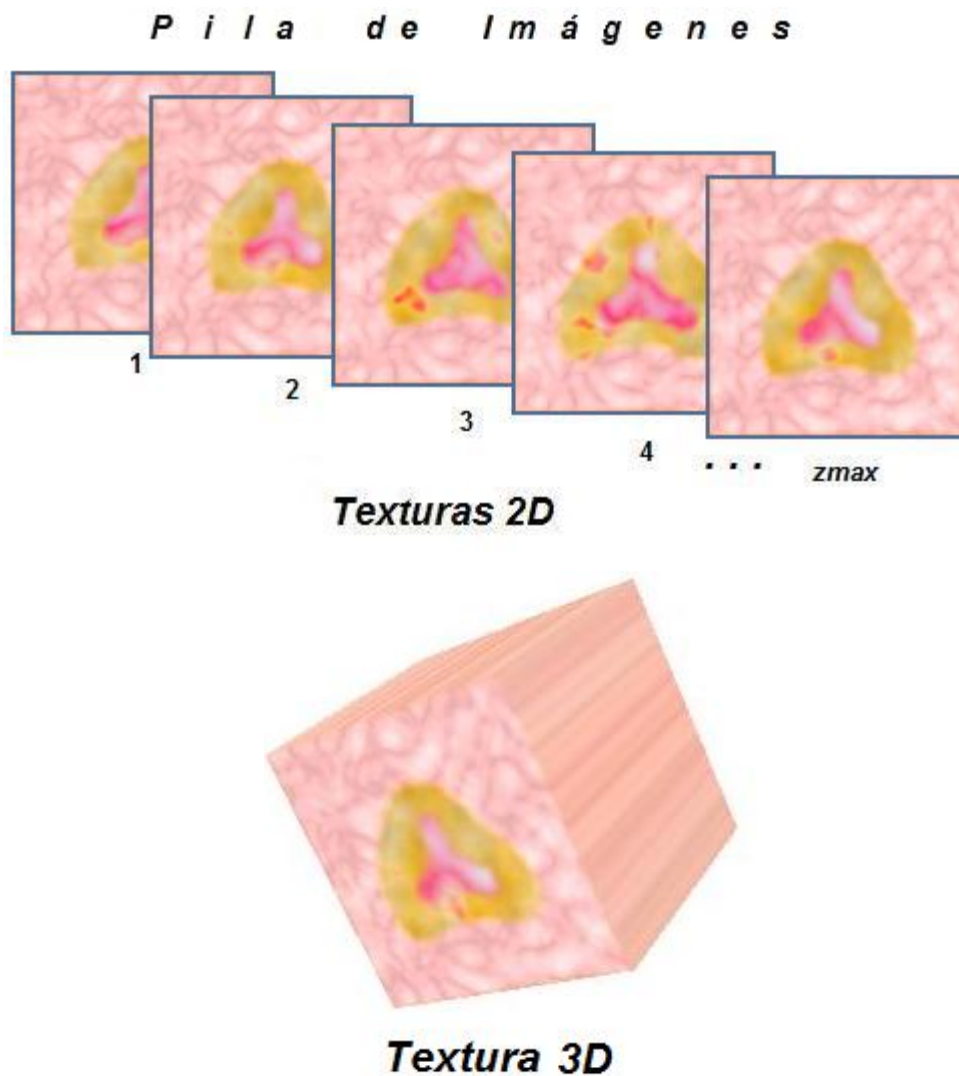


Figura 3.29 Cubo 3D. Próstata texturizada para ser aplicada al simulador virtual de cirugía.

3.4.2.5 Verificación del número de imágenes previas de contorno de próstata

Otro punto importante en la integración de las texturas con los contornos de la próstata, es la verificación del número de imágenes segmentadas en blanco y negro de glándula prostática en comparación con el número de texturas que forman los cubos sólidos generados de las texturas sintéticas.

Esto debido a que si el número de imágenes de los segmentos de la próstata es menor al número de texturas generadas, se tomará la última imagen del segmento de la glándula prostática, copiándola n veces hasta completar el número de texturas generadas para formar el cubo texturizado 3D.

Por lo anterior se debe tomar en cuenta que:

- **xmax, ymax, zmax** → Son los parámetros de altura, anchura y profundidad. Deben tener el mismo valor para poder formar un cubo (x,y,z).
- **num_imagenes_prostata** → El número de imágenes en la pila.

Sí $num_imagenes_prostata < zmax$, entonces:

Se repetirá la última imagen de la próstata, colocando su respectiva textura, como se puede apreciar en la ejemplificación gráfica de la figura 3.30.

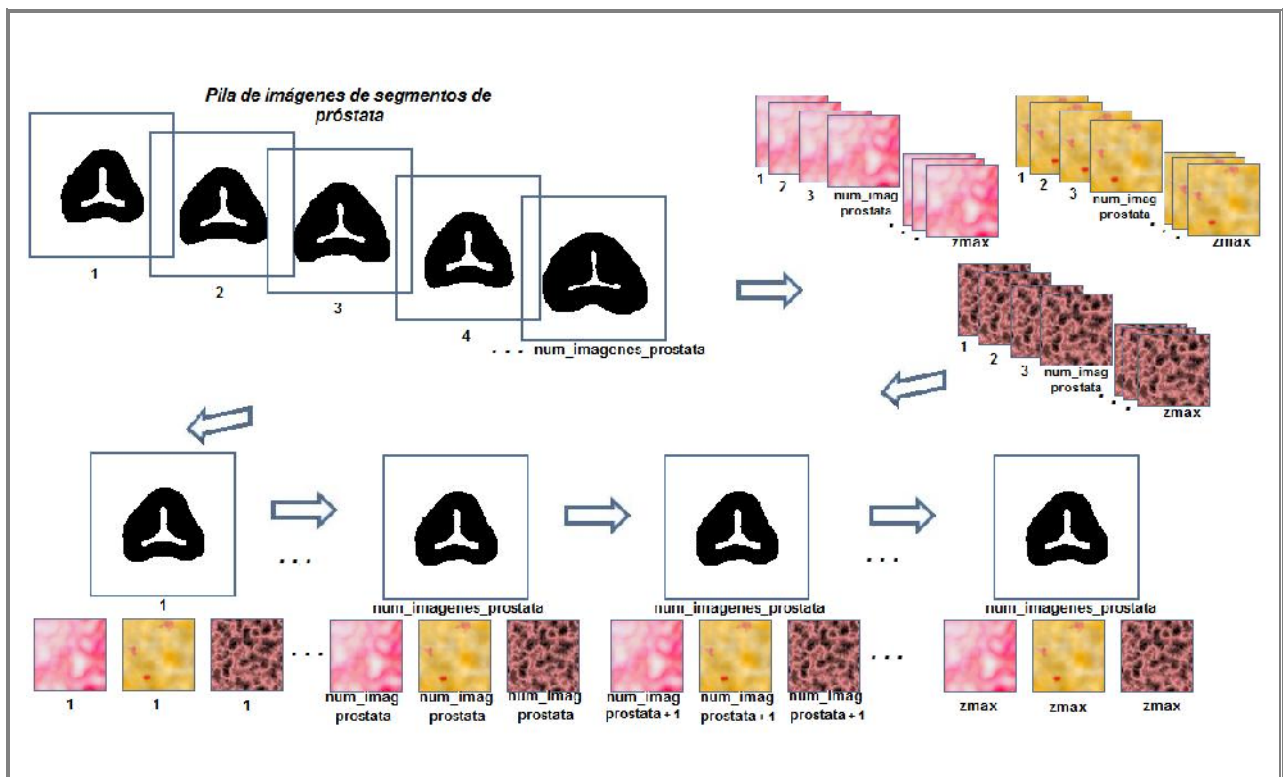


Figura 3.30 Ejemplificación del algoritmo cuando el número de imágenes de próstata en blanco y negro es menor que el cubo a generar.

Si $numsllices \geq zmax$, entonces:

Se generara el cubo con el número exacto de imágenes y texturas, sin la necesidad de agregar algo más dentro del algoritmo.

3.4.2.6 Agregar tapas de textura de tejido con vascularidades en los extremos del cubo

Finalmente se terminará la generación de la textura volumétrica del cubo, agregando a este, dos tapas de textura del tejido de la uretra con apariencia de vascularidades, esto con la finalidad de contar con una apariencia visual externa de vasos y venas antes de realizar un corte o cauterización sobre el tejido, que es cuando debe aparecer la textura amarilla.

En la figura 3.31, se puede apreciar el efecto de colocar las tapas al cubo.

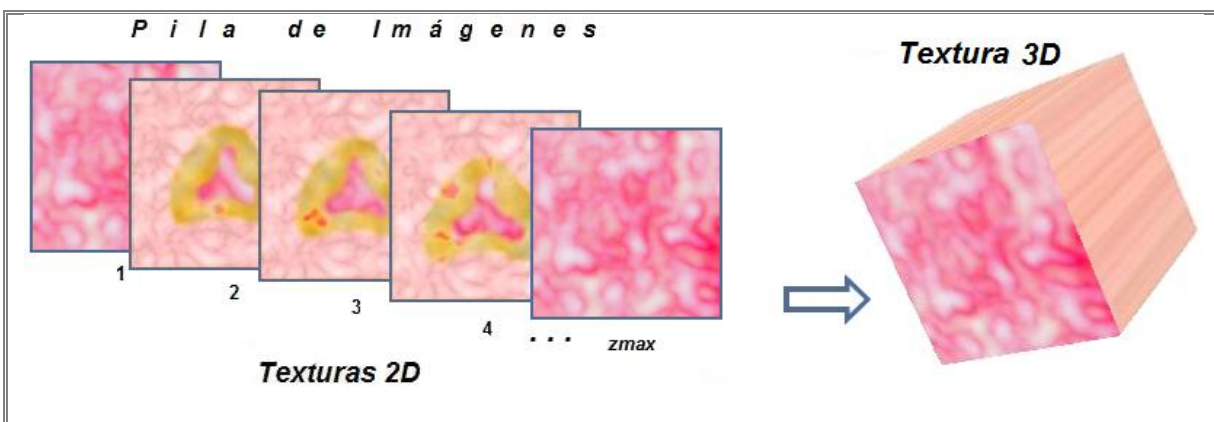


Figura 3.31 Cubo generado en 3D, agregando dos tapas de textura de tejido vascular para lograr una mejor visualización sobre el simulador.

3.5 Mapeo de texturas al simulador de cirugía virtual

Un método común para agregar detalles en la superficie de un objeto virtual es mapear patrones a dicha superficie. Generalmente se define el patrón de textura como una matriz rectangular o como un procedimiento que modifica los valores de intensidad de la superficie. Este procedimiento se llama *mapeo de textura*. (Foley, et al., 1996)

El mapeo de texturas en el simulador de cirugía virtual de próstata, se lleva a cabo con la ayuda de la librería de OpenGL, ya que cuenta con un modulo de carga de texturas en 3D. Los pasos básicos del mapeo de texturas 3D con OpenGL se realiza de la siguiente manera:

- Activar el mapeado de textura
- Especificar la imagen que va a ser utilizada
- Dibujar la escena

Para activar el mapeo de textura se realiza mediante la función **glEnable** eligiendo el parámetro **GL_TEXTURE_3D**.

Para la especificación de la imagen, se utiliza la función **glTexImage3D**, teniendo como parámetros: **GL_TEXTURE_3D**, **GLint level**, **GLint internalFormat**, **GLsizei width**, **GLsizei height**, **GLsizei depth**, **GLint border**, **GLenum format**, **GLenum type**, **const GLvoid *texels**

En donde:

- *Level*: indica el nivel de detalle de la textura. Habitualmente tiene un valor 0.
- *InternalFormat*: indica el número de componentes de color. Color indexado=1, RGB=3, RGBA=4
- *Width*: indica el ancho de la imagen de textura, el valor del parámetro debe proporcionarse en potencia de 2.
- *Height*: indica el alto de la imagen de textura, el valor del parámetro debe proporcionarse en potencia de 2.
- *Depth*: indica la profundidad de la imagen de textura, el valor del parámetro debe proporcionarse en potencia de 2.
- *Border*: indica si se utilizará un borde en la textura con el valor 1 o 0. Usualmente es 0, sin borde.
- *Format*: formato de la información de píxel. Pueden ser: GL_COLOR_INDEX, GL_RGB, GL_RGBA, etc.
- *Type*: tipo de datos de cada píxel. Puede ser uno de los siguientes valores:
 - GL_UNSIGNED_BYTE, GL_BYTE, GL_BITMAP, GL_UNSIGNED_SHORT, GL_SHORT, GL_UNSIGNED_INT, GL_INT o GL_FLOAT.
- *Texels*: Puntero al arreglo tridimensional en donde se almacenan los valores de la textura 3D.

Es importante mencionar que la función *glTexImage3D*, debe encontrarse antes de dibujar los polígonos sobre los que queremos aplicar la textura.

Finalmente para dibujar un objeto en la escena, se debe indicar a cada vértice de este la posición de textura que le corresponde. Esto se hará mediante la función **glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)**.

En donde (s,t,r) indicará la posición sobre el mapa de textura 3D. (Shreiner, et al., 2007)

Algunos resultados del mapeo de la textura volumétrica generada, sobre el simulador los podemos visualizar a continuación en la figura 3.32.

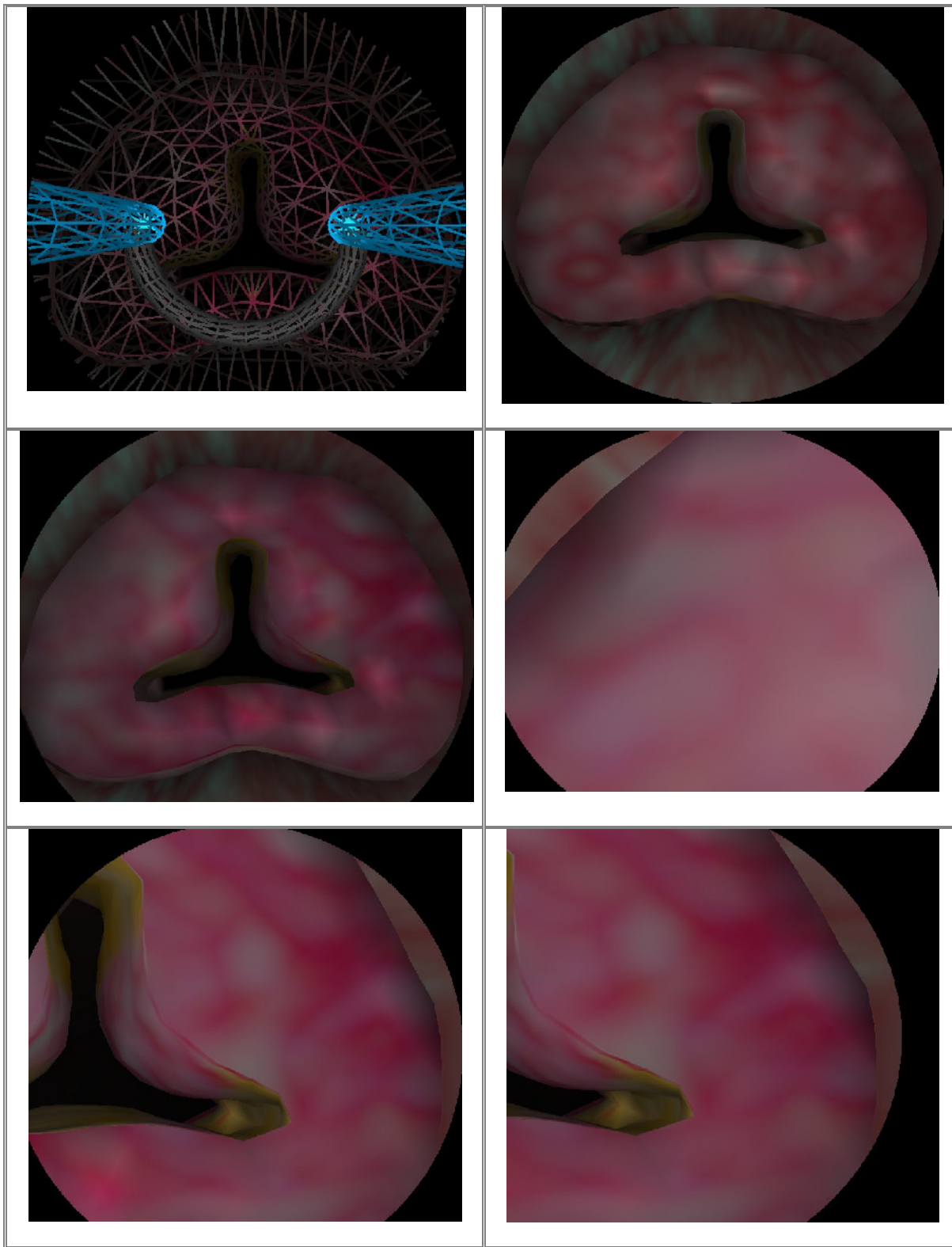


Figura 3.32 Diferentes resultados del mapeado de la textura volumétrica sobre el simulador de cirugía de próstata. Se pueden apreciar las vascularidades de la textura, apareciendo zonas con venas "inflamadas".

4

CAPÍTULO

4. PRUEBAS Y RESULTADOS OBTENIDOS EN LA GENERACIÓN DE TEXTURAS PROCEDURALES

En este capítulo se explicaran las pruebas y resultados de la generación de las texturas procedurales utilizando el software Matlab para el desarrollo del código.

Las texturas se generaron con las funciones base: *ruido de Perlin*, *función movimiento Browniano fraccional (fBm)*, *turbulencia*, *turbulencia modificada* y *terreno heterogéneo*, como se explicó en el capítulo anterior.

Posteriormente se mostraran las tablas con las pruebas y resultados de cada función variando cada uno de los parámetros de las mismas para cada textura generada, logrando visualizar los cambios que sufre la imagen final obtenida.

Finalmente se mostraran algunas imágenes del simulador virtual de cirugía con la texturización de la síntesis de texturas generada en este trabajo.

4.1 Matlab: generación de texturas

La implementación de los algoritmos generadores de texturas procedurales bien pueden ser programados en lenguajes de alto nivel, o bien en tiempo real en una tarjeta aceleradora de gráficos. En este trabajo se busco minimizar tiempo en el periodo de pruebas y correcciones de las funciones básicas generadoras de textura, así como pre-calcular fuera de línea las texturas para evitar realizar cálculos innecesarios en el simulador en tiempo real. Por tal motivo se ha resuelto utilizar la herramienta de software Matlab que con ayuda de su toolbox¹⁶ de procesamiento de imágenes acortará el tiempo de implementación haciendo más sencilla la realización de las pruebas necesarias.

La caja de herramientas de procesamiento de imágenes contiene un conjunto de las funciones más conocidas para trabajar con imágenes binarias, transformaciones geométricas, morfología y manipulación del color. Adecuando perfectamente la herramienta a las condiciones que necesitamos para modelar nuestras texturas.

4.2 Resultados del modelado de texturas procedurales al variar sus parámetros

Se realizaron las pruebas correspondientes a cada una de las funciones básicas presentes en la generación de cada una de las texturas. Quedando en el siguiente orden:

Textura del tejido de la uretra con vascularidades

- Función Terreno Heterogéneo(h, lacunaridad, offset, octavas)
- Función Turbulencia (escala, octava_inicial, octavas, factor)
- Función fBm (escala, h, lacunaridad, octavas)

Textura del tejido interno de la próstata con apariencia amarillo - pardo

- Función Turbulencia (escala, octava_inicial, octavas)
- Función fBm1 (escala, h, lacunaridad, octavas)
- Función fBm2 (escala, h, lacunaridad, octavas)

Textura del tejido de la cápsula de la próstata con apariencia rugosa

- Función Turbulencia modificada (escala, octava_inicial, octavas)

¹⁶ **Toolbox:** MATLAB dispone de un amplio abanico de programas de apoyo especializados, denominados Toolboxes, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el 'toolbox' de proceso de imágenes, señal, control, estadística, análisis financiero, matemáticas simbólicas, redes neurales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc.

En las siguientes secciones, se ejemplificarán los resultados más significativos que se obtuvieron al modificar los valores de los parámetros de cada función procedural básica, en cada textura generada. Esto nos fue realmente útil para poder conocer las variaciones que pueden existir en la generación de dichas texturas, pudiendo con ello conseguir una base de imágenes de texturas sintéticas, aplicables al modelo de cirugía virtual y ser capaz de presentar varias “texturizaciones”, para lograr tener varios efectos sobre el simulador.

4.2.1 Textura del tejido de la uretra con vascularidades

Como se explico en el capítulo anterior, la generación de esta textura se hizo bajo la utilización de las tres funciones básicas: *terreno heterogéneo*, *turbulencia* y *movimiento Browniano fraccional (fBm)*. A fin de conocer las variaciones que causa la modificación de los parámetros de entrada de cada función, se sintetizaron algunos de los resultados obtenidos, los cuales se mostrarán en las secciones siguientes.

4.2.1.1 Función terreno heterogéneo

Es importante recordar la utilización de esta función, en la generación de la textura de vascularidades, ya que ayudo a definir una zona dentro de la textura con una apariencia más “inflamada”, simulando así una zona de un color más intenso, en la cual en un futuro servirá para disparar el efecto de la simulación de sangrado, cuando se toque dicha zona. Los parámetros de esta función procedural son:

Funcion Terreno_heterogeneo (h, lacunaridad, offset, num_octavas)

Los resultados que se presentan a continuación, se lograron al ir modificando el valor de cada uno de estos parámetros.

Parámetro H

El parámetro H nos proporciona una “rugosidad” en la textura final generada. Cuando se cuenta con un valor pequeño en dicho parámetro, se obtendrá una textura con una apariencia demasiado “rugosa”, por el contrario al tener un valor muy grande, se generará una textura mucho más “lisa”.

En la tabla 1, se muestran los resultados al variar el valor de *H* comenzando con un *valor en H = 0* hasta un valor en *H= 4.5*. Se puede apreciar primeramente el resultado de la función terreno heterogéneo en escala de grises al variar el parámetro H. Posteriormente se encuentra el resultado de la textura sintética final generada, apreciando las diferencias y los distintos efectos que podemos lograr al manipular este parámetro.

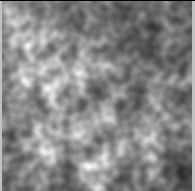
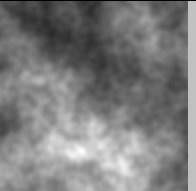

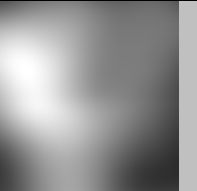
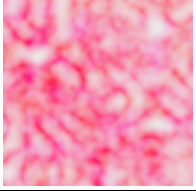
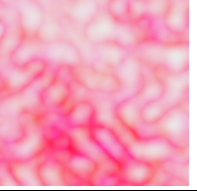
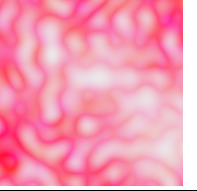
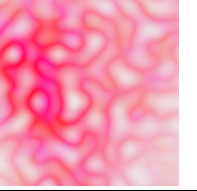
H	0	0.8	2.2	4.5
<i>Textura función terreno heterogéneo</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 1. Resultados de la modificación del parámetro H, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.

Cabe mencionar que los valores de los demás parámetros, se colocaron los mismos valores que sirvieron en la generación de esta textura, mostrados en el capítulo anterior. Y en conclusión al analizar la *tabla 1*, podemos apreciar, que efectivamente este parámetro nos altera la “rugosidad” en la textura obtenida con la función básica procedural. Y los efectos sobre la textura final, nos arrojan que se necesita un valor “grande” en este parámetro para tener una mejor definición de la zona “congestionada”. Sin embargo la decisión del valor del parámetro estará también determinado por las condiciones y características finales que queramos obtener dentro de la síntesis de la textura.

Parámetro LACUNARIDAD

El parámetro lacunaridad nos genera en la textura huecos o “espacios vacíos”, incrementándose el número de huecos en la textura cuando el valor del parámetro es un número “grande”. En la *tabla 2* podemos apreciar los resultados que se obtuvieron al variar dicho parámetro.

Como el caso anterior los valores de los demás parámetros quedan definidos por los mismos valores que se utilizaron en la generación de esta textura.

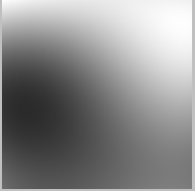

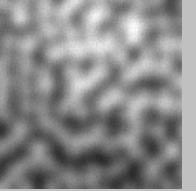
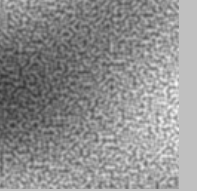
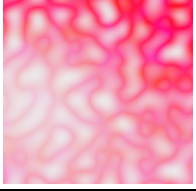


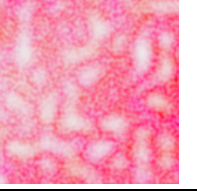
Lacunaridad	0.4	2.2	10	50
<i>Textura función terreno heterogéneo</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 2. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.

Nuevamente podemos comprobar en los resultados, el efecto al variar el parámetro de lacunaridad en la función básica procedural, logrando tener una textura con más “huecos” al agregar un valor de *lacunaridad* = 50.

En los resultados de la obtención de la textura final de vascularidades, podemos apreciar que se necesita un valor “pequeño” dentro de este parámetro, ya que agrega en la textura las características que más se asimilan a lo que se desea obtener.

Parámetro OFFSET

Este parámetro agrega en la textura de la función básica terreno heterogéneo, la elevación y declive sobre una superficie de terreno (por eso el nombre de la función, ya que cuenta con zonas altas y bajas dentro de una misma textura, muy útil en la generación de terrenos).

Sin embargo en nuestro caso y analizando los resultados que obtenemos en la tabla 3, podemos observar que este parámetro nos proporciona un efecto más denso dentro de la textura cuando el valor del parámetro es “grande”, por el contrario al contar con un valor “pequeño” podemos ver que el resultado en la textura es más “ligero”.

En los resultados de la textura final de vascularidades podemos observar que los resultados son más aceptables cuando se cuenta con un valor “pequeño” en este parámetro, ya que la textura tiene mejor definidas las vascularidades y la zona de congestionamiento se visualiza más claramente.

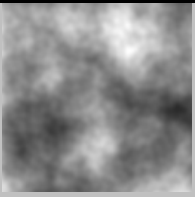
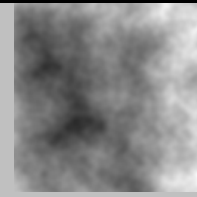
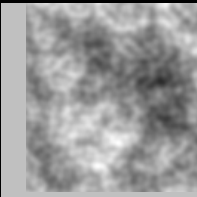
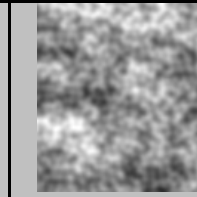
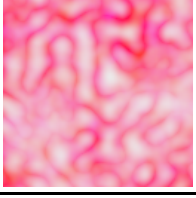

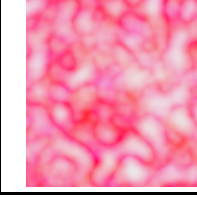
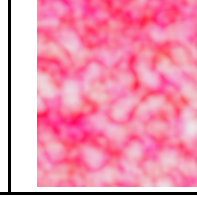
Offset	0.8	2.2	10	50
<i>Textura función terreno heterogéneo</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 3. Resultados de la modificación del parámetro OFFSET, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.

Parámetro Número de OCTAVAS

Finalmente el último parámetro de esta función procedural, es el valor del número de octavas. Este parámetro, se utiliza para definir el número de iteraciones dentro del algoritmo de la función generadora; en el cual en cada iteración suma y agrega más detalles a la textura final generada.

Pudiendo comprobar los resultados en la tabla 4, se verifica que al agregar un número mayor en este parámetro, podemos ver el resultado de la función procedural con

más detalles en comparación a la textura obtenida con un valor del parámetro más pequeño.

En los resultados de la textura final generada, parece que este parámetro no modifica significativamente los resultados al manipular el valor del parámetro, ya que en todos los casos podemos apreciar, que cuentan con las características de venas bien definidas, apreciando también la zona de “inflamación” sin complicaciones.

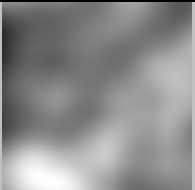
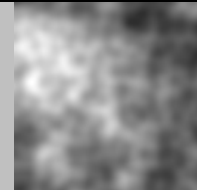
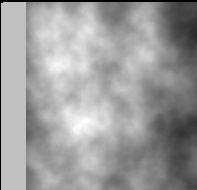
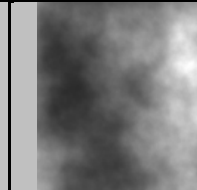


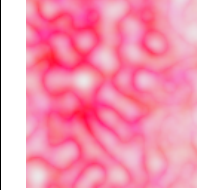
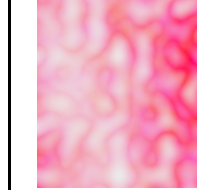
Octavas	1.5	3.67	10.25	12.14
<i>Textura función terreno heterogéneo</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 4. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural terreno heterogéneo en la generación de la textura con vascularidades.

4.2.1.2 Función turbulencia

La segunda función que nos ayuda en la generación de esta textura vascular es la función conocida como función de turbulencia. Como se ha mencionado en el capítulo anterior, esta función nos da la simulación de las vascularidades que debe contener la textura final generada. Por ende es importante conocer las alteraciones que sufre la textura al modificar los parámetros de esta función procedural.

Comenzamos por recordar cuáles son sus parámetros:

Funcion turbulencia (escala, octava_inicial, num_octavas, factor)

Al igual que los resultados de las tablas anteriores, se obtendrán los resultados de las diferencias que se obtienen al modificar uno a uno el valor de cada parámetro de esta función, recordando de igual forma que los valores para los demás parámetros permanecerán con el mismo valor que se utilizó para generar la textura “base” de vascularidades.

A continuación se explican los resultados obtenidos para cada parámetro de esta función procedural.

Parámetro ESCALA

El parámetro escala es uno de los parámetros más interesantes que tiene esta función, ya que con este, podemos controlar el tamaño de los detalles en la textura de la función procedural. Asiendo una analogía podemos visualizar el resultado, como si se le

aplicara un acercamiento o alejamiento a la textura (efecto de zoom). Teniendo la posibilidad de poder jugar con este valor, dependiendo de las características finales que queramos obtener.

En la tabla 5, podemos observar los resultados que se obtuvieron al modificar el parámetro *escala*, primeramente visualizaremos las texturas resultados de la función turbulencia (segundo renglón de la tabla), posteriormente los resultados con los efectos que hay en la textura final de vascularidades (tercer renglón de la tabla).


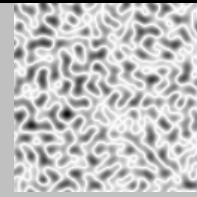
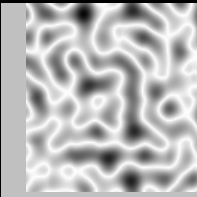
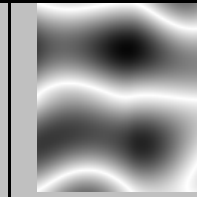
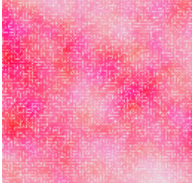


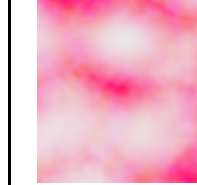
Escala	16	64	128	512
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 5. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural turbulencia en la generación de la textura con vascularidades.

Parámetro OCTAVA INICIAL

El siguiente parámetro de esta función procedural, es el conocido como *octava inicial*. Este parámetro está muy relacionado con el parámetro *números de octavas*, ya que los dos marcan las iteraciones y formas que se generaran en la textura simulada.

En particular esta función tiende a cambiar significativamente al modificar su valor, ya que las formas que se generan son diferentes entre sí, como se puede visualizar en los resultados de la tabla 6. Es importante mencionar que se debe utilizar siempre un valor entero positivo menor que el valor del parámetro de número de octavas.

Por lo cual para lograr tener una apariencia de vascularidades, es importante que el valor de este parámetro y el valor del parámetro *número de octavas*, difieran muy poco entre sí. De hecho se recomienda colocar un valor entero en el parámetro octava inicial, por ejemplo: *octava_inicial=4*; y en el parámetro número de octavas agregar el mismo valor del parámetro octava inicial, sumado a este, solo una parte fraccional, por ejemplo: *num_octavas=4.27*.

Los resultados que se obtienen cuando estos parámetros difieren mucho entre sí, bien puede ser usados para la simulación de otro tipo de efectos.

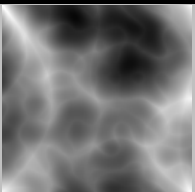
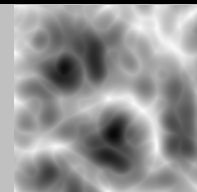
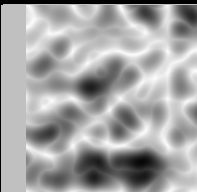
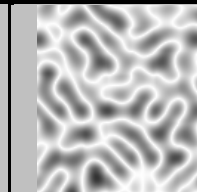

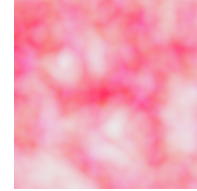

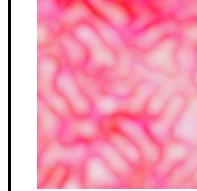
Octava inicial	1	2	3	4
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 6. Resultados de la modificación del parámetro OCTAVA INICIAL, dentro de la función procedural turbulencia en la generación de la textura con vascularidades.

Parámetro Número de OCTAVAS

El valor de este parámetro como bien se menciona, está muy relacionado con el parámetro *octava inicial*. Es importante que el valor que sea colocado en este parámetro, varíe muy poco con respecto al valor del parámetro *octava inicial*, para lograr el aspecto de la apariencia vascular.

Los resultados al cambiar los valores de este parámetro se pueden apreciar en la tabla 7. Encontrando que al momento de incrementar el valor, tiende a formar un apariencia diferente y no muy definida de las características vasculares.

Sin embargo se pueden utilizar los diferentes resultados para lograr otros aspectos, en la creación de texturas aplicables a otros modelos virtuales.

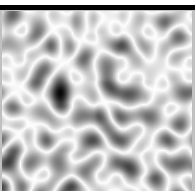
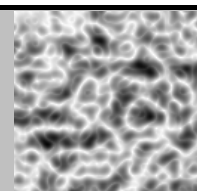
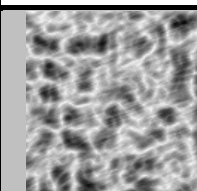
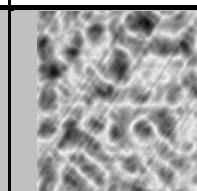
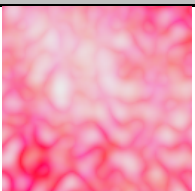
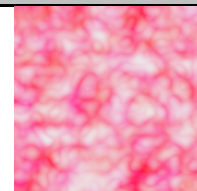
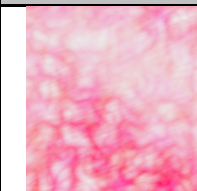
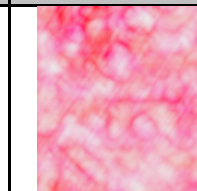
Octavas	4.2	5.42	8.76	12.45
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 7. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural turbulencia en la generación de la textura con vascularidades.

Parámetro FACTOR

El ultimo parámetro de esta función esta dado por un valor constante denominado *factor*. El valor de este parámetro se ha utilizado para definir las tonalidades finales de la

textura generada, el cual se puede manipular para obtener una serie de texturas de diferentes matices como se puede apreciar en la tabla 8.


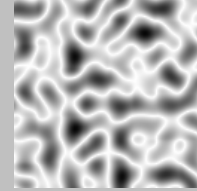
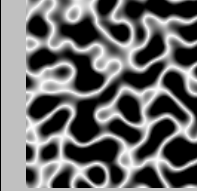
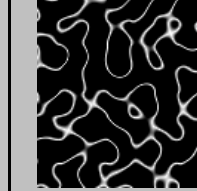
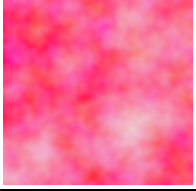

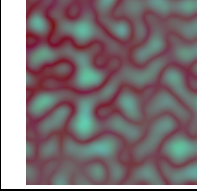
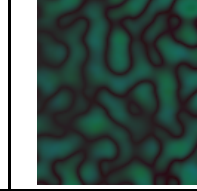
Factor	0.5	3.2	10	30
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 8. Resultados de la modificación del parámetro FACTOR (valor constante), dentro de la función procedural turbulencia en la generación de la textura con vascularidades.

Es importante mencionar que dicho valor siempre debe ser mayor a cero para lograr que nos proporcione un resultado visible.

Lo que se hace dentro del algoritmo con este parámetro, es multiplicar los valores del resultado de la función básica turbulencia agregando “brillo” o contraste a la textura generada, con lo cual es posible obtener una serie de diferentes resultados al realizar la conversión al modelo RGB. En nuestro caso se utilizó el valor de $factor=3.24$; con lo cual se obtuvieron las tonalidades deseadas y las zona de vascularidades se pueden apreciar claramente.

4.2.1.3 Función movimiento Browniano fraccional (fBm)

La última función que se utilizó en la generación de esta textura, es la función procedural, conocida por sus siglas en inglés, como *fBm*. El resultado de esta función se debe colocar en el canal de matiz (H), del modelo HSV para posteriormente realizar la conversión al modelo RGB. Por lo cual podemos deducir que está función participa en la obtención de los colores de la textura final generada de vascularidades.

Los parámetros de esta función son:

Funcion fBm (escala, h, lacunaridad, octavas)

Se procederá a la explicación de los resultados que se obtienen con la modificación de los valores de cada uno de sus parámetros de entrada. Como antes se ha mencionado, los valores de los demás parámetros quedaran definidos con los mismos valores que se utilizaron en la generación de la textura de vascularidades, vistos en el capítulo anterior.

Nota: Los resultados obtenidos de la función procedural fBm, se les modifíco el contraste de la imagen con fines de impresión de la tesis, ya que la visualización de esta textura es un color blanco con pequeñas manchas en gris claro, con lo cual dificulta su apreciación visual.

Parámetro ESCALA

El parámetro *escala* es uno de los parámetros más vistosos en la obtención de las texturas generadas, ya que este, agrega el efecto de alejamiento o acercamiento a la textura, logrando con ello un “efecto zoom” dentro de la textura sintética.

Con los resultados mostrados en la tabla 9, podemos visualizar las características que nos agrega esta función a la textura de vascularidades final. Logrando con este las tonalidades de **rosas - rojos** que deben estar presentes en esta textura final de vascularidades, para asemejarse a la textura de tejido real, siendo las venas las que poseen dichos colores. Igualmente los resultados obtenidos de la función procedural nos muestran los efectos que se alcanzan con este parámetro, *efecto zoom*.

El valor que se asigne a este parámetro dependerá en gran medida de las condiciones que necesitemos en la textura final deseada.

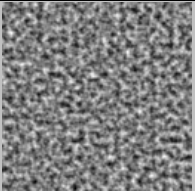
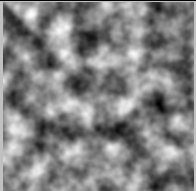
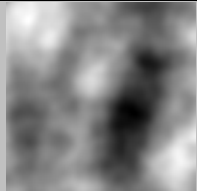
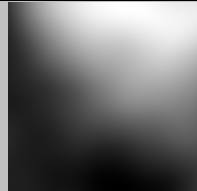
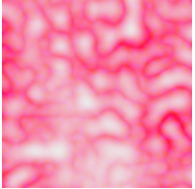

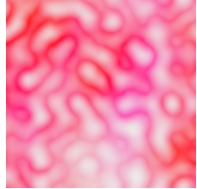

Escala	4	16	64	256
<i>Textura función fBm</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 9. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural fBm en la generación de la textura con vascularidades.

Parámetro H

Parámetro H, define el grado de rugosidad en la textura simulada. Cuando el valor en este parámetro sea “pequeño” es decir un rango entre los valores 0 a 2 , las texturas alcanzaran un grado *mayor de rugosidad*. Mientras que para los valores arriba de 2 , alcanzara una textura con una apariencia con menos grado de rugosidad.

En los resultados obtenidos de la tabla 10, podemos visualizar el efecto que agrega la modificación de este parámetro. Sin embargo en los resultados que se obtienen de la función final de vascularidades no parece haber mucho cambio, dado que las texturas mostradas no cambian mucho entre sí, contando todas con los elementos necesarios para la generación de esta textura, es decir: color rosa – roja, con una zona definida de “inflamación” y las formas vasculares dentro de la textura.

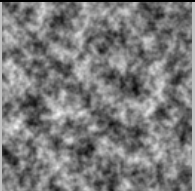
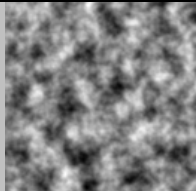
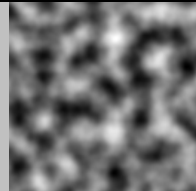
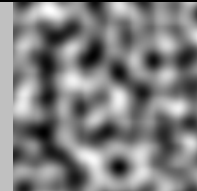

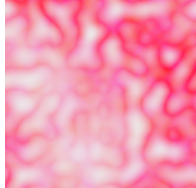


H	0.2	0.8	2.2	10
Textura función fBm				
Textura de tejido final generada (vascularidades)				

Tabla 10. Resultados de la modificación del parámetro H, dentro de la función procedural fBm en la generación de la textura con vascularidades.

Parámetro LACUNARIDAD

Este parámetro como bien se recordará, nos permite agregar a la textura una serie de lagunas o huecos. Si el valor del parámetro es “pequeño”, habrá una cantidad menor de huecos comparado con un valor mayor en el cual habrá una cantidad mayor de huecos o “espacios vacíos” sobre la textura final generada.

Los resultados que se obtuvieron al modificar este parámetro se puede apreciar en la tabla 11. Obteniendo la apariencia de una textura con más espacios vacíos al incrementar el valor de esta parámetro. Bien se puede ir variando los valores de este parámetro, sobre todo para conocer la visualización final de la textura vascular generada. En este caso el resultado con el valor de lacunaridad en 2.2, es el más cercano a lo que deseamos, ya que cuenta con una mejor definición de una zona de “inflamación”, lo que se traduce en un color más intenso en una zona específica.

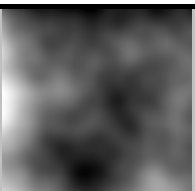
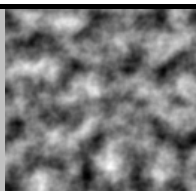
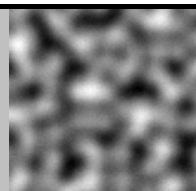
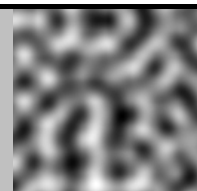

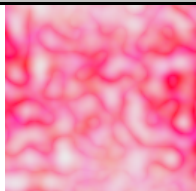
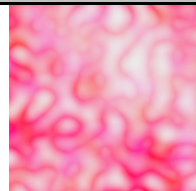
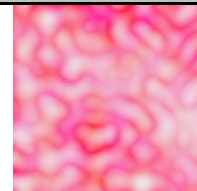
Lacunaridad	0.4	2.2	10	50
Textura función fBm				
Textura de tejido final generada (vascularidades)				

Tabla 11. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural fBm en la generación de la textura con vascularidades.

Parámetro Número de OCTAVAS

El último parámetro de esta función nos lo da el número de octavas, el cual es el responsable del número de iteraciones que hay dentro del algoritmo generador, en el cual con un valor mayor en este parámetro, se podrán obtener más pequeños detalles dentro de la textura.

Los resultados obtenidos en la manipulación del valor de este parámetro, lo podemos apreciar en la tabla 12. Aunque el resultado final de la textura vascular no parece sufrir un cambio significativo en la manipulación de este parámetro.

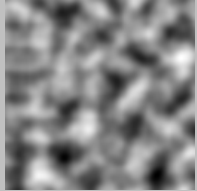
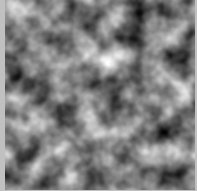
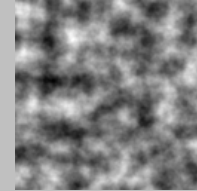
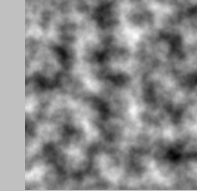




Octavas	1.5	3.43	5.78	10.45
<i>Textura función fBm</i>				
<i>Textura de tejido final generada (vascularidades)</i>				

Tabla 12. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural fBm en la generación de la textura con vascularidades.

4.2.2 Tejido colaginoso interno de la próstata (apariencia amarillo-pardo)

Las pruebas de esta textura se realizaron con el uso de las funciones procedurales básicas: *turbulencia y movimiento Browniano fraccional (fBm)*. Cabe recordar que la generación de esta textura, se utilizó la función fBm para obtener dos distintos resultados, en los cuales cada uno cumple con algún efecto distinto dentro de la generación final de la textura de apariencia amarillo-pardo.

Los resultados se obtuvieron con la modificación de los valores de cada parámetro de las funciones básicas. Estos valores se fueron cambiando uno a uno, de cada función procedural básica, dejando los demás parámetros con el valor por omisión que se utilizó en la generación de la textura 2D de apariencia amarillo-pardo; hacerlo de esta manera ayudó a identificar las diferencias principales, logrando tener más control sobre el resultado final que se desee en la generación de la textura, para ser aplicada posteriormente al modelo virtual de cirugía.

A continuación se presentan las tablas con los resultados más significativos de los cambios que sufre la textura final generada al modificar los valores de los parámetros de cada una de estas funciones.

4.2.2.1 Función turbulencia

Esta función fue muy útil por contar con las características para ser la textura “base” del tejido interno de la próstata, ya que es la que cuenta con las condiciones visuales de una apariencia esponjosa parecida a la imagen médica real.

Es importante mencionar que a pesar de ser la función de turbulencia que se utilizó para agregar las vascularidades en la generación de la textura anterior; en este caso se utilizó por contener una apariencia visual muy parecida a la mostrada en la imagen médica real del tejido interno de la próstata.

Para lograr estas diferencias en el resultado de la generación de dichas texturas procedurales, solo bastó con modificar los valores de sus parámetros de entrada, sobre todo sus parámetros de *octava inicial* y *número de octava*, logrando con esto obtener una textura procedural totalmente diferente. De hecho esto suele ser una de las ventajas más significativas de las funciones procedurales, ya que con solo manipular algunos de los parámetros de sus entradas podremos crear características y diferentes efectos aplicables a otros modelos virtuales de una misma función procedural básica.

Pues bien, los parámetros de esta función son:

Funcion turbulencia (escala, octava_inicial, num_octavas)

A continuación se desarrollaran los comentarios y los resultados obtenidos al variar el valor de cada parámetro de esta función.

Parámetro ESCALA

Este parámetro causa el efecto de zoom sobre la textura, ya que pareciera que se realiza un alejamiento o acercamiento sobre la misma. Como bien se mencionó anteriormente, esta función nos da la característica “base” de la textura final generada.

En la tabla 13, podemos apreciar los resultados que se obtienen al variar el valor de este parámetro. En el segundo renglón de la tabla 13 se pueden observar los resultados de la función básica de turbulencia en escala de grises, apreciando el efecto *zoom* con el incremento del valor.

En el tercer renglón de la tabla 13, podemos apreciar el resultado de la textura final obtenida conservando la forma de la función procedural básica, añadiendo a está solo las tonalidades en blanco, amarillo y rojo.

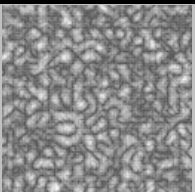
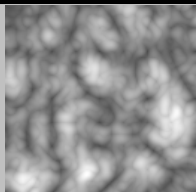
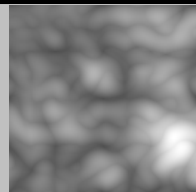
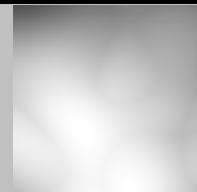
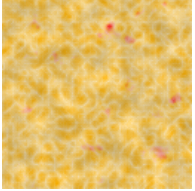
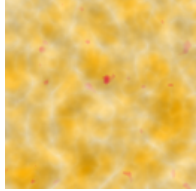
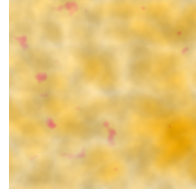
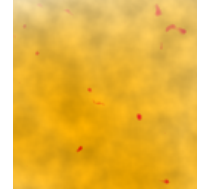
Escala	16	64	128	512
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (apariciencia amarillo-pardo)</i>				

Tabla 13. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural turbulencia en la generación de la textura con apariencia amarillo-pardo.

Parámetro OCTAVA INICIAL

Este parámetro se relaciona mucho con el parámetro *número de octavas* de esta misma función, dado que ambos nos definirán la apariencia final de la textura generada.

Siendo muy útil el número que aquí se defina, ya que con esto se marcará la forma final de la textura. Para este caso es necesario contar con un valor que difiera significativamente entre el valor de este parámetro y el valor del parámetro *número de octavas*. Para entender mejor esto, se deben visualizar los resultados de la tabla 14.

Así nuestro valor del parámetro número de octavas es: *num_octavas=4.5*. Comparando los resultados en la tabla 14, para *octava inicial= 1*, tenemos una textura tipo “nube” con una apariencia de esponjosidad. Sin embargo para el valor de la octava inicial= 4, las cosas cambian drásticamente, ya que contamos con el efecto de las vascularidades, función que nos fue muy útil en la generación de textura anterior, pero para este caso este resultado no nos sirve, por lo cual se opta por trabajar con un valor menor en este parámetro.

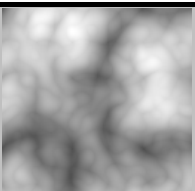
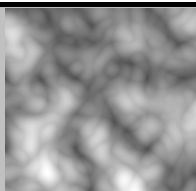
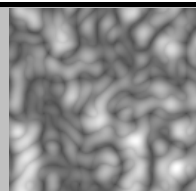
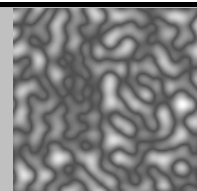
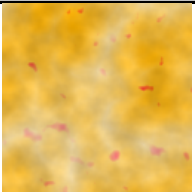
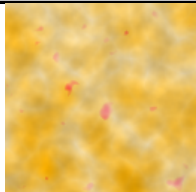
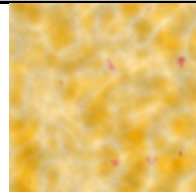
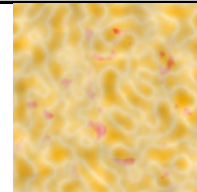
Octava inicial	1	2	3	4
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (apariciencia amarillo-pardo)</i>				

Tabla 14. Resultados de la modificación del parámetro OCTAVA INICIAL, dentro de la función procedural turbulencia en la generación de la textura con apariencia amarillo-pardo.

Parámetro Número de OCTAVAS

Finalmente llegamos al último parámetro de esta función, *número de octavas*, el cual nos agregará más detalles, cuando se vaya incrementado el valor de dicho parámetro.

Los resultados aquí presentados se pueden visualizar en la tabla 15, pudiendo observar que entre mayor sea el valor en este parámetro, la textura resultante presentará una estructura más densa, en comparación con el resultado de un valor menor en el mismo parámetro.

En el caso del resultado de la textura final generada con apariencia amarillo-pardo, se pueden realizar diferentes texturas manipulando este valor, con lo cual podemos obtener diferentes resultados con pequeñas variaciones sobre la textura generada, dando lugar a tener aspectos más cargados de un tono amarillo, que podría utilizarse para contar con diferentes grados de tejido cauterizado sobre el simulador.

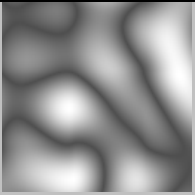
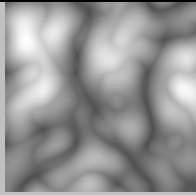
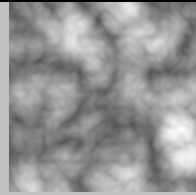
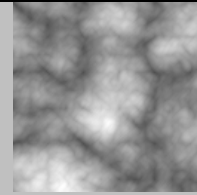
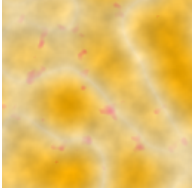
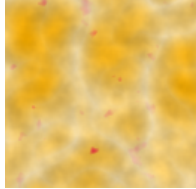
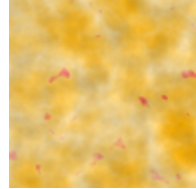
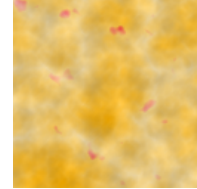
Octavas	2.38	3.43	5.63	10.54
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (apariciencia amarillo-pardo)</i>				

Tabla 15. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural turbulencia en la generación de la textura con apariencia amarillo-pardo.

4.2.2.2 Función movimiento Browniano fraccional (fBm1)

La segunda función que se utiliza para la generación de la textura de apariencia amarillo-pardo, es la conocida como movimiento Browniano fraccional (*fBm*), con esta función lograremos agregar en la textura pequeñas manchas en color rojo. Siendo esta también la que determine primordialmente el color amarillo sobre la textura final generada.

Veamos cuales son los parámetros de esta función:

Funcion fBm1 (escala, h, lacunaridad, octavas)

Ahora se explicaran los diferentes efectos que se logran con la modificación de cada uno los parámetros de esta función.

Parámetro ESCALA

Nuevamente contamos con el parámetro de la escala dentro de esta función, recordando que es el que nos da el efecto “zoom” dentro de la textura.

Observando los resultados en la tabla 16, se puede observar que aparte del color amarillo que proporciona a la textura final generada, agrega a está, pequeñas manchas en color rojo; cabe mencionar que los puntos en color negro resultado de la función procedural básica, son lo que al convertirse en el modelo de color RGB, estos cambian a un color rojo.

Por lo cual en el valor de la *escala= 4*, contamos con muchos pequeños puntos rojos, mientras que en el valor *escala=256*, se cuenta con un mancha grande. Para nuestro caso se opto por contar con un valor intermedio, para lograr algunas manchas sobre nuestra textura final generada (amarillo-pardo).

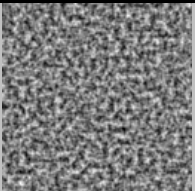
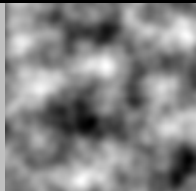

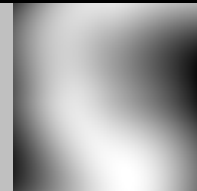
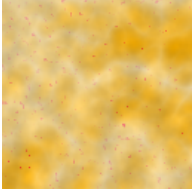
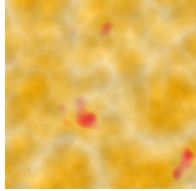
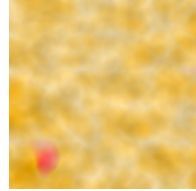
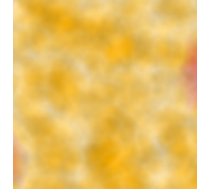
Escala	4	32	64	256
<i>Textura función fBm1</i>				
<i>Textura de tejido final generada (apariencia amarillo-pardo)</i>				

Tabla 16. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.

Parámetro H

Este parámetro quita o agrega las tonalidades de las manchas rojas que aparecen en la textura final generada, es decir, siendo H el parámetro responsable del grado de rugosidad en la textura de la función fBm, agrega una apariencia de rugosidad cuando el valor es pequeño, mientras que al contener un valor mayor, tendrá una apariencia con menos grado de rugosidad, finalmente al realizar la conversión del modelo de color RGB, el grado de rugosidad dentro de la textura determinaran las zonas de las tonalidades de manchas rojas en la textura sintética.

En la tabla 17 se pueden observar los resultados en la variación de los valores del parámetro H. En nuestro caso se eligió un valor intermedio para contar con dichas tonalidades rojas.

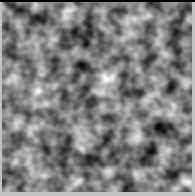

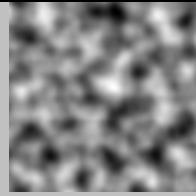
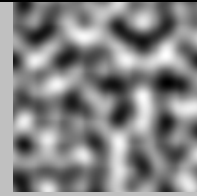
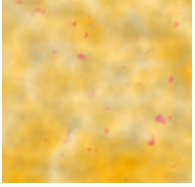
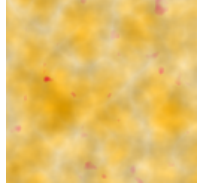
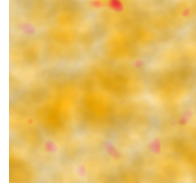
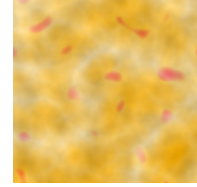
H	0	0.4	2.2	10
Textura función fBm1				
Textura de tejido final generada (apariencia amarillo-pardo)				

Tabla 17. Resultados de la modificación del parámetro H, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.

Parámetro LACUNARIDAD

Este parámetro al igual que el anterior participa igualmente en la definición de manchas rojas sobre el tejido final deseado dentro de las tonalidades amarillas.

Sin embargo analizando los resultados obtenidos en la tabla 18, se pudo concluir que, con la modificación del valor de este parámetro, se agregaran las manchas rojas sobre la textura cuando este parámetro tenga un valor pequeño. Por el contrario si no queremos que nuestra textura final en tonalidades amarillas cuente con pocas o ninguna mancha en rojo, solo será necesario colocar un valor grande en el valor del mismo.

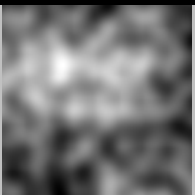
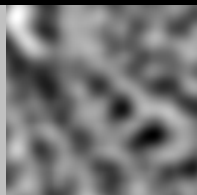
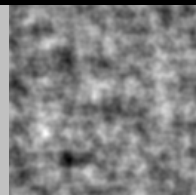
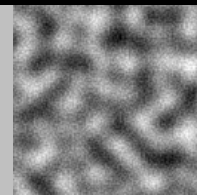
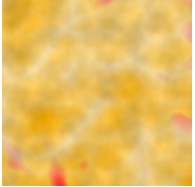
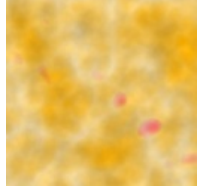
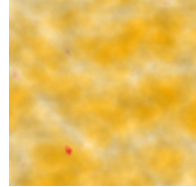
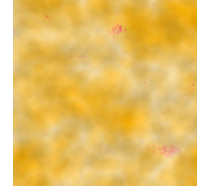
Lacunaridad	0.2	0.8	2.2	10
Textura función fBm1				
Textura de tejido final generada (apariencia amarillo-pardo)				

Tabla 18. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.

Parámetro Número de OCTAVAS

Llegamos al último parámetro de esta función procedural básica, que como se ha mencionado en otras funciones procedurales, este parámetro es el encargado de agregar un número de detalles mayor a la textura sintética cuando se le asigna un valor “grande”.

Los resultados que se obtienen al manipular los valores de entrada de este parámetro no parecen alterar significativamente el resultado final de la textura de apariencia amarillo-pardo. Se pueden visualizar los resultados en la tabla 19.


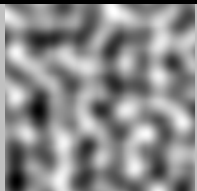
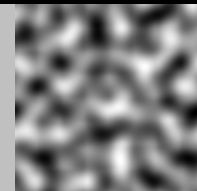

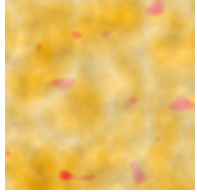
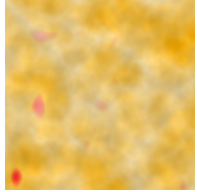
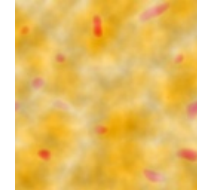
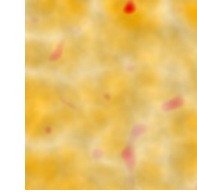
Octavas	1.5	2.84	5.63	10.82
<i>Textura función fBm1</i>				
<i>Textura de tejido final generada (apariciencia amarillo-pardo)</i>				

Tabla 19. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural fBm1 en la generación de la textura con apariencia amarillo-pardo.

4.2.2.3 Función movimiento Browniano fraccional (fBm2)

La última función que se utilizó en la generación de esta textura, es nuevamente la función procedural *fBm*, a diferencia de la función *fBm* anterior, los parámetros que se utilizan en esta función tienen un valor diferente obteniendo con ellos resultados totalmente diferentes a pesar de tratarse de la misma función procedural básica. El resultado de esta función se debe colocar en el canal de valor (V), del modelo HSV para posteriormente realizar la conversión al modelo RGB. Por lo cual podemos deducir que esta función participa en agregar los detalles de los efectos de color blanco dentro de esta textura final generada.

Así los parámetros de esta función son:

Funcion fBm2 (escala, h, lacunaridad, octavas)

Se procederá a la explicación de los resultados que se obtienen con la modificación de los valores de cada uno de sus parámetros de entrada. Como antes se ha mencionado, los valores de los demás parámetros quedaran definidos con los mismos valores que se utilizaron en la generación de la textura de apariencia amarillo-pardo, vistos en el capítulo anterior.

Parámetro ESCALA

Este parámetro nos proporciona dentro de la textura una característica muy vistosa, realizando un efecto de “zoom” al incrementar el valor de este parámetro.

Los resultados que se obtuvieron al variar dicho parámetro los podemos visualizar en la tabla 20. Cabe mencionar que esta función agrega a la textura final generada de apariencia amarillo, el tono del color blanco. Logrando con ello una textura con tonalidades muy parecidas a las encontradas en la imagen de tejido real.

En los resultados también podemos observar el efecto “zoom” que sufren las texturas al ir incrementando el valor de este parámetro.



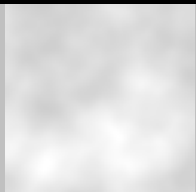
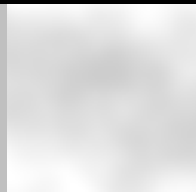
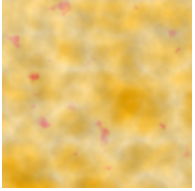
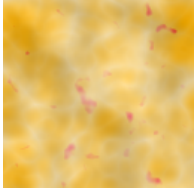
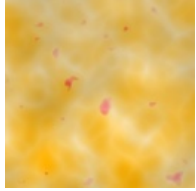
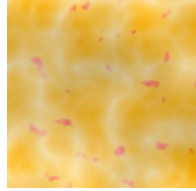
Escala	16	32	64	128
<i>Textura función fBm2</i>				
<i>Textura de tejido final generada (apariencia amarillo-pardo)</i>				

Tabla 20. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.

Parámetro H

Parámetro H, define el grado de rugosidad en la textura simulada. Cuando el valor en este parámetro es “pequeño”, las texturas alcanzaran un grado *mayor de rugosidad*. Mientras que para los valores “grandes”, alcanzara una textura con una apariencia con menos grado de rugosidad.

En los resultados obtenidos de la tabla 21, podemos visualizar el efecto que agrega la modificación de este parámetro a la textura final generada. Logrando agregar más efectos de color blanco cuando se cuenta con un valor más pequeño en dicho parámetro, es decir un valor entre el rango $0 \leq H \leq 0.9$. Mientras que para un valor mayor en este parámetro arrojará un valor con menos tonalidades en blanco.


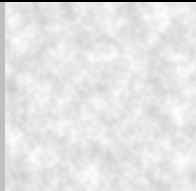
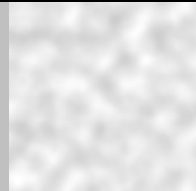
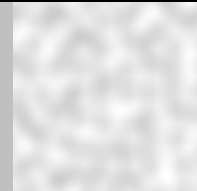
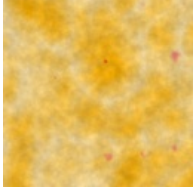
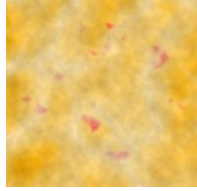
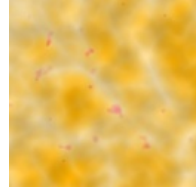
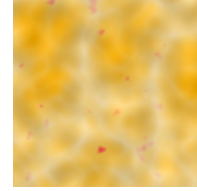
H	0	0.4	2.2	10
<i>Textura función fBm2</i>				
<i>Textura de tejido final generada (apariencia amarillo-pardo)</i>				

Tabla 21. Resultados de la modificación del parámetro H, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.

Parámetro LACUNARIDAD

Este parámetro como se ha mencionado anteriormente, nos permite agregar a la textura una serie de lagunas o huecos. Si el valor del parámetro es “pequeño”, habrá una cantidad menor de huecos comparado con un valor mayor en el cual habrá una cantidad mayor de huecos o “espacios vacíos” sobre la textura final generada.

Los resultados que se obtuvieron al modificar este parámetro se puede apreciar en la tabla 22. Obteniendo la apariencia de una textura con más espacios vacíos al incrementar el valor de esta parámetro. Bien se puede ir variando los valores de este parámetro, sobre todo para conocer la visualización final de la textura vascular generada con apariencia amarillo-pardo.

En nuestro caso podemos generar varios resultados con la manipulación de este valor ya que podremos contar diferentes tonalidades amarillas, con más o menos manchas en rojo, así como más o menos zonas en tonos blancos. Siendo este parámetro ideal para realizar varias combinaciones de esta textura sintética.


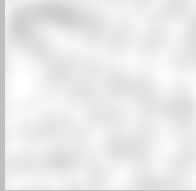


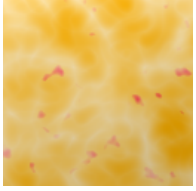
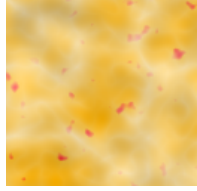
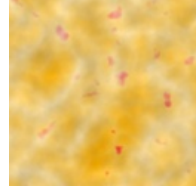
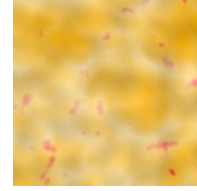
Lacunaridad	0.2	0.8	2.2	10
<i>Textura función fBm2</i>				
<i>Textura de tejido final generada (apariencia amarillo-pardo)</i>				

Tabla 22. Resultados de la modificación del parámetro LACUNARIDAD, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.

Parámetro Número de OCTAVAS

El último parámetro de esta función nos lo da el *número de octavas*, el cual es el responsable del número de iteraciones que hay dentro del algoritmo generador, en el cual con un valor mayor en este parámetro, se podrán obtener más pequeños detalles dentro de la textura.

Los resultados obtenidos en la manipulación del valor de este parámetro, lo podemos apreciar en la tabla 23. Los resultados sobre la textura final generada, se basa nuevamente en la cantidad de manchas rojas presentes en la textura. También se puede utilizar este parámetro para la obtención de diferentes texturas dependiendo de las tonalidades y efectos que se desean obtener.





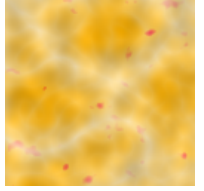
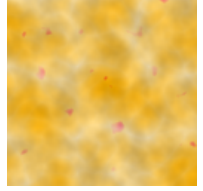
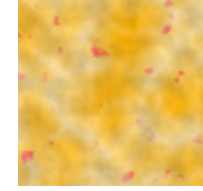
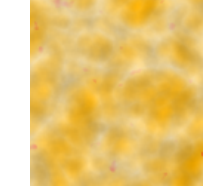
Octavas	1.5	2.84	5.63	10.82
<i>Textura función fBm2</i>				
<i>Textura de tejido final generada (apariencia amarillo-pardo)</i>				

Tabla 23. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural fBm2 en la generación de la textura con apariencia amarillo-pardo.

4.2.3 Tejido fibroso de la cápsula (apariencia rugosa)

Para la generación de esta textura, solo se hizo uso de la función básica *turbulencia modificada*, cabe recordar que la función turbulencia modificada fue obtenida a partir de la función *turbulencia*, la diferencia radica en la multiplicación de un valor constante en los valores de sus coordenada en “y”.

A fin de conocer las variaciones que causan las modificaciones de los parámetros de entrada de su función generadora, se sintetizaron algunos de los resultados obtenidos, recordando como en los casos anteriores, que los demás parámetros contarán con el valor por omisión que se utilizó en la generación de la textura 2D de apariencia rugosa.

4.2.3.1 Función turbulencia modificada

Esta función cuenta con los mismos parámetros de entrada que la función básica *turbulencia*, ya que es dentro del algoritmo el cambio que se realizó para obtener la función turbulencia modificada.

Los parámetros de esta función son:

Funcion turbulencia (escala, octava_inicial, num_octavas)

A continuación se presentan las tablas con los resultados más significativos de los cambios que sufre la textura final generada al modificar cada uno de los valores de los parámetros de esta función básica. Igualmente solo se utilizaron los resultados obtenidos con el color de textura *rosa con negro*, ya que, en los demás casos se obtienen los mismos resultados.

Parámetro ESCALA

Este parámetro nos es útil en la generación de la textura a varias escalas, logrando obtener el mismo comportamiento en la textura final con apariencia rugosa, ya que en esta generación de textura sólo hay un cambio de color sin que este altere las formas base que se obtiene de la función procedural básica.

En la tabla 24 se pueden visualizar los resultados antes mencionados, variando la apariencia de la escala al incrementar el valor del parámetro, sin embargo la textura final obtenida no sufre cambios en la forma, solo sufre cambios en tonalidad del color.

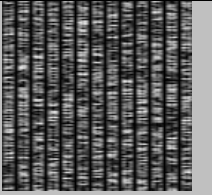
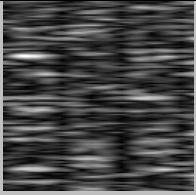
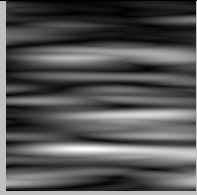
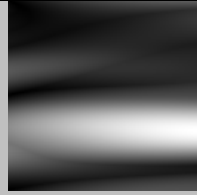
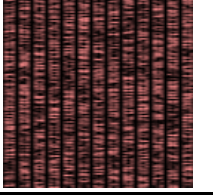
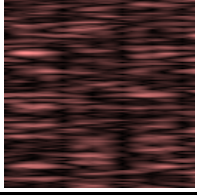
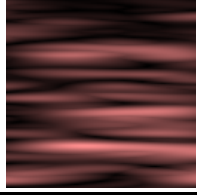
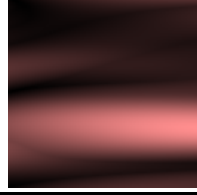
Escala	4	16	64	256
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (apariencia rugosa)</i>				

Tabla 24. Resultados de la modificación del parámetro ESCALA, dentro de la función procedural turbulencia modificada en la generación de la textura con apariencia rugosa.

Parámetro OCTAVA INICIAL

El siguiente parámetro de esta función procedural, es el conocido como *octava inicial*. Este parámetro está muy relacionado con el parámetro *números de octavas*, ya que los dos marcan las iteraciones y formas que se generaran en la textura simulada.

A diferencia de los resultados obtenidos con este parámetro en la generación de las dos texturas anteriores, en este caso no hay un cambio tan significativo al manipular el valor del parámetro con respecto al valor del parámetro número de octavas. El único efecto que se logra es obtener más o menos pliegues dentro de la textura, conservando sus características de ondulaciones.

En la tabla 25, podemos visualizar algunos de los resultados más significativos que se obtuvieron al variar el valor de este parámetro. Es importante recordar que el valor del parámetro debe ser un número entero positivo menor al valor del parámetro número de octavas.


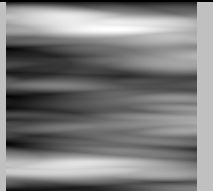
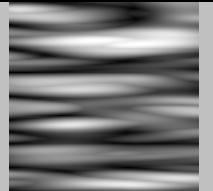
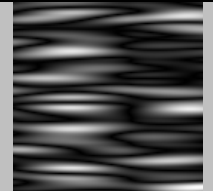
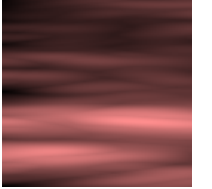
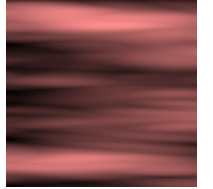
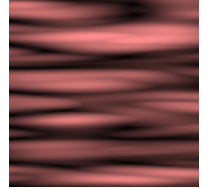
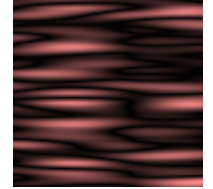
Octava inicial	1	2	3	4
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (apariencia rugosa)</i>				

Tabla 25. Resultados de la modificación del parámetro OCTAVA INICIAL, dentro de la función procedural turbulencia modificada en la generación de la textura con apariencia rugosa.

Parámetro Número de OCTAVAS

Llegamos al último parámetro de esta función procedural básica, que como se ha mencionado en otras funciones procedurales, este parámetro es el encargado de agregar un número de detalles mayor a la textura sintética cuando a este se le asigna un valor “grande”.

Los resultados que se obtienen al manipular los valores de entrada de este parámetro se pueden observar en la tabla 26. Nuevamente observamos que las características de forma de la textura procedural se conservan cuando se realiza la conversión del modelo en RGB, agregando a esta solo la tonalidad del color. Por otro lado también se siguen conservando los pliegues de la textura final.

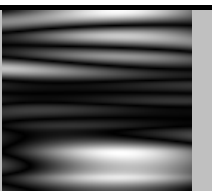
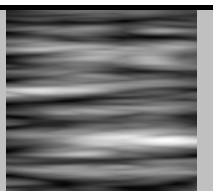
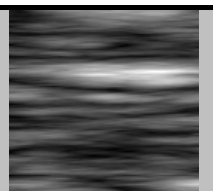
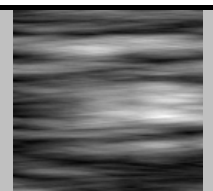
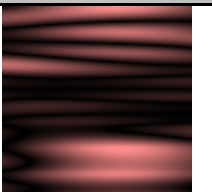
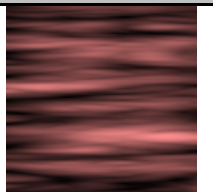
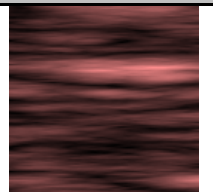
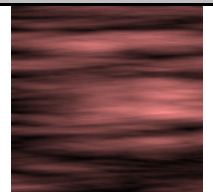



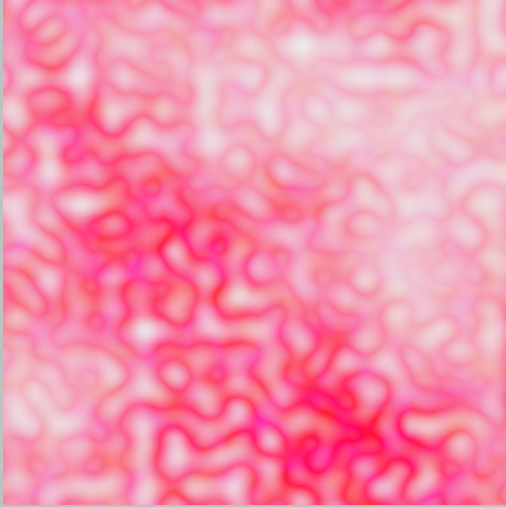
Octavas	3.75	5.75	8.43	12.55
<i>Textura función turbulencia</i>				
<i>Textura de tejido final generada (apariencia rugosa)</i>				


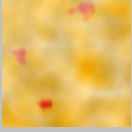
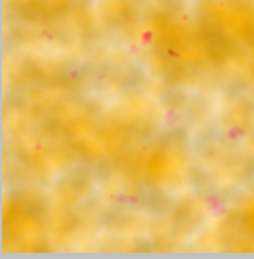
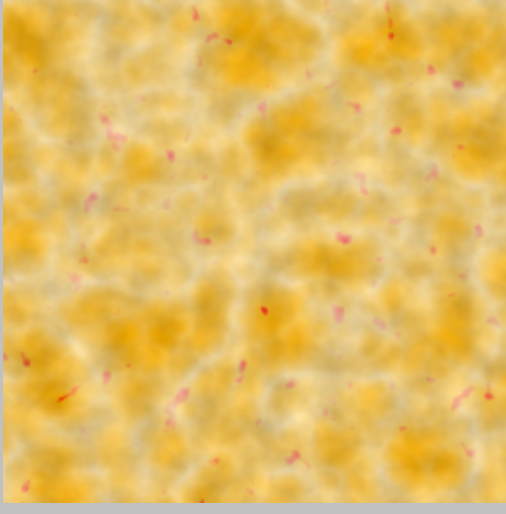
Tabla 26. Resultados de la modificación del parámetro número de OCTAVAS, dentro de la función procedural turbulencia modificada en la generación de la textura con apariencia rugosa.

4.3 Modelos procedurales de las texturas generadas

En este apartado se presenta una de las ventajas más significativas de las funciones procedurales para la generación de texturas. Es decir, la ventaja de poder generar una textura a diferentes tamaños y magnitudes, la cual no importando sus dimensiones, no perderá sus cualidades visuales, conservando los detalles y sus efectos logrados con la función procedural.

En la tabla 27 se muestran las diferentes texturas procedurales descritas a lo largo de este trabajo. Pudiendo observar dicha característica con la obtención de texturas en diferentes tamaños.

	32 x 32 píxeles	64 x 64 píxeles	128 x 128 píxeles	256 x 256 píxeles
Textura tejido (vascularidades)				

Textura tejido (apariciencia amarilla)				
--	---	---	---	--

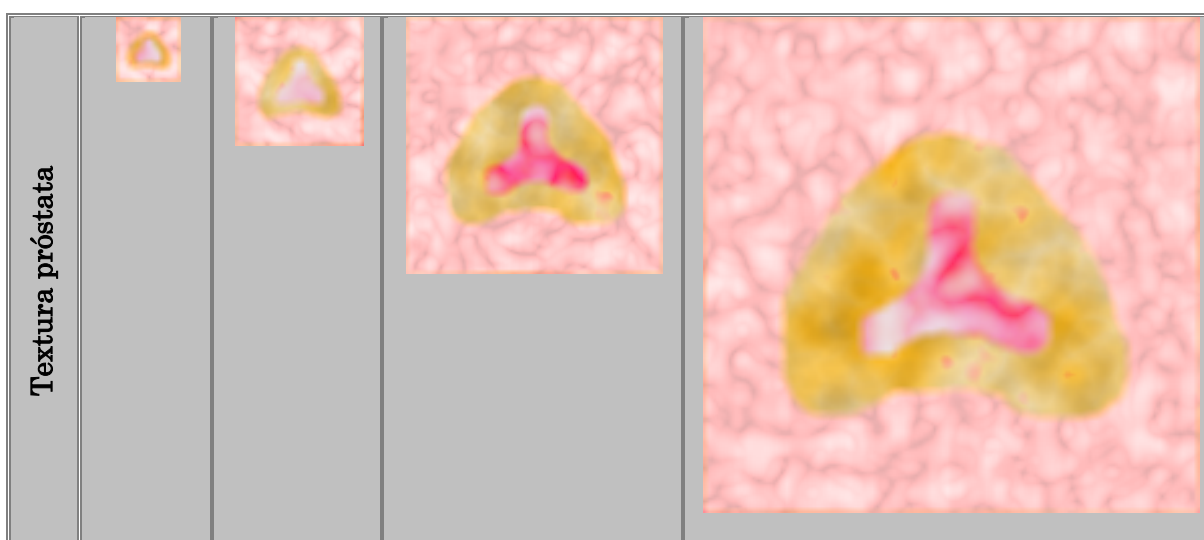
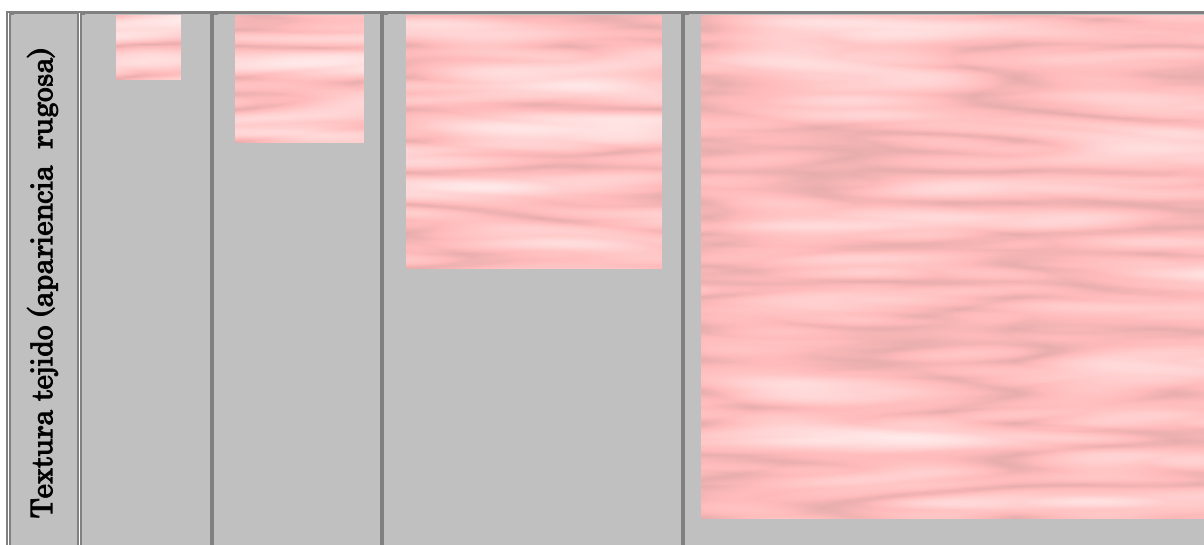


Tabla 27. Visualización de resultados, en donde se puede apreciar las diferentes texturas sintéticas generadas a la largo de este trabajo de tesis. En donde se aprecia una de las ventajas de los métodos procedurales, es decir, la generación de texturas a diferentes tamaños, las cuales no pierden sus características visuales no importando su tamaño o dimensión.

4.4 Comparación de la textura generada con la muestra de tejido real

Se procedió a realizar una comparación entre las texturas generadas de cada tejido con la muestra de tejido real obtenido de las imágenes médicas reales. Brevemente se muestran los resultados obtenidos, así como las conclusiones a las cuales se llegaron.

4.4.1 Textura tejido de la uretra con vascularidades

En la figura 4.1 se muestran las conversiones de los modelo de color HSV a RGB, de la textura real y la textura sintética de apariencia vascular. Observando que las texturas en los canales del modelo HSV de la textura real son muy similares a las texturas generados por computadora. Dando lugar a una textura sintética muy parecida visualmente al tejido real, como también se puede apreciar en la figura.

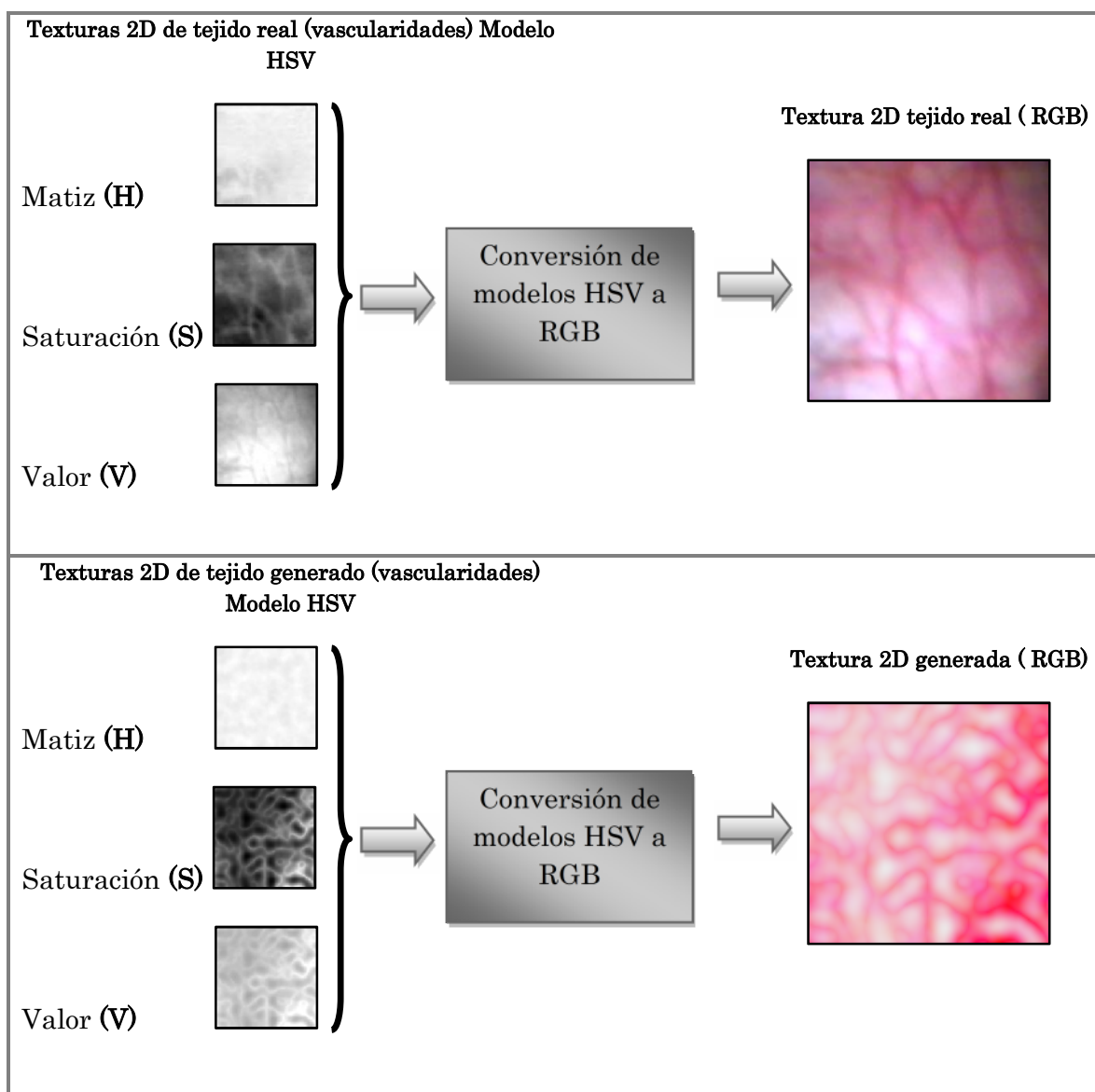


Figura 4.1 Comparación de la conversión del modelo de color HSV a RGB de la textura real y sintética del tejido con vascularidades.

Histogramas del modelo HSV de la imagen de textura real y textura generada del tejido con apariencia de vascularidades.

Como parte de las pruebas, se generaron los histogramas correspondientes a la textura real y textura generada, los cuales podemos apreciar en la figura 4.2. Los histogramas se realizaron bajo el modelo de color HSV, englobando en la gráfica sus tres canales: matiz, saturación y brillo. El rango de valores de los histogramas empieza con el valor 0 que corresponde al color negro, hasta el valor 255 que corresponde al color blanco. El rango entre estos valores pertenece a los valores en escala de grises.

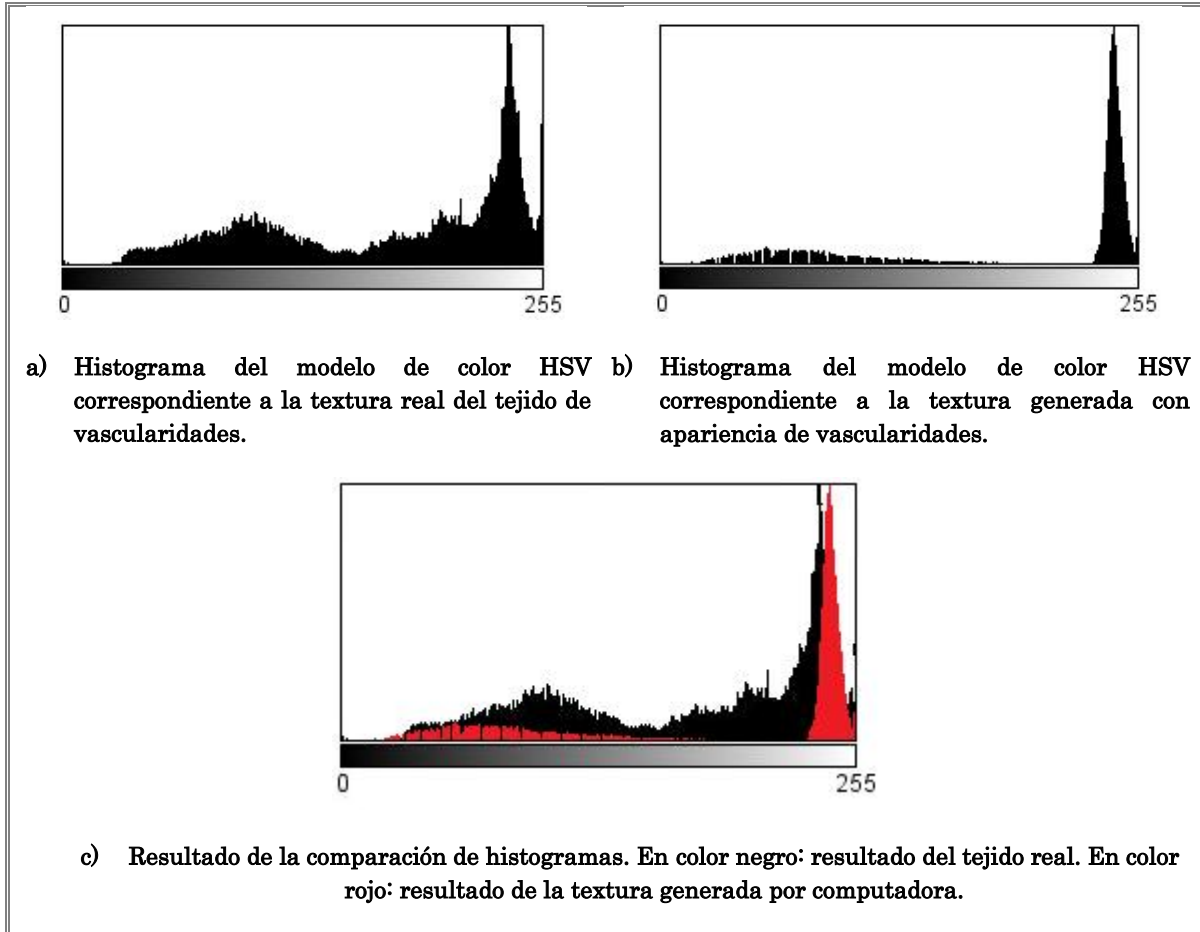


Figura 4.2 Resultados de los histogramas de la textura real y sintética del tejido con apariencia de vascularidades.

Finalmente comparando los resultados, se puede concluir que la textura generada tiende a tomar la misma forma de la gráfica del resultado de la textura de la imagen real. Con lo cual puede considerarse que está dentro del rango de valores de la textura real, dando finalmente un resultado visual muy aceptable para ser usada en el simulador virtual de cirugía.

4.4.2 Textura del tejido interno de la próstata (amarillo-pardo)

Los resultados de la conversión de modelos de la textura real y la textura sintética del tejido interno de la próstata (amarillo-pardo), los podemos visualizar en la figura 4.3. La conversión se realizó del modelo de color HSV a RGB. Observando que las texturas en los canales del modelo HSV de la textura real son muy similares a las texturas generadas por computadora.

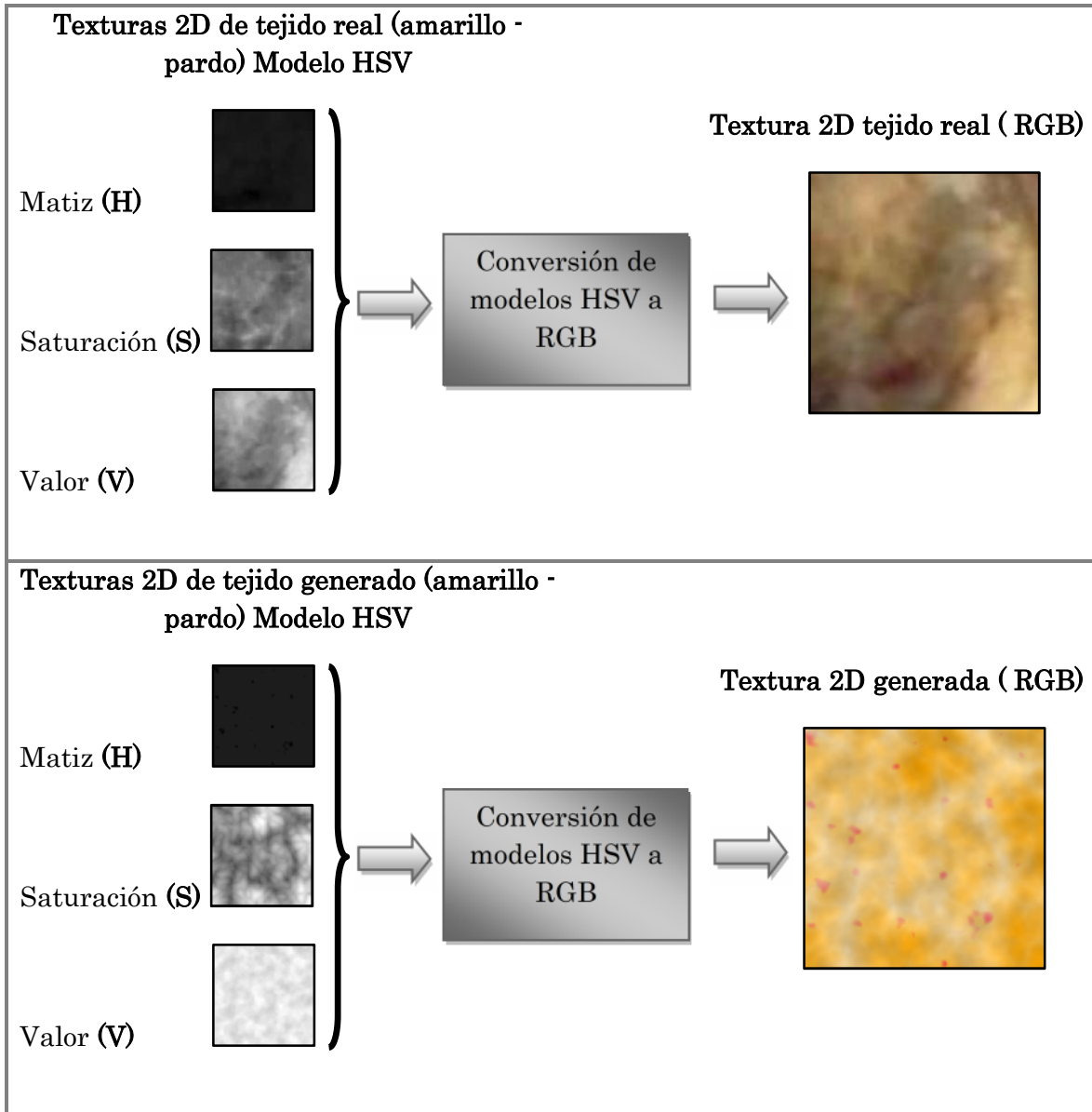


Figura 4.3 Comparación de la conversión del modelo de color HSV a RGB de la textura con apariencia amarillo-pardo.

Histogramas del modelo HSV de la imagen de textura real y textura generada del tejido (amarillo-pardo).

Como en el caso anterior, nuevamente podemos observar los resultados de los histogramas en la figura 4.4, correspondientes a la textura real y la textura generada del tejido con apariencia amarillo-pardo, teniendo una gráfica muy similar en el resultado de la textura generada.

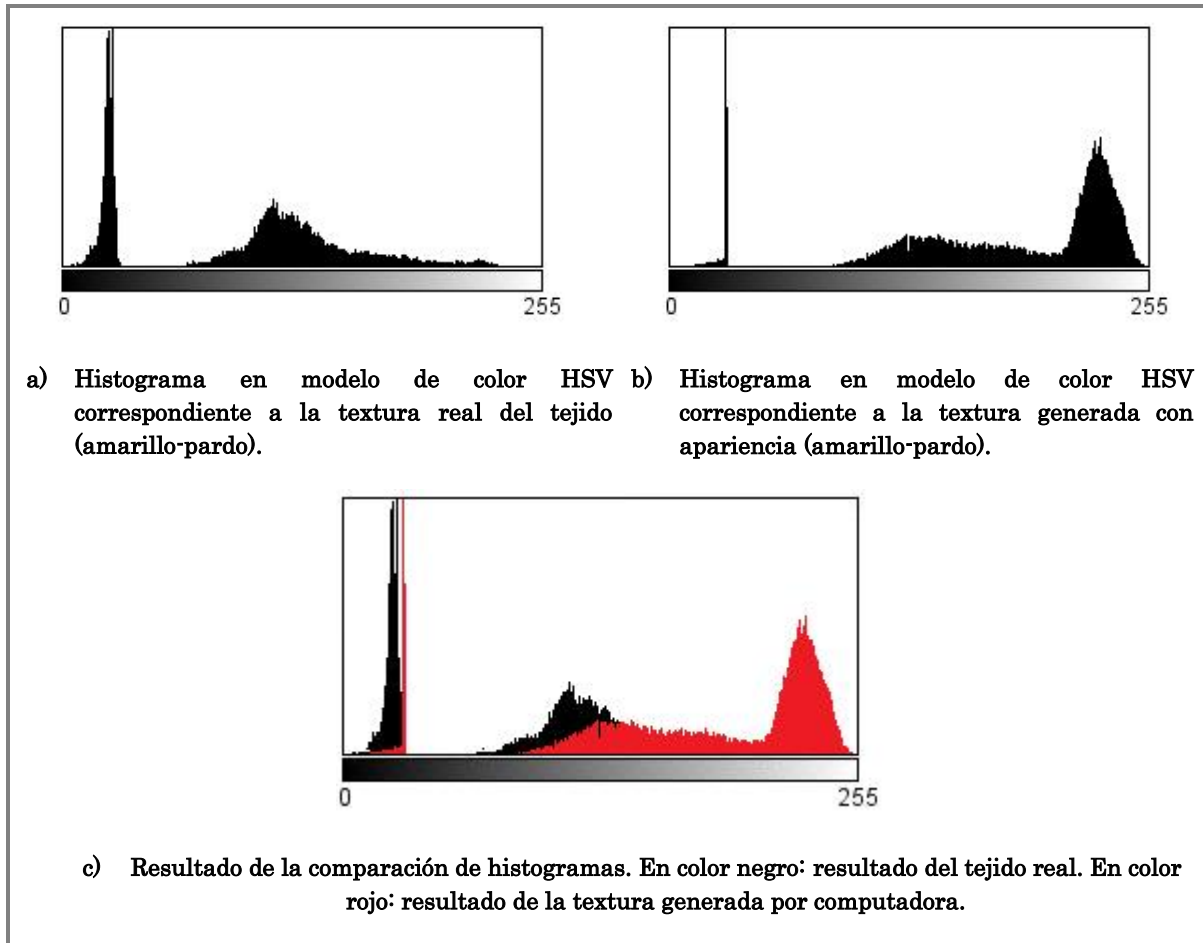


Figura 4.4 Resultados de los histogramas de la textura real y sintética del tejido con apariencia amarillo-pardo.

Al revisar y comparar los resultados anteriores, se puede comentar, que el resultado de la textura generada de este tejido, tiende a ser un poco más encendido en comparación de la imagen real. De hecho en el histograma de la textura generada, se puede apreciar un pico que se encuentra muy cercano a las tonalidades cercanas del valor 255 (correspondiente al color blanco), lo que nos da como resultado una textura con mayor grado de saturación. Sin embargo para nuestros intereses y objetivos, se decidió trabajar con dicha textura sintética bajo esas tonalidades, ya que se obtienen resultados aceptables al ser aplicada dentro del simulador virtual de cirugía.

4.4.3 Textura del tejido de la cápsula de la próstata (apariencia rugosa)

Finalmente los resultados de la conversión de modelos HSV a RGB para la imagen de textura real y la textura generada, correspondiente a la textura del tejido de la cápsula (apariencia rugosa), se pueden apreciar en la figura 4.5.

Se puede ver que hay mucha similitud entre las texturas de los canales: matiz, saturación y brillo entre ambas texturas antes de la conversión.

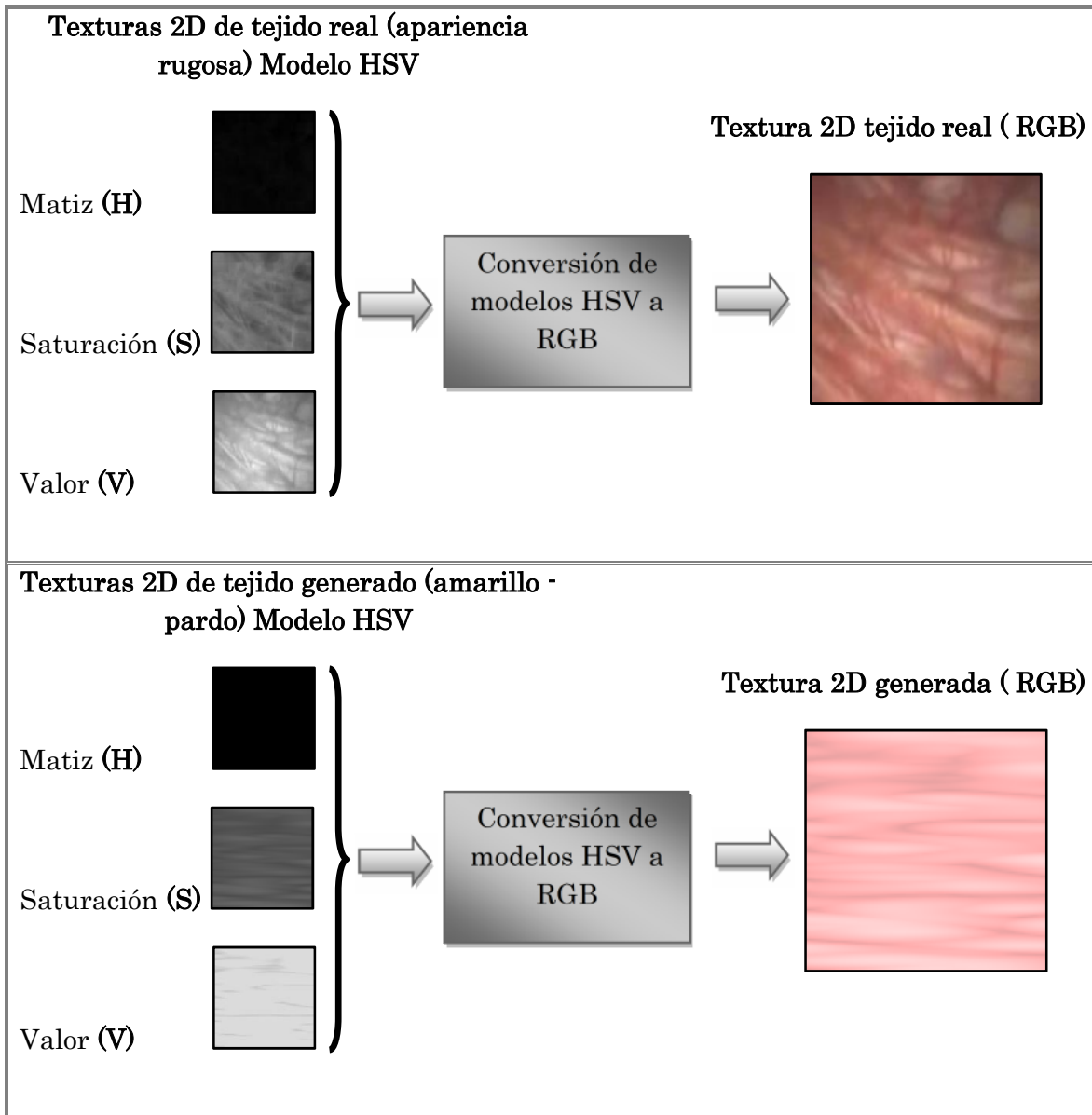


Figura 4.5 Comparación de la conversión del modelo de color HSV a RGB de la textura con apariencia rugosa.

Histogramas del modelo HSV de la imagen de textura real y textura generada del tejido con apariencia rugosa.

En la figura 4.6 podemos ver los resultados de los histogramas de la textura real en comparación con la textura sintética. A diferencia de las texturas anteriores estos histogramas difieren mucho entre sí. A pesar de que las texturas antes de la conversión sean muy similares.

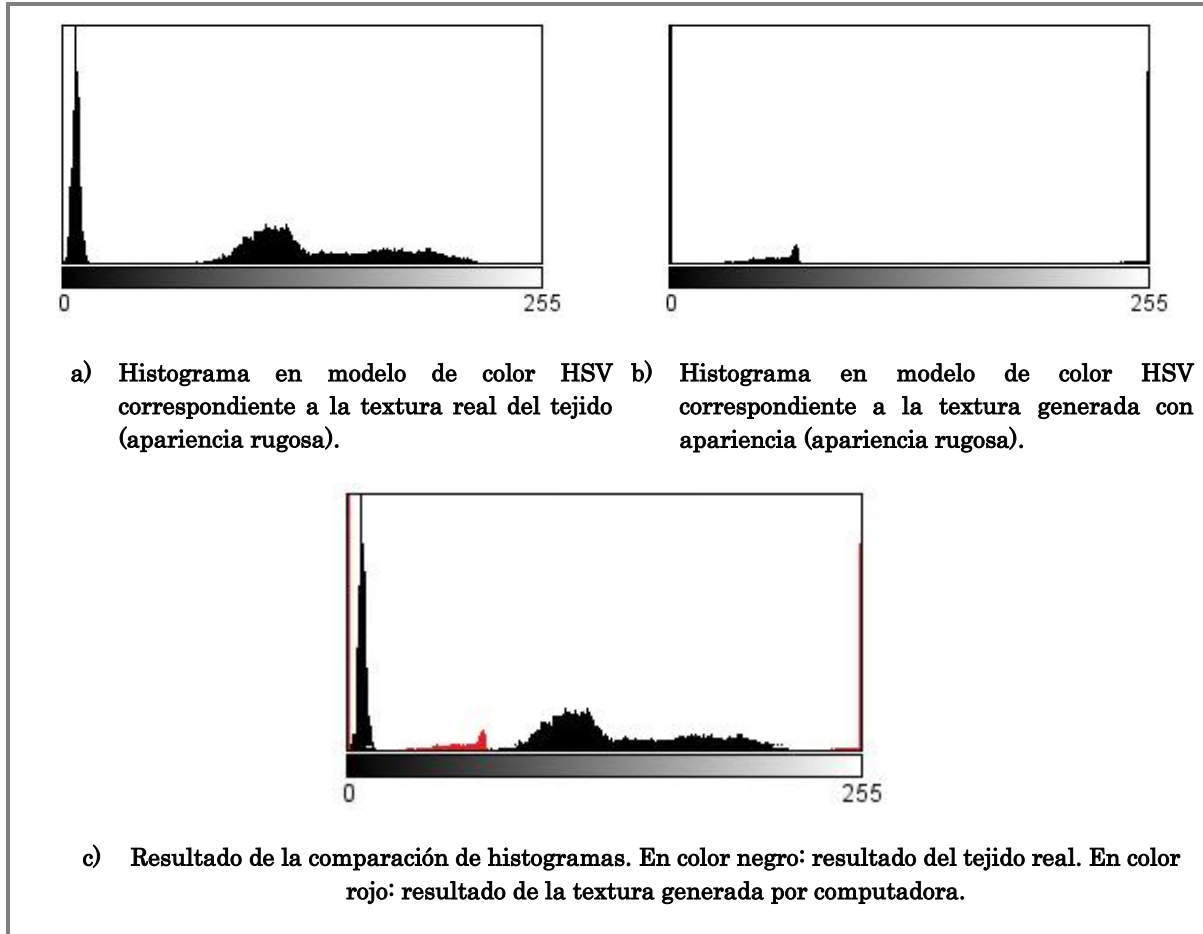


Figura 4.6 Resultados de los histogramas de la textura real y sintética del tejido con apariencia rugosa.

Estas diferencias pueden deberse, a que en la generación de esta textura solo participa un función procedural. Sin tener más combinaciones de funciones que puedan darle otro tipo de efectos. Sin embargo y a pesar de los resultados que nos muestran los histogramas, se decidió utilizar esta textura bajo estas tonalidades; ya que esta, cumple con los efectos deseados para el simulador. Siendo los pliegues y las rugosidades primordiales en la generación de esta textura para ser aplicada posteriormente.

4.5 Comparación de tiempos en la generación de texturas

Otro parámetro importante a considerar es el tiempo de generación de texturas. Para lo cual se hicieron diferentes pruebas con diferentes equipos, para conocer el tiempo que tomaba generar la síntesis de textura procedural.

El tiempo se midió considerando el tamaño del cubo de textura a generar, desglosado en horas, minutos y segundos. Obteniendo los siguientes resultados.

Equipo 1

Características Equipo: Hewlett-Packard Pavilion dv2500 Notebook PC

Procesador Intel (R) Core (TM)² Duo 1.50 GHz. RAM 2.00 GB, Sistema operativo de 32 bits Windows Vista Service Pack 1. Disco Duro: 160 GB.

Generación Imágenes 3D	Tamaño del cubo a generar (x,y,z)	Tiempo
Próstata Cerrada	16	00 : 00 : 04
Próstata Cerrada	32	00 : 00 : 31
Próstata Cerrada	64	00 : 04 : 04
Próstata Cerrada	96	00 : 14 : 19
Próstata Cerrada	128	00 : 37 : 36

Equipo 2

Características Equipo: Dell. Pentium 4

Procesador Intel (R) Pentium(R) 4 CPU 3.00 GHz 2.99 GHz. RAM 1.00 GB, Sistema operativo Microsoft Windows XP Versión 2002. Service Pack 3. Disco Duro: 149 GB.

Generación Imágenes 3D	Tamaño del cubo a generar (x,y,z)	Tiempo
Próstata Cerrada	16	00 : 00 : 15
Próstata Cerrada	32	00 : 02 : 15
Próstata Cerrada	64	0 : 16 : 27
Próstata Cerrada	96	00 : 38 : 12
Próstata Cerrada	128	01 : 16 : 25

Equipo 3

Características Equipo: Dell. Intel Core

Procesador Intel (R) Core (TM)² CPU 6600 @ 2.40 GHz 2.39 GHz. RAM 1.00 GB, Sistema operativo de 32 bits Windows Vista Ultimate Service Pack 1. Disco Duro: 146 GB.

Generación Imágenes 3D	Tamaño del cubo a generar (x,y,z)	Tiempo
Próstata Cerrada	16	00 : 00 : 07
Próstata Cerrada	32	00 : 01 : 05
Próstata Cerrada	64	0 : 07 : 07
Próstata Cerrada	96	00 : 22 : 14
Próstata Cerrada	128	00 : 55 : 35

Equipo 4

Características Equipo: Sun Microsystems

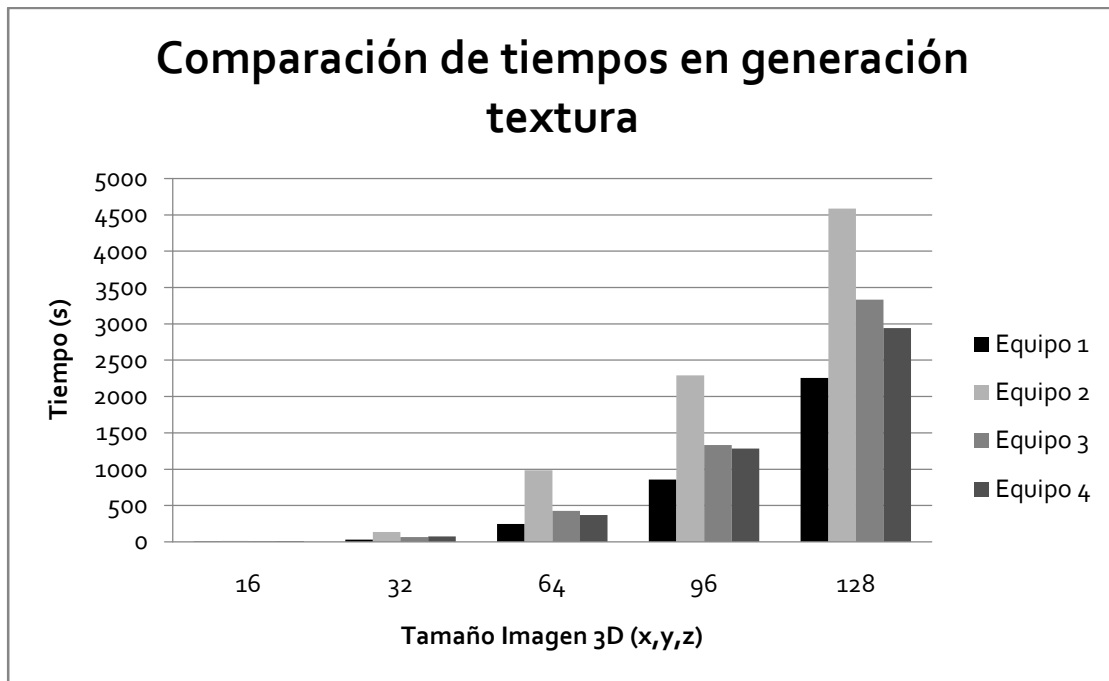
Procesador Dual-Core AMD Opteron™ CPU 1222; 3.00 GHz. RAM 4.00 GB, Sistema operativo Microsoft Windows XP Professional x 64 Edition, Service Pack 2. Disco Duro: 232 GB.

Generación Imágenes 3D	Tamaño del cubo a generar (x,y,z)	Tiempo
Próstata Cerrada	16	00 : 00 : 08
Próstata Cerrada	32	00 : 01 : 13
Próstata Cerrada	64	0 : 06 : 08
Próstata Cerrada	96	00 : 21 : 23
Próstata Cerrada	128	00 : 49 : 04

Dichas pruebas sirven para detectar y mejorar el código fuente en la generación de cada textura de tejido, logrando así que sea más efectiva la generación de la textura volumétrica final.

Por otro se recomienda también utilizar un equipo con propiedades de hardware y software eficientes, para lograr un mejor procesamiento en la generación del volumen de textura y esta pueda generarse en el menor tiempo posible.

A continuación se muestra una gráfica de tiempos en la generación de la textura sólida en 3D. Siendo el equipo 1 en el cual se desarrollo en un tiempo menor.



Gráfica 1. Comparación de tiempos de 4 equipos, en la generación de la textura sólida en 3D de la glándula prostática.

4.6 Interfaz gráfica para la generación de texturas procedurales para el simulador virtual

Con la finalidad de contar una herramienta más eficiente, sobre todo a la hora de manipular y cambiar los parámetros de cada función procedural, se desarrollo una interfaz gráfica para trabajar con la generación de texturas procedurales para el simulador virtual de cirugía.

La interfaz gráfica fue desarrollada con la herramienta wxDev C++ versión 7.0. Dicha interfaz lo que hace es generar una cadena de valores válidos de los parámetros de cada función de la textura a sintetizar, para que esta sea posteriormente ejecutada en la ventana de comandos de Matlab. En la figura 4.7 se observa la imagen de la interfaz gráfica.

Nota: Se debe tener Matlab instalado en el equipo en donde se vaya a utilizar la interfaz gráfica.

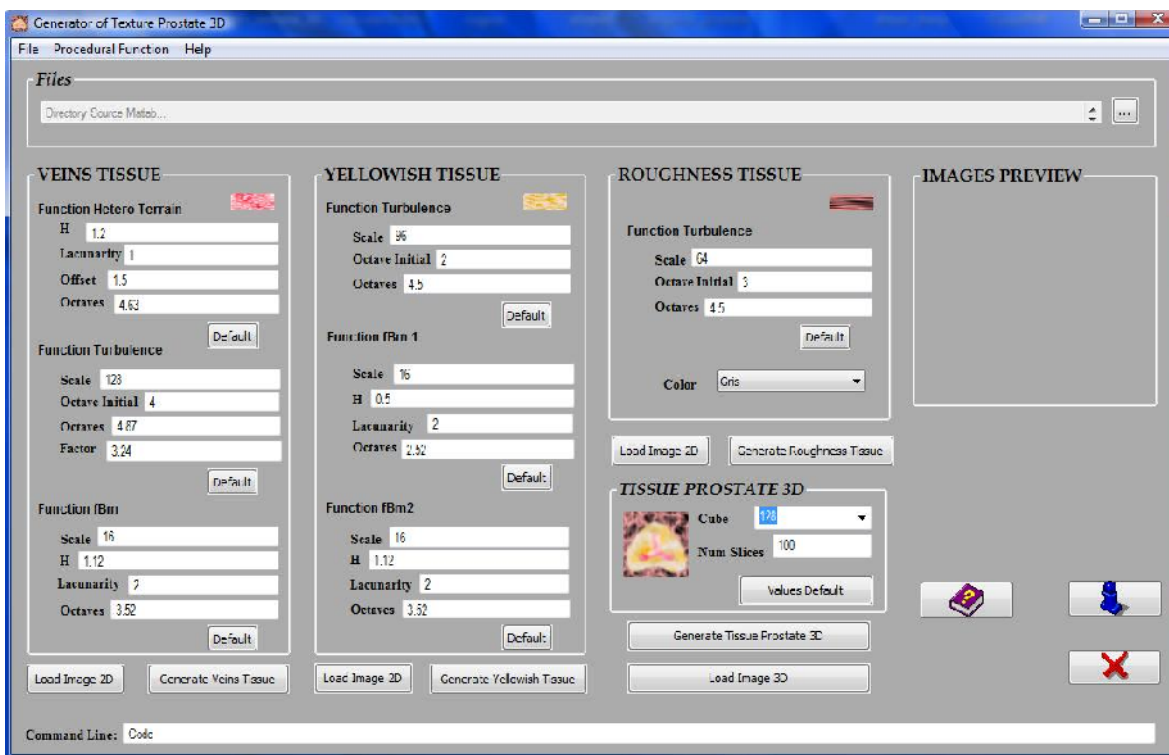
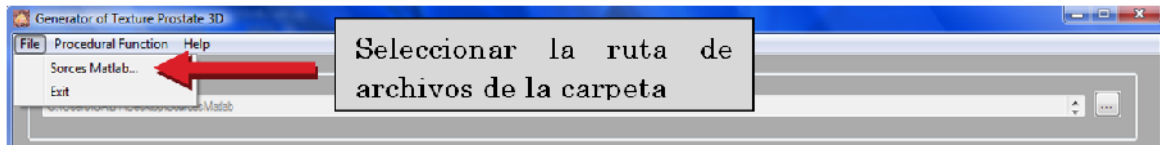


Figura 4.7 Interfaz gráfica que facilita la generación de las texturas procedurales para el simulador virtual de cirugía del procedimiento RTU.

4.6.1 Funcionamiento básico de la interfaz

A fin de entender mejor el funcionamiento y uso de la interfaz gráfica, se presentaran los pasos para la generación de texturas procedurales por medio de la herramienta antes mencionada.

Paso1. Indicar en el programa la ruta de los archivos de código fuente de las texturas procedurales. Ya que el código de la funciones procedurales se seguirá utilizando con Matlab.



Paso 2. Verificar los valores de los parámetros de la textura que nos interesa generar. Se puede modificar el valor de cada parámetro sí es que así se desea, de lo contrario se pueden dejar los valores que el programa trae por omisión.

Paso 3. Dar clic en el botón “GENERATE” de la textura que queremos generar. El programa generara automáticamente la siguiente cadena, la cual abrirá la ventana de comandos de Matlab, en la cual se ejecutará la instrucción de la cadena proporcionada, generando la textura deseada.

Command Line: `matlab -nodesktop -sd C:\Users\GABY\Desktop\Sources\Matlab -r tejido_amarillo*(128,128,96,2,4.5,16,0.5,2,2.52,16,1.12,2,3.52)*`

Paso 4. Debemos esperar a que la instrucción termine de ejecutarse en Matlab. Una vez generada la textura, la ventana emergente de Matlab se cerrara automáticamente.

Los botones y las partes básicas del programa, se pueden visualizar en la figura 4.8. En los cuales tenemos la posibilidad de manipular el valor de los parámetros de cada función o bien colocar los valores que trae por omisión.

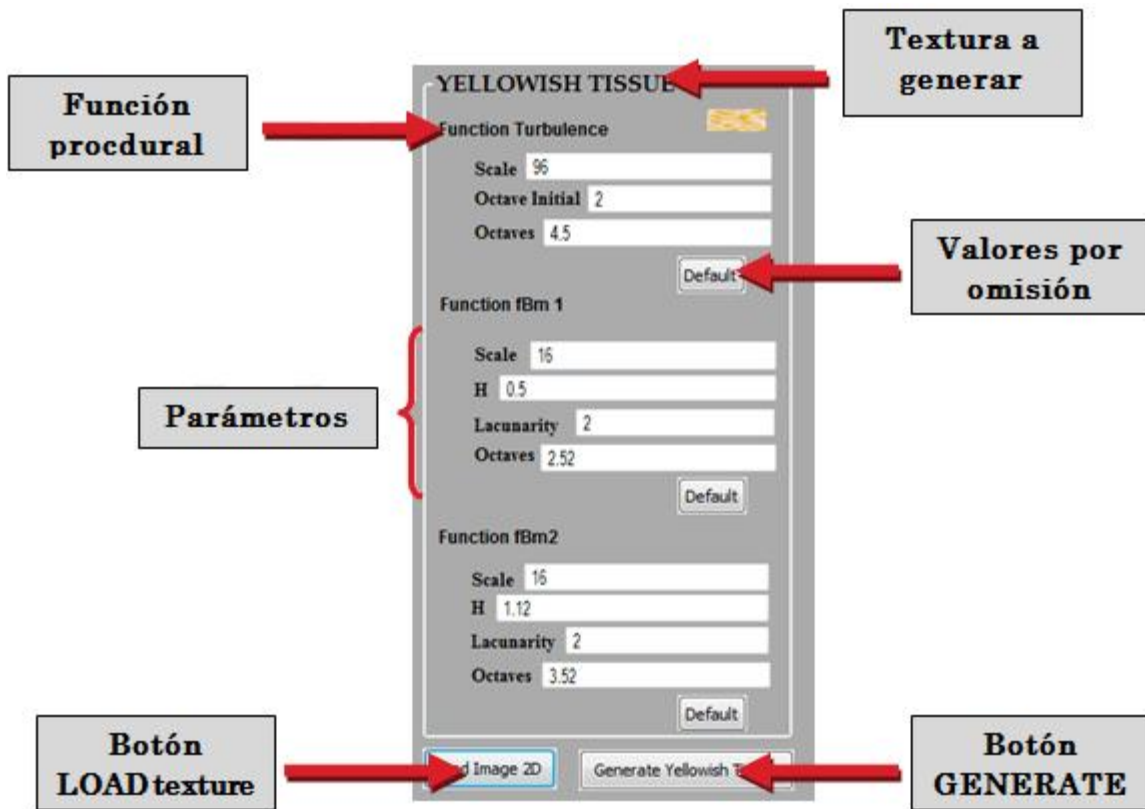
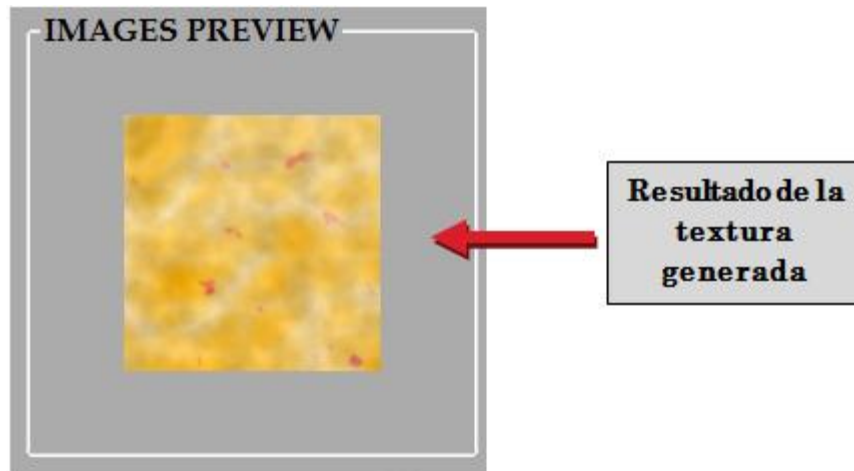


Figura 4.8 Botones y partes básicas de la interfaz gráfica para la generación de texturas procedurales en 2D.

Paso 5. Podemos ahora ver el resultado de la textura generada dando clic en el botón “LOAD”. La textura se visualizara en el recuadro disponible para ello.



Para el caso de la generación de la textura en 3D, se hará de la misma forma que la mencionada anteriormente, la única diferencia radicaré en que debemos indicar al programa los valores de los parámetros: tamaño de la textura a generar y número de imágenes de segmentos de próstata (ver figura 4.9).

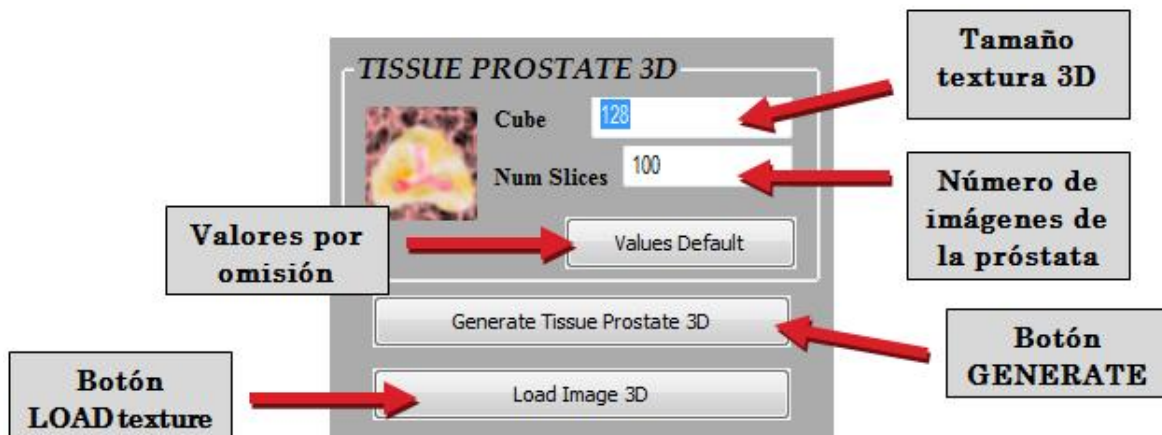


Figura 4.9 Botones y partes básicas de la interfaz gráfica para la generación de texturas procedurales en 3D.

Esta interfaz es muy útil ya que con ella se pueden manipular los valores de los parámetros de las funciones procedurales de una manera más rápida y efectiva, teniendo la posibilidad de realizar pruebas sobre las texturas 2D, antes de generarla en 3D, ahorrando mucho tiempo en la generación de la textura final.

4.7 Pruebas y resultados en el simulador de RTU

Se procedió a realizar las pruebas correspondientes en el simulador virtual de Resección Transuretral de Próstata (RTU), cargando la síntesis de textura procedurales (x,y,z) antes obtenida. En la figura 4.10 se puede visualizar el cambio y transformación del simulador con y sin textura. Apreciando las cualidades que agregan las imágenes de textura, dando realce y una mejor visualización del modelo virtual.

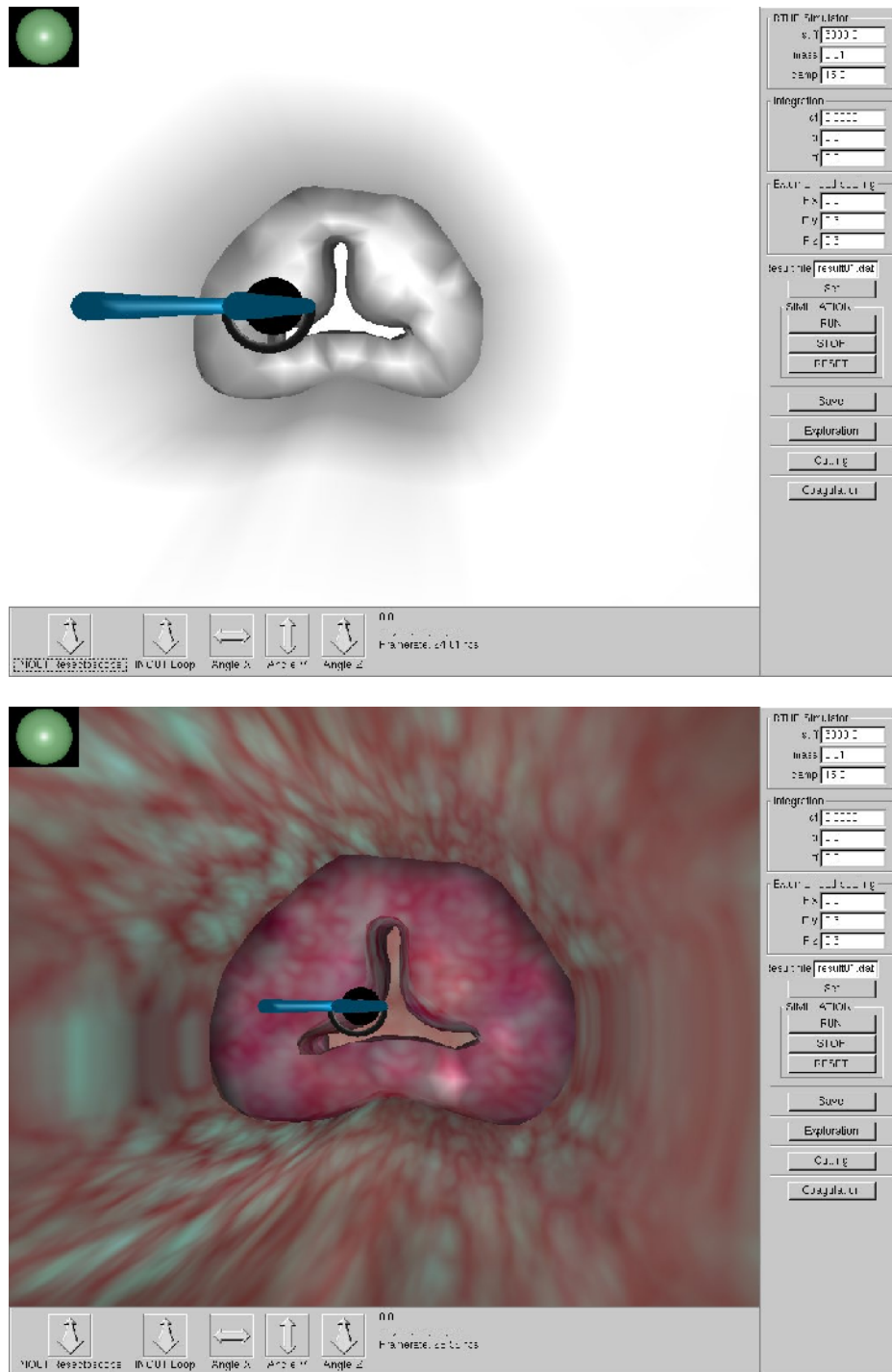


Figura 4.10 Imágenes del simulador virtual RTU, CCADET. Con y sin textura.

En las figuras 4.11, 4.12 y 4.13 podemos visualizar los cambios en cuanto a texturas se refiere, las cuales se han venido modificando y optimizando a lo largo de las mejoras que se han hecho sobre el modelo del simulador virtual.

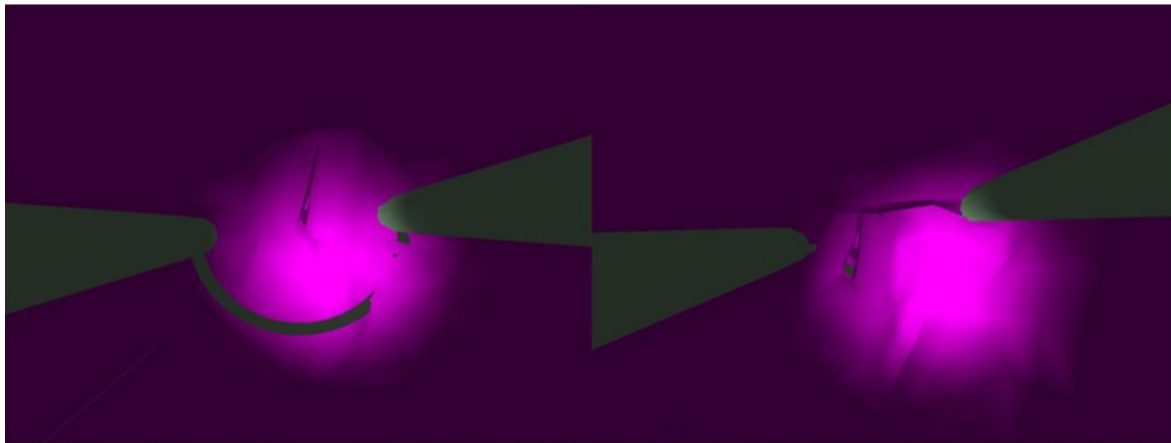


Figura 4.11 Imagen del modelo del simulador RTU , CCADET. Sin textura.

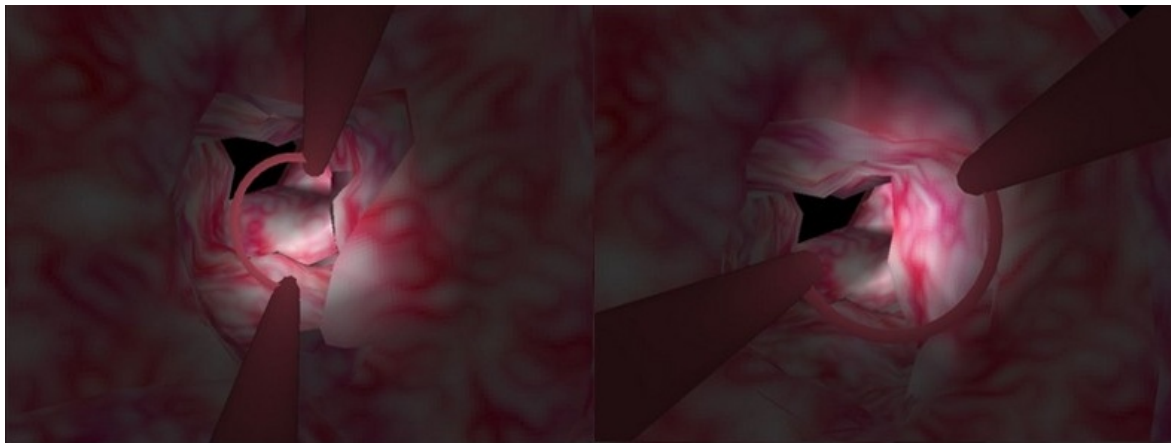


Figura 4.12 Imagen del modelo del simulador RTU, CCADET. Texturizado una sola imagen uniforme.

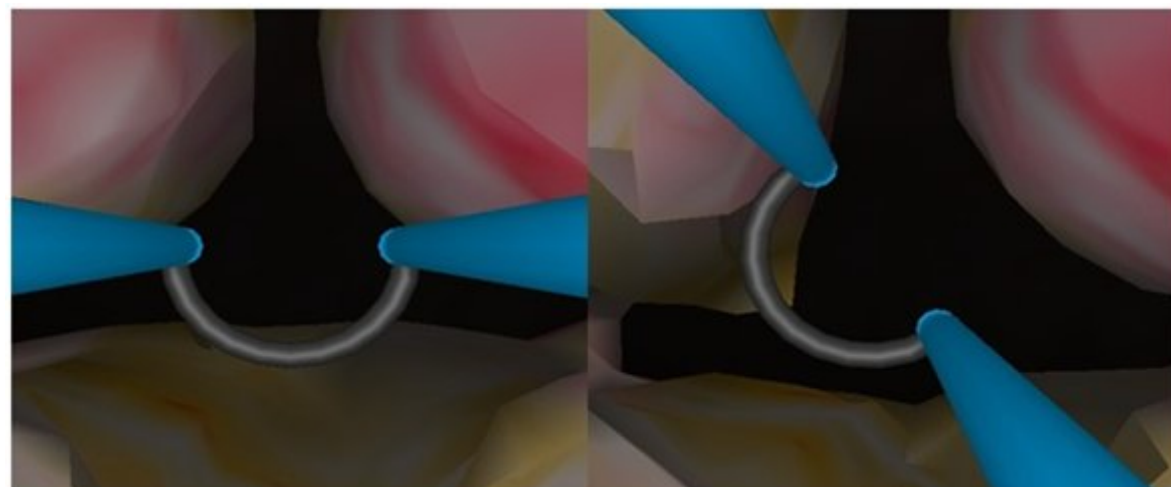


Figura 4.13 Imagen del modelo del simulador RTU, CCADET. Modelo texturizado con la síntesis generada a partir de tres texturas procedurales.

Finalmente se termina este capítulo mostrando una serie de imágenes obtenidas de las pruebas hechas sobre el simulador virtual (ver figura 4.14, 4.15 y 4.16). Mostrando el resultado final al aplicar la síntesis de textura que se genero para tal fin. Así como algunos resultados de la apariencia de la próstata sobre el simulador al realizar algunos cortes sobre el tejido prostático.

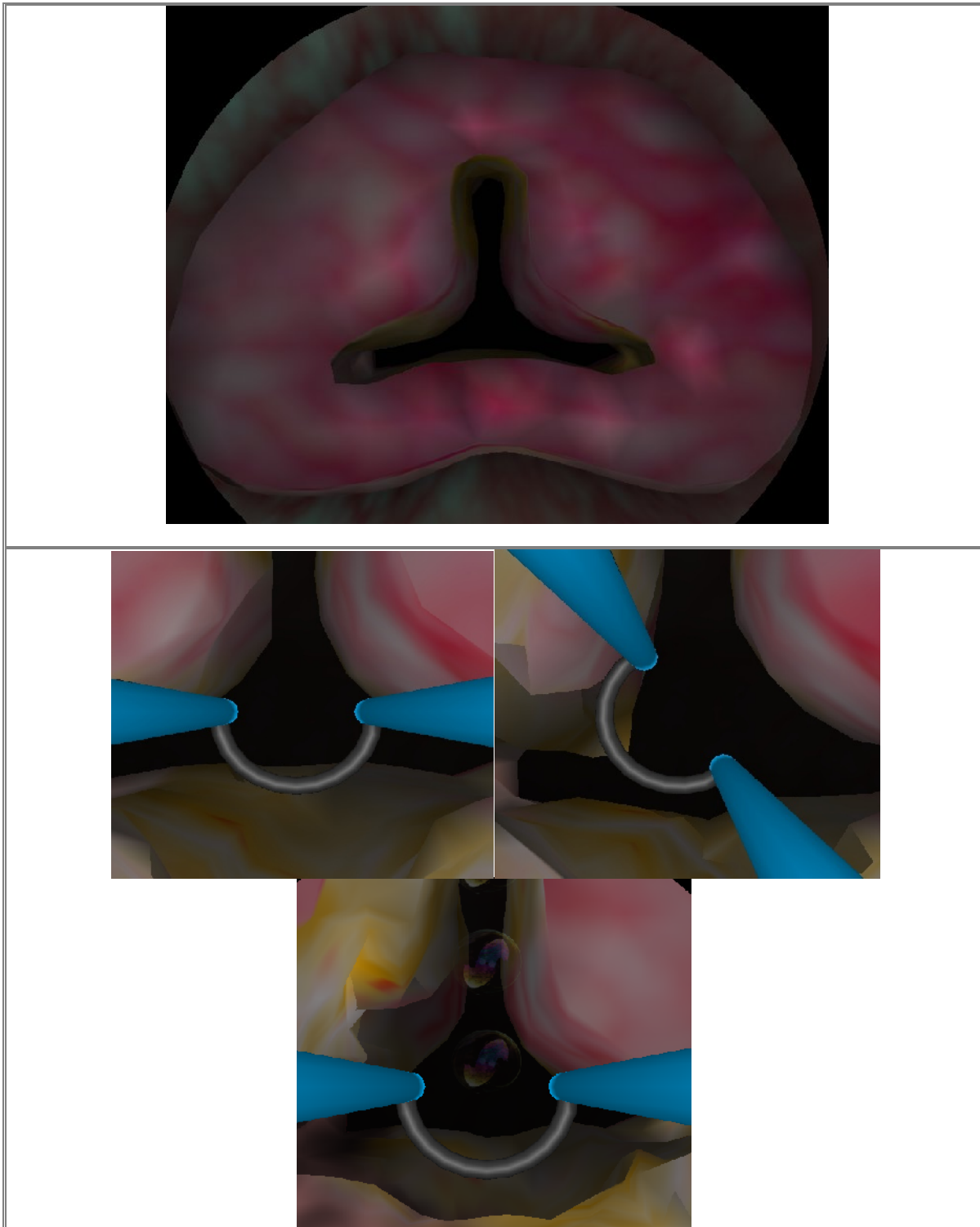


Figura 4.14 Imágenes de diferentes tomas del simulador virtual RTU, texturizado con la síntesis de texturas procedurales.

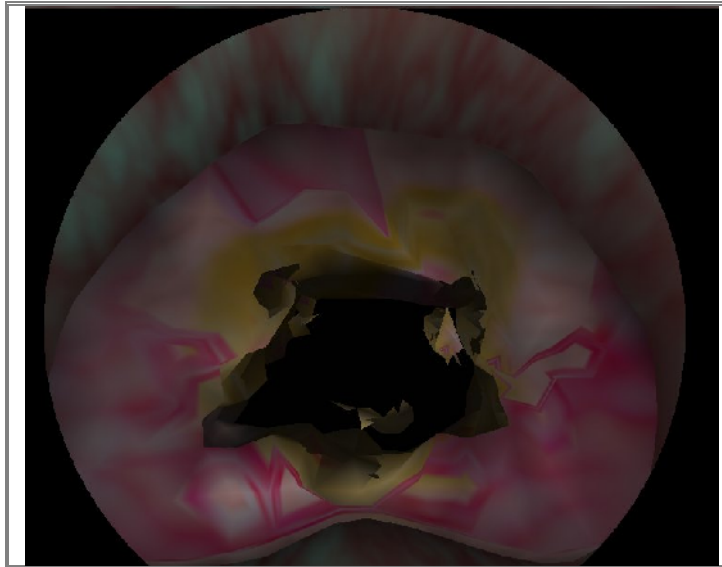


Figura 4.15 Imagen del simulador virtual RTU, una vez realizados algunos cortes.

Al verificar los resultados visuales de la figura anterior, podemos comentar que se ha logrado el objetivo principal de este trabajo de tesis. Ya que los efectos visuales que se aprecian en el simulador virtual de cirugía se acercan mucho a las condiciones reales que se presentan en el procedimiento de resección transuretral de próstata.

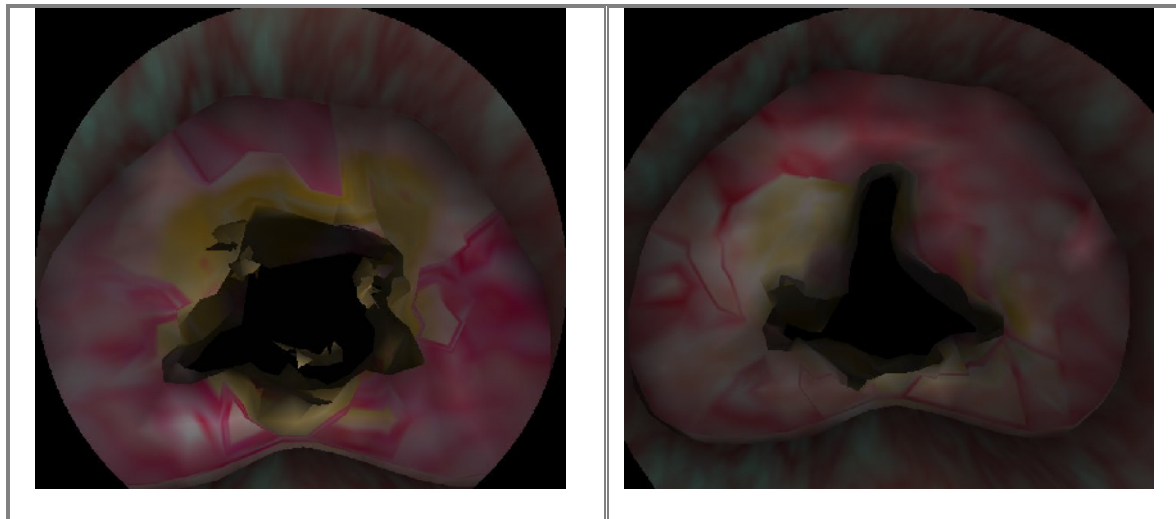


Figura 4.16 Imágenes del simulador virtual RTU, una vez realizados algunos cortes y cauterización del tejido.

Gracias al uso de métodos procedurales de textura, se pudo generar una textura útil y visualmente muy aceptable sobre el simulador virtual de cirugía. Sin que este perdiera su forma, ni sus efectos visuales. Además de permitir la revisión del interior de la próstata, teniendo la posibilidad de realizar cortes desde su interior descubriendo el tejido interno (modelado de textura sólida).

5. CONCLUSIONES Y PERSPECTIVAS

La realización de este trabajo de tesis consistió en el desarrollo de un modelo generador de texturas procedurales con apariencia de tejido blando real, para ser utilizadas en un simulador de entrenamiento del procedimiento de resección transuretral de próstata (RTU) desarrollado en el laboratorio de Análisis de Imágenes y Visualización CCADET, UNAM. Dichas texturas sintéticas debían cumplir con ser visualmente muy parecidas al procedimiento real visto en previas imágenes médicas.

Una vez hecho la integración de funciones básicas para la generación de texturas procedurales, se realizaron algunas pruebas pudiendo llegar a las siguientes conclusiones.

5.1 Alcances generales

Es importante mencionar que la apariencia de las imágenes reales del tejido blando de la próstata, cuentan con tonalidades y efectos de apariencia irregular, con lo cual es necesario trabajar con elementos que nos faciliten la simulación para lograr el mismo efecto en las texturas generadas por computadora, la forma que se encontró más viable para lograr dichos efectos fue con el uso de métodos de texturas procedurales para su generación.

Se logro generar una textura con vascularidades, logrando que algunas de las vascularidades se visualicen de un color rojo más encendido. Lo cual nos da impresión de tener un tejido inflamado o con una mayor concentración de sangre, tal y como se muestran las imágenes medicas reales.

Se logro obtener un volumen de textura (textura 3D), ayudando notablemente en la apariencia visual del simulador, ya que nos da la impresión de contar con un material continuo, sin costuras o artefactos no deseados. Contando con ello a contar con un material “sólido” con el cual seremos capaces de hacer cortes transversales sobre el material no perdiendo la calidad de la textura.

El modelo de síntesis de texturas procedurales además de mejorar la apariencia visual sobre el simulador; entre otros efectos, también se logro obtener una apariencia visual diferente al realizar la cauterización del tejido, al agregar el matiz amarillo-pardo, el cual se visualiza en el modelo al realizar algunos cortes o cauterización del tejido.

Al utilizar funciones procedurales para la generación de dichas texturas, contamos con algunas de sus ventajas que es no tener un tamaño fijo, lo cual se pueden generar imágenes de diferentes tamaños adaptándolos a las dimensiones que se requieran sin perder la calidad de detalle en la textura.

Otra ventaja de la construcción de texturas procedurales, es que al contar con los algoritmos base de las funciones básicas para la generación de texturas (*turbulencia*, *turbulencia modificada*, *terreno heterogéneo* y *fBm*), se cuenta adicionalmente con una herramienta general para la generación de diferentes texturas, las cuales pueden ser aplicables a otros modelos virtuales o de simulación computarizada.

Una de las desventajas al usar este método es que muchas veces los resultados no pueden ser 100% predecibles, ocasionando que el obtener la textura que deseemos tendremos que hacerlo mediante técnicas estadísticas muchas de ellas a base de prueba y error, tarea que nos puede llevar mucho tiempo.

Otra desventaja en la generación de textura por medio de un método procedural, es que dependiendo de la complejidad de los cálculos puede llevarse mucho tiempo de proceso en el equipo de cómputo. Lo cual se recomienda utilizar una computadora con buena capacidad de procesamiento.

En general puedo mencionar que se logro el objetivo de este trabajo de tesis, generando una síntesis de texturas procedurales la cual agrego el detalle necesario para tener un simulador virtual del procedimiento de resección transuretral de próstata que cada vez más se va asemejando a las condiciones reales.

5.2 Trabajo a futuro

El trabajo a futuro por hacer sobre este sistema de simulación se puede comentar lo siguiente.

En términos de texturización se debe seguir trabajando en mejorar aún más los detalles del tejido como son las tonalidades, las formas y la apariencia de esponjosidad en algunas partes de la glándula (próstata), dicho efecto debe venir directamente de la imagen de textura en 3D.

Otro a trabajo a futuro consiste en la implementación de sangrado. Este deberá dispararse cuando se toque o corte con el elemento de resección alguna de las vascularidades inflamadas o congestionadas.

Como trabajo futuro también queda la implementación de los algoritmos y funciones procedurales bases para la generación de texturas en un lenguaje de alto nivel (C++), incorporando en el código fuente las librerías de la tarjeta gráfica, esto con la finalidad de que los cálculos de generación de texturas se lleven a cabo en ella, logrando así que la generación de texturas se procese más rápido y eficientemente, además de contar con la posibilidad de agregar más complejidad sin que esto sacrifica el tiempo de generación y sobre todo ahorrando espacio en memoria para almacenar las texturas pre-calculadas.

Se cuenta además con una GCI para ser usada en un futuro en la generación de una base de imágenes procedurales capaces de ser usadas a diferente modelos o casos clínicos en la laboratorio de Análisis de Imágenes y Visualización, CCADET UNAM.

Finalmente para contar con una herramienta más completa capaz generar una colección más amplia de imágenes de texturas, se deben de incluir más funciones procedurales. Teniendo con ello la gran ventaja de conocer la “función base” o “fórmula” que con otros programas no podemos conocer. Con lo cual podemos adecuarlos a las necesidades que se requieran y obtener exactamente la imagen que necesitamos.

APÉNDICE

A. Código fuente: Pila de Tejido de la Próstata en 3D

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TISSUE PROSTATE 3D%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function tissueProstate3D(cube, numslices,color_rugoso, var_h_heteroterrain,var_lacunarity_heteroterrain,
var_offset_heteroterrain, var_octaves_heteroterrain, veins_scale_turbulence, veins_initoctave_turbulence,
veins_octaves_turbulence, veins_factor_turbulence, veins_scale_fbm, veins_h_fbm, veins_lacunarity_fbm,
veins_octaves_fbm, yellow_scale_turbulence, yellow_octaves_turbulence, yellow_initoctave_turbulence,
yellow_scale_fbm1, yellow_h_fbm1, yellow_lacunarity_fbm1, yellow_octaves_fbm1,yellow_scale_fbm2,
yellow_h_fbm2, yellow_lacunarity_fbm2, yellow_octaves_fbm2, roughness_scale_turbulence,
roughness_initoctave_turbulence, roughness_octaves_turbulence)

%INITIAL VALUES
t1=fix(clock);

%SIZE OF CUBE (X,Y,Z)
xmax = cube;
ymax = cube;
zmax = cube;

%FILES IMAGES
prefixfile='3D\\';
filesImage = 'ProstataAbierta\\Region%d.tif';
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INITIATION OF NOISE AND SPECTRAL FUNCTIONS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[p, g1, g2, g3] = init_noise;
spectralweights_heteroterrain = spectralWeights(var_h_heteroterrain, var_lacunarity_heteroterrain,
var_octaves_heteroterrain); %TISSUE VARIATION (HeteroTerrain)
spectralweights_fbm_veins = spectralWeights(veins_h_fbm, veins_lacunarity_fbm, veins_octaves_fbm); %GROUND
TISSUE (fbM)
spectralweights_fbm1_yellow = spectralWeights(yellow_h_fbm1, yellow_lacunarity_fbm1, yellow_octaves_fbm1);
%GROUND TISSUE (fbM)
spectralweights_fbm2_yellow = spectralWeights(yellow_h_fbm2, yellow_lacunarity_fbm2, yellow_octaves_fbm2);
%GROUND TISSUE (fbM)
%spectralweights_fbm_roughness = spectralWeights(roughness_h_fbm, roughness_lacunarity_fbm,
roughness_octaves_fbm); %GROUND TISSUE (fbM)

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PRECOMPUTES THE ROTATION MATRICES FOR ANTIALIASING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:veins_octaves_turbulence,
    if i < 6, RAUX=zeros(3,3); RAUX(1,1)=1.0; RAUX(2,2)=1.0; RAUX(3,3)=1.0;
    else RAUX=rand(3);
    end;
    RAUX(1,4)=0.0; RAUX(2,4)=0.0; RAUX(3,4)=0.0;
    RAUX(4,1)=0.0; RAUX(4,2)=0.0; RAUX(4,3)=0.0; RAUX(4,4)=1.0;
    R2(:, :, i)=RAUX;
end
for i=1:veins_octaves_fbm,
    if i < 6, RAUX=zeros(3,3); RAUX(1,1)=1.0; RAUX(2,2)=1.0; RAUX(3,3)=1.0;
    else RAUX=rand(3);
    end;
    RAUX(1,4)=0.0; RAUX(2,4)=0.0; RAUX(3,4)=0.0;
    RAUX(4,1)=0.0; RAUX(4,2)=0.0; RAUX(4,3)=0.0; RAUX(4,4)=1.0;
    R3(:, :, i)=RAUX;
end
for i=1:yellow_octaves_turbulence,
    if i < 6, RAUX=zeros(3,3); RAUX(1,1)=1.0; RAUX(2,2)=1.0; RAUX(3,3)=1.0;
    else RAUX=rand(3);
    end;
    RAUX(1,4)=0.0; RAUX(2,4)=0.0; RAUX(3,4)=0.0;
    RAUX(4,1)=0.0; RAUX(4,2)=0.0; RAUX(4,3)=0.0; RAUX(4,4)=1.0;
    R4(:, :, i)=RAUX;
end
```

```

for i=1:yellow_octaves_fbm1,
    if i < 6, RAUX=zeros(3,3); RAUX(1,1)=1.0; RAUX(2,2)=1.0; RAUX(3,3)=1.0;
    else RAUX=rand(3);
    end;
    RAUX(1,4)=0.0; RAUX(2,4)=0.0; RAUX(3,4)=0.0;
    RAUX(4,1)=0.0; RAUX(4,2)=0.0; RAUX(4,3)=0.0; RAUX(4,4)=1.0;
    R5(:, :, i)=RAUX;
end
for i=1:yellow_octaves_fbm2,
    if i < 6, RAUX=zeros(3,3); RAUX(1,1)=1.0; RAUX(2,2)=1.0; RAUX(3,3)=1.0;
    else RAUX=rand(3);
    end;
    RAUX(1,4)=0.0; RAUX(2,4)=0.0; RAUX(3,4)=0.0;
    RAUX(4,1)=0.0; RAUX(4,2)=0.0; RAUX(4,3)=0.0; RAUX(4,4)=1.0;
    R6(:, :, i)=RAUX;
end
for i=1:roughness_octaves_turbulence,
    if i < 6, RAUX=zeros(3,3); RAUX(1,1)=1.0; RAUX(2,2)=1.0; RAUX(3,3)=1.0;
    else RAUX=rand(3);
    end;
    RAUX(1,4)=0.0; RAUX(2,4)=0.0; RAUX(3,4)=0.0;
    RAUX(4,1)=0.0; RAUX(4,2)=0.0; RAUX(4,3)=0.0; RAUX(4,4)=1.0;
    R7(:, :, i)=RAUX;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% COMPUTES THE 3D DENSITY FUNCTION(TISSUE 3D) %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for z=1:zmax,

    z
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%% VEINS TISSUE (EXTERNAL) %%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Function_heteroterrain3D = tex_heteroterrainSlice( xmax, ymax, zmax, z, var_h_heteroterrain,
var_lacunarity_heteroterrain, var_octaves_heteroterrain, var_offset_heteroterrain,
spectralweights_heteroterrain, p, g3 );
    Function_heteroterrain3D = 0.2*ones(size(Function_heteroterrain3D))+Function_heteroterrain3D; %%ATTENUATION
FUNCTION
    Function_heteroterrain3D =(1/max(max(Function_heteroterrain3D)))*Function_heteroterrain3D; %NORMALIZE
FUNCTION

    Function_turb_veins3D = tex_turbSlice(xmax, ymax, zmax, z, veins_scale_turbulence,
veins_octaves_turbulence, veins_inioctave_turbulence, p, g3, R2);
    Function_turb_veins3D =(Function_turb_veins3D*(0.3*(veins_factor_turbulence))); %MULTIPLY FUNCTION BY A
CONSTANTE
    Function_turb_veins3D = ones(size(Function_turb_veins3D))-Function_turb_veins3D; %INVERT COLOR OF FUNCTION
    Function_turb_veins3D =(1/max(max(Function_turb_veins3D)))*Function_turb_veins3D;

    Function_fBm_veins3D = tex_fBmSlice(xmax, ymax, zmax, z, veins_scale_fbm, veins_h_fbm,
veins_lacunarity_fbm, veins_octaves_fbm, p, g3, spectralweights_fbm_veins, R3);
    Function_fBm_veins3D =0.92*ones(size(Function_fBm_veins3D))+0.1*Function_fBm_veins3D; %ATTENUATION FUNCTION
    Function_fBm_veins3D =(1/max(max(Function_fBm_veins3D)))*Function_fBm_veins3D; %NORMALIZE FUNCTION

    Result_Heteroterrain_x_Turbulence=Function_heteroterrain3D.*Function_turb_veins3D; %MULTIPLY FUNCTION
HETEROTERRAIN_3D BY FUNCTION TURBULENCE_3D
    Function_Value=0.92*ones(size(Result_Heteroterrain_x_Turbulence))+0.1*Result_Heteroterrain_x_Turbulence;

    %CONVERSION OF HSV TO RGB
    HSV=[];
    HSV(:, :, 1)=Function_fBm_veins3D; HSV(:, :, 2)=Result_Heteroterrain_x_Turbulence; HSV(:, :, 3)=Function_Value;
    TEX=hsv2rgb(HSV);

    %SEPARATE CANALS OF COLOR (RGB)
    TISSUEURETHRA_R(:, :, z) = TEX(:, :, 1);
    TISSUEURETHRA_G(:, :, z) = TEX(:, :, 2);
    TISSUEURETHRA_B(:, :, z) = TEX(:, :, 3);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%% YELLOWISH TISSUE %%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Function_turb_yellow3D = tex_turbSlice(xmax, ymax, zmax, z, yellow_scale_turbulence,

```

```

yellow_octaves_turbulence, yellow_initoctave_turbulence, p, g3, R4);
Function_turb_yellow3D=0.4*ones(size(Function_turb_yellow3D))+0.9*Function_turb_yellow3D;
Function_turb_yellow3D=(1/max(max(Function_turb_yellow3D)))*Function_turb_yellow3D; %NORMALIZE FUNCTION

Function_fBm1_yellow = tex_fBmSlice(xmax, ymax, zmax, z, yellow_scale_fbm1, yellow_h_fbm1,
yellow_lacunarity_fbm1, yellow_octaves_fbm1, p, g3, spectralweights_fbm1_yellow, R5);
Function_fBm1_yellow = ones(size(Function_fBm1_yellow)) - Function_fBm1_yellow; %INVERT COLOR OF FUNCTION
Function_fBm1_yellow=(1/max(max(Function_fBm1_yellow)))*Function_fBm1_yellow; %NORMALIZE FUNCTION

Result_Turbulence_x_fBm_Yellow = Function_turb_yellow3D.*Function_fBm1_yellow; %MULTIPLY FUNCTION
TURBULENCE BY FUNCTION fBm
Result_Threshold_Turb_x_fBm = threshold2(Result_Turbulence_x_fBm_Yellow,30); %APPLY THRESHOLD AT MULTIPLY
FINAL OF FUNCTION TURBULENCE BY FUNCTION fBm1

Function_fBm2_yellow = tex_fBmSlice(xmax, ymax, zmax, z, yellow_scale_fbm2, yellow_h_fbm2,
yellow_lacunarity_fbm2, yellow_octaves_fbm2, p, g3, spectralweights_fbm2_yellow, R6);
Function_fBm2_yellow = 0.8*ones(size(Function_fBm2_yellow))+0.2*Function_fBm2_yellow; % ATTENUATION
FUNCTION
Function_fBm2_yellow=(1/max(max(Function_fBm2_yellow)))*Function_fBm2_yellow; %NORMALIZE FUNCTION

%CONVERSION OF HSV TO RGB
HSV=[];
HSV(:, :,1)=Result_Threshold_Turb_x_fBm; HSV(:, :,2)=Function_turb_yellow3D; HSV(:, :,3)=Function_fBm2_yellow;
TEX=hsv2rgb(HSV);

%SEPARATE CANALS OF COLOR (RGB)
TISSUEPROSTATE_R(:, :,z) = TEX(:, :,1);
TISSUEPROSTATE_G(:, :,z) = TEX(:, :,2);
TISSUEPROSTATE_B(:, :,z) = TEX(:, :,3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Function_roughness_3D = tex_turbSliceAnisotropic(xmax, ymax, zmax, z, roughness_scale_turbulence,
10*roughness_scale_turbulence, roughness_scale_turbulence, roughness_octaves_turbulence,
roughness_initoctave_turbulence, p, g3, R7);
Function_roughness_3D=(1/max(max(Function_roughness_3D)))*Function_roughness_3D; %NORMALIZE FUNCTION

%MATRIX_ZEROS=zeros(size(Function_roughness_3D)); %GENERATE A MATRIX OF ZEROS
%MATRIX_ONES=ones(size(Function_roughness_3D))/1; %GENERATE A MATRIX OF ONES
%MATRIX_SCALE_GRAY=ones(size(Function_roughness_3D))/2.5; %GENERATE A MATRIX OF GRAY SCALE

%ADD THE VALOR DEFAULT IF THE STRING DOESN'T TRY THE VALUE OF COLOR
if (color_rugoso > 4) || (color_rugoso <= 0)
color_rugoso=4;
end

%SELECTION COLOR FINAL OF THE CAPSULE TEXTURE
%COLOR= 1; FINAL COLOR OF THE TEXTURE= GRAY
%COLOR= 2; FINAL COLOR OF THE TEXTURE= PINK
%COLOR= 3; FINAL COLOR OF THE TEXTURE= LIGHT PINK
if color_rugoso==1
%CONVERSION OF HSV TO RGB
MATRIX_ZEROS=zeros(size(Function_roughness_3D)); %GENERATE A MATRIX OF ZEROS
Function_roughness_3D=ones(size(Function_roughness_3D))-Function_roughness_3D; %INVERT COLOR OF FUNCTION
HSV=[];
HSV(:, :,1)=MATRIX_ZEROS; HSV(:, :,2)=MATRIX_ZEROS; HSV(:, :,3)=Function_roughness_3D;
TEX=hsv2rgb(HSV);

%SEPARATE CANALS OF COLOR (RGB)
TISSUECAPSULE_R(:, :,z) = TEX(:, :,1);
TISSUECAPSULE_G(:, :,z) = TEX(:, :,2);
TISSUECAPSULE_B(:, :,z) = TEX(:, :,3);

elseif color_rugoso==2
%CONVERSION OF HSV TO RGB
Function_roughness_3D=ones(size(Function_roughness_3D))-Function_roughness_3D; %INVERT COLOR OF FUNCTION
MATRIX_ZEROS=zeros(size(Function_roughness_3D)); %GENERATE A MATRIX OF ZEROS
%MATRIX_ONES=ones(size(Function_roughness_3D))/1; %GENERATE A MATRIX OF ONES
MATRIX_SCALE_GRAY=ones(size(Function_roughness_3D))/2.5; %GENERATE A MATRIX OF GRAY SCALE
HSV=[];
HSV(:, :,1)=MATRIX_ZEROS; HSV(:, :,2)=MATRIX_SCALE_GRAY; HSV(:, :,3)=Function_roughness_3D;
TEX=hsv2rgb(HSV);

```

```

%SEPARATE CANALS OF COLOR (RGB)
TISSUECAPSULE_R(:,:,z) = TEX(:,:,1);
TISSUECAPSULE_G(:,:,z) = TEX(:,:,2);
TISSUECAPSULE_B(:,:,z) = TEX(:,:,3);

elseif color_rugoso==3

    Function_Canal_Saturation= 0.3*ones(size(Function_roughness_3D))+ 0.9*Function_roughness_3D;
    Function_Canal_Value= 0.5*ones(size(Function_roughness_3D))+ 0.1*Function_roughness_3D;
    MATRIX_ZEROS=zeros(size(Function_roughness_3D)); %GENERATE A MATRIX OF ZEROS
    %Function_roughness_3D=Function_roughness_3D.*MATRIX_SCALE_GRAY; %MULTIPLY FUNCTION TURBULENCE MODIFICATE
    BY MATRIX_SCALE_GRAY

    %Roughness_fBm_3D = tex_fBmSlice(xmax, ymax, zmax, z, roughness_scale_fbm, roughness_h_fbm,
    roughness_lacunarity_fbm, roughness_octaves_fbm, p, g3, spectralweights_fbm_roughness, R8);
    %Roughness_fBm_3D=1.95*ones(size(Roughness_fBm_3D))+0.1*Roughness_fBm_3D; % ATTENUATION FUNCTION
    %Roughness_fBm_3D=(1/max(max(Roughness_fBm_3D)))*Roughness_fBm_3D; %NORMALIZE FUNCTION

    %CONVERTION OF HSV TO RGB
    HSV=[];
    HSV(:,:,1)=MATRIX_ZEROS; HSV(:,:,2)=Function_Canal_Saturation; HSV(:,:,3)=Function_Canal_Value;
    TEX=hsv2rgb(HSV);

    %SEPARATE CANALS OF COLOR (RGB)
    TISSUECAPSULE_R(:,:,z) = TEX(:,:,1);
    TISSUECAPSULE_G(:,:,z) = TEX(:,:,2);
    TISSUECAPSULE_B(:,:,z) = TEX(:,:,3);

    elseif color_rugoso==4

    Function_Canal_Saturation= 0.25*ones(size(Function_roughness_3D))+ 0.1*Function_roughness_3D;
    Function_Canal_Value= 0.9*ones(size(Function_roughness_3D))+ 0.5*Function_roughness_3D;
    MATRIX_ZEROS=zeros(size(Function_roughness_3D)); %GENERATE A MATRIX OF ZEROS
    %Function_roughness_3D=Function_roughness_3D.*MATRIX_SCALE_GRAY; %MULTIPLY FUNCTION TURBULENCE MODIFICATE
    BY MATRIX_SCALE_GRAY

    %Roughness_fBm_3D = tex_fBmSlice(xmax, ymax, zmax, z, roughness_scale_fbm, roughness_h_fbm,
    roughness_lacunarity_fbm, roughness_octaves_fbm, p, g3, spectralweights_fbm_roughness, R8);
    %Roughness_fBm_3D=1.95*ones(size(Roughness_fBm_3D))+0.1*Roughness_fBm_3D; % ATTENUATION FUNCTION
    %Roughness_fBm_3D=(1/max(max(Roughness_fBm_3D)))*Roughness_fBm_3D; %NORMALIZE FUNCTION

    %CONVERTION OF HSV TO RGB
    HSV=[];
    HSV(:,:,1)=MATRIX_ZEROS; HSV(:,:,2)=Function_Canal_Saturation; HSV(:,:,3)=Function_Canal_Value;
    TEX=hsv2rgb(HSV);

    %SEPARATE CANALS OF COLOR (RGB)
    TISSUECAPSULE_R(:,:,z) = TEX(:,:,1);
    TISSUECAPSULE_G(:,:,z) = TEX(:,:,2);
    TISSUECAPSULE_B(:,:,z) = TEX(:,:,3);
end
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% ROTATION OF FINAL CANALS OF COLOR (RGB) ROUGHNESS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for z=1:zmax,
    for y=1:ymax,
        for x=1:xmax,
            AUX_R(x,z,y) = TISSUECAPSULE_R(x,y,z);
            AUX_B(x,z,y) = TISSUECAPSULE_B(x,y,z);
            AUX_G(x,z,y) = TISSUECAPSULE_G(x,y,z);

        end;
    end;
end;
TISSUECAPSULE_R = AUX_R;
TISSUECAPSULE_G = AUX_G;
TISSUECAPSULE_B = AUX_B;

%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% INTEGRATE THE THREE TEXTURE WITH THE IMAGES %%%
%% PREVIOUS GENERATE OF THE PROSTATE %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

% VERIFICATE NUM OF IMAGES OF THE PROSTATE
offset = round((zmax - numslices))-2;
%ASSING ZERO WHEN ZMAX <= NUM OF IMAGES OF THE PROSTATE
if offset < 0, offset = 0;
else
%ASSING ONE WHEN ZMAX > NUM OF IMAGES OF THE PROSTATE
  offset=1;
end;
%LOOP FOR ADD THE TEXTURE TO THE IMAGES OF THE PROSTATE
for z=1:zmax,
  if z > offset & z<=zmax-offset %VERIFICATE RANGE THE NUM OF THE IMAGES
    zaux = z; %AUX THAT EXAMINE THE IMAGES ONE BY ONE
    if z<=numslices-1
      file = sprintf(filesImage,zaux-1); % ADD THE ROUTE OF THE FILE
    else
      file = sprintf(filesImage,numslices-1); % ADD THE ROUTE OF THE FILE
    end;
    IMAGEN_REAL = imread(file); % READ THE IMAGES OF THE PROSTATE IN B/W
    IMAGEN_REAL = imresize(IMAGEN_REAL, [xmax ymax], 'nearest' ); % SCALE THE IMAGES (X,Y)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%% SEPARATION OF CONTOUR OF THE PROSTATE %%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%% IN THE IMAGE %%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    IMAGEN_REAL=double(IMAGEN_REAL); % CHANGE VALUES OF IMAGES
    IMAGEN_REAL_COMPLE=ones(size(IMAGEN_REAL))-IMAGEN_REAL; %INVERT COLOR
    CONTORNO_EXT=imfill(IMAGEN_REAL_COMPLE); % CONTOUR EXTERIOR
    CAPSULE=ones(size(CONTORNO_EXT))-CONTORNO_EXT; %INVERT COLOR

    CONTORNO_INT= xor(IMAGEN_REAL,CAPSULE); %%%CONTOUR INTERIOR
    URETHRA=ones(size(CONTORNO_INT))-CONTORNO_INT; % %INVERT COLOR

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%% EROTION CONTOUR EXTERIOR %%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    se = strel('ball',1,1); % METHOD FOR EROTION, FUNCTION OF MATLAB
    CAPSULE_EROTION = imdilate(CAPSULE,se); % EROTION IMAGE
    h = ones(5,5) / 25; % PARAMETER FOR THE FUNCTION FILTER, VALUE PROPER OF MATLAB
    CAPSULE_ERO_FILTRE= imfilter(CAPSULE_EROTION,h); % FILTER IMAGE
    CAPSULE_ERO_FILTRE=(1/max(max(CAPSULE_ERO_FILTRE)))*CAPSULE_ERO_FILTRE; %NORMALIZE IMAGE

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%% DILATE CONTOUR INTERIOR %%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    sel = strel('rectangle', [5 5]); % METHOD FOR EROTION, FUNCTION OF MATLAB
    URETHRA_DILATE = imerode(URETHRA,sel); % DILATE IMAGE
    h = ones(5,5) / 25; % PARAMETER FOR THE FUNCTION FILTER, VALUE PROPER OF MATLAB
    URETHRA_DILATE_FIL= imfilter(URETHRA_DILATE,h); % FILTER IMAGE
    URETHRA_DILATE_FIL=(1/max(max(URETHRA_DILATE_FIL)))*URETHRA_DILATE_FIL; %NORMALIZE IMAGE

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%% INTEGRATE OF TEXTURE WITH THE CONTOURS %%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    URETHRA_DILATE_FIL_COMPLE = ones(size(URETHRA_DILATE_FIL))-URETHRA_DILATE_FIL; %IMAGE COMPLEMENT

    % THE MULTIPLICATION OF CONTOUR BY THE YELLOW TEXTURE PLUS THE MULTIPLICATION OF CONTOUR IMAGE
    COMPLEMENT BY THE VEINS TEXTURE FOR EVERY CANAL OF COLOR
    TISSUE_UREandPROS_R(:, :, zaux) = URETHRA_DILATE_FIL_COMPLE.*TISSUEURETHRA_R(:, :, zaux) +
    URETHRA_DILATE_FIL.*TISSUEPROSTATE_R(:, :, zaux);
    TISSUE_UREandPROS_G(:, :, zaux) = URETHRA_DILATE_FIL_COMPLE.*TISSUEURETHRA_G(:, :, zaux) +
    URETHRA_DILATE_FIL.*TISSUEPROSTATE_G(:, :, zaux);
    TISSUE_UREandPROS_B(:, :, zaux) = URETHRA_DILATE_FIL_COMPLE.*TISSUEURETHRA_B(:, :, zaux) +
    URETHRA_DILATE_FIL.*TISSUEPROSTATE_B(:, :, zaux);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%% INTEGRATE THE FINAL TEXTURE (CAPSULE) %%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    CAPSULE_ERO_FILTRE_COMPLE = ones(size(CAPSULE_ERO_FILTRE))-CAPSULE_ERO_FILTRE; %IMAGE COMPLEMENT

    %INTEGRATE THE FINAL ROUGHNESS TEXTURE WITH THE RESULT THE TEXTURIZATE IMAGES

```

```

        TISSUE_R(:, :, zaux) = CAPSULE_ERO_FILTRE.*TISSUECAPSULE_R(:, :, zaux) +
CAPSULE_ERO_FILTRE_COMPLE.*TISSUE_UREandPROS_R(:, :, zaux);
        TISSUE_G(:, :, zaux) = CAPSULE_ERO_FILTRE.*TISSUECAPSULE_G(:, :, zaux) +
CAPSULE_ERO_FILTRE_COMPLE.*TISSUE_UREandPROS_G(:, :, zaux);
        TISSUE_B(:, :, zaux) = CAPSULE_ERO_FILTRE.*TISSUECAPSULE_B(:, :, zaux) +
CAPSULE_ERO_FILTRE_COMPLE.*TISSUE_UREandPROS_B(:, :, zaux);

        %SEPARATE CANALS OF COLOR (RGB) OF THE FINAL IMAGE OBTAINED
        TEX(:, :, 1) = TISSUE_R(:, :, zaux);
        TEX(:, :, 2) = TISSUE_G(:, :, zaux);
        TEX(:, :, 3) = TISSUE_B(:, :, zaux);

    else
        %SEPARATE CANALS OF COLOR (RGB) OF THE FINAL IMAGE OBTAINED
        TEX(:, :, 1)= TISSUEURETHRA_R(:, :, z);
        TEX(:, :, 2)= TISSUEURETHRA_G(:, :, z);
        TEX(:, :, 3)= TISSUEURETHRA_B(:, :, z);
    end;

    % SAVE IMAGE FINAL
    file=sprintf('%s%d.bmp',prefixfile,z);
    imwrite(TEX, file);
end;
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% TIME OF RUN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

horal= t1(1,4);
min1=t1(1,5);
segundo1=t1(1,6);

t2=fix(clock);
hora2= t2(1,4);
min2=t2(1,5);
segundo2=t2(1,6);

horafinal=hora2-horal;
minfinal=min2-min1;

if segundo2 > segundo1
    segfinal=segundo2 - segundo1;
else
    segfinal=segundo1-segundo2;
end

if minfinal < 0
    var1= (horafinal*60) + (minfinal);
    var2= (var1/60); var2=fix(var2);
    var3= var1 - (var2*60);
    horafinal=var2;
    minfinal=var3;
end
sprintf ('%d : %d : %d',horafinal,minfinal,segfinal)
exit;

```

BIBLIOGRAFÍA Y REFERENCIAS WEB

Abbou Claude y Dubernard Jean-Michel, Cirugía de la próstata. Técnicas quirúrgicas, urología. España : Elsevier Masson, 2007.

Ahearn Luke, 3D game textures. USA : Focal Press, 2006.

Apodaca Anthony A., Larry Gritz, Ronen Barzel, Advanced RenderMan: Creating CGI for Motion Pictures. USA : Morgan Kaufmann, 1999.

Arámbula Cosio F., Padilla Castañeda M., Sevilla Martínez P., Computer assisted prostate surgery training. International Journal of Humanoid Robotics. Mexico, 2006. - Vol. 3.

Azarang Mohammad R. y Dunna Eduardo García, Simulación y análisis de modelos estocásticos. México: McGraw-Hill, 1996.

Azcárraga Dr. Gustavo, Urología. México : Mendez Editores, 2000.

Blender User Guide, Texture. Septiembre 2004.
<http://www.blender.org/documentation/html/c4607.html>

Capilla Pascual, Fundamentos de colorimetría. España : Universidad de Valencia, 2002 .

CEC Electromedicina, Manual Electrobisturi URO 400. Argentina : CEC.

Chen C. H., Pau L. F. y Wang S. P., Handbook of pattern recognition & computer vision. USA : World Scientific, 1998.

Clemente Ramos L.M., Rueda F. Ramasco y Platas A., Síndrome de Reabsorción Post-Resección Transuretral (R.T.U.) de Próstata: Revisión de aspectos fisiopatológicos, diagnóstico y terapéuticos. España : Actas Urológicas Españolas, 2001. - 25 : Vol. 1.

Ebert David, Musgrave F. Kenton y Peachey, Texturing & Modeling: A Procedural Approach. USA : Morgan Kaufmann, 2002.

Elias Hugo Perlin Noise. 2003.
http://freespace.virgin.net/hugo.elias/models/m_perlin.htm

ERCO-LightScout Guide, Sombreado y textura. 1997.
http://www.erco.com/guide_v2/guide_2/simulation_95/shading_2678/es/es_shading_intro_1.htm.

Ferguson R. Stuart, *Practical algorithms for 3D computer graphics*. USA : A K Peters, 2001.

Foley James D, Van Dam A., Hughes S.K., *Introducción a la graficación por computador*. Delawere, USA : Addison-Wesley Iberoamericana, 1996.

Gudlaugsson Bjarki, *Procedural Content Generation*. USA : Publication, 2006.

Hearn Donald y Baker M. Pauline, *Gráficas por computadora*. México : Prentice Hall Hispanoamericana, 1995.

Heinz-Otto Peitgen, Hartmut Jürgens, Dietmar Saupe, *Chaos and fractals: new frontiers of science*. USA : Springer, 2004.

Jagnow Robert Carl, *Stereological Techniques for Synthesizing Solid Textures*. USA : Massachusetts Institute of Technology, 2004.

Mandelbrot Benoit, *La Geometria fractal de la naturaleza*. Barcelona, España : Tusquets, 1997.

Mayo Clinic, *Transurethral resection of the prostate (TURP)*. 2000. - <http://www.mayoclinic.com/health/turp/MY00633>.

McConnell Jeffrey J., *Computer Graphics. Theory into practice*. EUA : Jones & Bartlett Publishers, 2006.

Noise Perlin, *Noise and Turbulence*. 1997. <http://mrl.nyu.edu/~perlin/doc/oscar.html#noise>.

Padilla Castañeda M. y Arámbula Cosío F., *Deformable for turp sugery simulation*. *Computer and Graphics* 28, 2004.

Perlin Ken, *Making Noise*. Diciembre de 1999. - <http://www.noisemachine.com/talk1/>.

Poissant Yves, *Musgrave. Materials Plugins*. 2002. <http://www.ypoart.com/downloads/Musgrave.htm>.

Resnick Robert, *Física Vol. 1*. México : CECSA, 1999.

Shirley Peter y Ashikhmin Michael, *Fundamentals of computer graphics*. USA : A k Peters, 2005.

Shreiner Dave, Woo Mason y Neider Jackie, *The official Guide to Learning OpenGL. Versión 2.1*. Massachusetts, USA : Addison-Wesley, 2007.

Stewart Ian, *The New Mathematics of Chaos*. USA : Penguin Press, 1997.

Stoelting Robert K. y Dierdorf Stephen F., *Anestesia y enfermedad coexistente.* España : Elsevier, 2003. - Vol. Cuarta.

Texture Procedural, *Procedural Texture.* 2006.
http://www.timaxmedia.com/html/help/Layer_Type_Procedural_Texture.htm.

Turcotte Donald Lawson, *Fractals and chaos in geology and geophysics.* USA : Cambridge University Press, 1997.

Turk Greg, *Generating Textures on Arbitrary Surfaces.* EUA : (A Thesis Proposal) , 1991.

Universidad Nacional del Sur Dpto. Ciencias e Ingeniería de la Computación, *Computación Gráfica.* 2008. - <http://cs.uns.edu.ar/cg/clases.htm>.

Valdéz Soriano David, *Generación de modelos de mallas de tetraedros para simuladores quirúrgicos.* México DF : Tesis Profesional. Facultad de Ingeniería. UNAM, 2008.

Van Leigh, *LightWave 3D 8 Texturing.* USA : Wordware Publishing, Inc., 2004.

Wein Alan J. y Kavoussi Louis R., *Urología de Campbell & Walsh.* USA : Editorial Medica Panamericana, 2008.