



Benemérita Universidad  
Autónoma de Puebla

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN  
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

---

Tesis que para obtener el grado de:  
Maestro en Ciencias de la Computación

Detección y Localización de Defectos  
en la Superficie del Esmalte Dental en  
3 Dimensiones.

Presenta:

Javier Caldera Miguel

Asesores:

Dr. José Italo Cortez

Dr. Manuel Rubín Falfán

Puebla, Pue.

Septiembre del 2005



# ÍNDICE

## Índice

### Índice de Figuras

### Introducción

### Capítulo 1. Antecedentes Estomatológicos y Computacionales

	i
<b>Capítulo 1. Antecedentes Estomatológicos y Computacionales</b>	<b>1</b>
1.1 Morfología de las Piezas Dentales	2
1.1.1 Tejidos Peridentarios.	2
1.1.2 Capas del diente	3
1.1.3 Dentición	5
1.1.4 Enfermedades Dentales	7
1.2 Métodos Utilizados para el Diagnóstico Estomatológico	9
1.2.1 Rayos X	9
1.2.1.1 Composición de la Película	10
1.2.1.2 Características de la Imagen Radiográfica	10
1.2.1.3 Evaluación de las Radiografías	10
1.2.1.4 Radiografías Digitales	11
1.2.1.5 Sistema de Diagnóstico con Placa Radiográfica Digital	11
1.2.1.6 La Tecnología de la Placa Radiográfica en Odontología	11
1.2.2 Transiluminación	12
1.2.3 Explorador	12
1.2.4 Láser Dental	13
1.2.4.1 Historia del Láser	13
1.2.4.2 Clasificación del Láser	14
1.2.4.3 Diagnóstico por Medio de Láser	15
1.2.5 Cámaras Intraorales	15
1.2.5.1 Imagen de la Cámara Intraoral Color.	15
1.2.5.2 Cámara Dental Visión en Televisión.	16
1.2.6 Método de Diagnóstico para la Detección de Fisuras Dentales	16
1.2.7 Historia Dental	17
1.2.7.1 Examen Subjetivo	17
1.2.7.2 Examen Visual	17
1.2.7.3 Examen Exploratorio	17
1.2.7.4 Tinción	17
1.2.7.5 Transiluminación	18
1.3 Métodos Ópticos	18
1.4 Método de Reflectometría	18
1.5 Mecanismo Automático	21
1.6 Tarjeta de Adquisición de datos.	22
1.7 Métodos de Eliminación de Ruido	23
1.8 Topologías	24
1.9 Reconstrucción en 3 Dimensiones	30



1.9.1	Imágenes de Rango	31
1.9.2	Proceso de Reconstrucción	31
1.9.3	Aplicaciones	33
1.9.4	Digitalización de Imágenes de Rango	36
1.10	Lenguajes de Programación.	37
1.10.1	Lenguaje de Programación Delphi	41
1.11	Imágenes Digitales.	42
1.12	Gráficos	45
1.12.1	APIs de Gráficos 3D	49
<b>Capítulo 2. Planteamiento del Problema</b>		<b>53</b>
<b>Capítulo 3. Sistema para el Análisis de la Rugosidad del Esmalte Dental.</b>		<b>57</b>
3.1	Dispositivo Mecánico	58
3.2	Adquisición de Datos	60
3.3	Imagen en 2 Dimensiones	62
3.4	Eliminación de Ruido	64
3.5	Interpolación	66
3.6	Representación de la rugosidad en 3D	67
3.7	Reconstrucción del Volumen.	68
3.8	Software de Representación de Datos	69
<b>Capítulo 4. Experimentos y Resultados</b>		<b>76</b>
<b>Conclusiones</b>		<b>93</b>
<b>Aportaciones</b>		<b>94</b>
<b>Perspectivas</b>		<b>95</b>
<b>Bibliografía</b>		<b>96</b>
<b>Apéndice A (Códigos principales del software de representación de datos)</b>		<b>98</b>
<b>Apéndice B (Diagrama principal del software de representación de datos)</b>		<b>105</b>
<b>Apéndice C (OpenGL en Delphi)</b>		<b>107</b>
<b>Apéndice D (Trabajos Publicados)</b>		<b>120</b>



## ÍNDICE DE FIGURAS

Figura 1	Estructura interna de una pieza dental	5
Figura 2	Clasificación de las piezas dentales según su localización	7
Figura 3	Caries	7
Figura 4	Placa Bacteriana	8
Figura 5	Gingivitis	8
Figura 6	Modelo del método de reflectometría	19
Figura 7	Mecanismo automático para el análisis de las piezas dentales	22
Figura 8	Tracto Emisor-Receptor	22
Figura 9	Diagrama de bloques del sistema de adquisición de datos	23
Figura 10	Vecindad de un punto con coordenadas (x,y)	25
Figura 11	Líneas digitales	28
Figura 12	Conexión de dos regiones por una línea limitadas por una curva cerrada sin intersección con la curva	29
Figura 13	Imágenes de Rango Tomadas desde Múltiples Vistas	32
Figura 14	Molde de Fabricación	34
Figura 15	Animación por computador	35
Figura 16	Piezas de Museo	35
Figura 17	Taxonomía de los métodos de adquisición de rango activo	37
Figura 18	Diagrama de bloques del sistema completo	58
Figura 19	Mecanismo automático para la adquisición de datos	59
Figura 20	Tracto Emisor-Receptor	60
Figura 21	Diagrama de la Adquisición de datos	60
Figura 22	Archivo de adquisición de datos de un premolar, sano, el paciente es de sexo masculino y tiene 21 años de edad	62
Figura 23	Método de Interpolación	67
Figura 24	Método de los 8 vecinos	68
Figura 25	Método del Cubo	69
Figura 26	Pantalla principal del sistema de representación de datos	70
Figura 27	Modulo de la rugosidad de piezas dentales	71
Figura 28	Imagen en 2D y con radiografía	72
Figura 29	Interfaz de la animación en 3 dimensiones	72
Figura 30	Modulo de la rugosidad de materiales dieléctricos	74
Figura 31	Interfaz de reconstrucción de la rugosidad de materiales dieléctricos en 3 dimensiones	75
Figura 32	Piezas dentales extraídas y almacenadas	77
Figura 33	Datos en forma matricial y en forma de imagen de la cara vestibular de una molar	79
Figura 34	Imágenes del filtro de la Transformada de Fourier de la cara vestibular de una premolar	79
Figura 35	Imagen limpia de la cara vestibular de una molar	80
Figura 36	Reconstrucción en 3D de la rugosidad de la cara vestibular de una molar con textura en sólido a escala de grises	80
Figura 37	Reconstrucción en 3D de la rugosidad de la cara vestibular de una molar con textura en mallas a escala de grises	81
Figura 38	Reconstrucción en 3D de la rugosidad de la cara vestibular de una molar con textura en sólido a color	81
Figura 39	Reconstrucción en 3D de la rugosidad de la cara vestibular de una	82



	molar con textura en mallas a color	
Figura 40	Resalte del defecto de la pieza dental a color	82
Figura 41	Resalte del defecto de la pieza dental con la división anatómica	83
Figura 42	Datos en forma matricial y en imagen de 2 dimensiones en escala de grises de una pieza dental	84
Figura 43	Reconstrucción de tres dimensiones de una pieza dental sin defecto	84
Figura 44	Reconstrucción de tres dimensiones de una pieza dental sin defecto con otro ángulo de visión	85
Figura 45	Imagen de la rugosidad de la pieza dental analizada con microscopio metalográfico	85
Figura 46	Realce del defecto de una pieza dental en tres dimensiones	86
Figura 47	Realce del defecto de una pieza dental en tres dimensiones, observada desde otro ángulo	86
Figura 48	Imagen de una pieza dental con defecto analizada con microscopio.	87
Figura 49	Imagen de la parte central de una pieza dental sin defecto	88
Figura 50	Imagen de la parte central de una pieza dental con defecto	88
Figura 51	Imagen de una pieza dental toma de la parte inferior hacia la superior de la parte de en medio de una pieza dental sin defecto	89
Figura 52	Imagen de una pieza dental toma de la parte inferior hacia la superior de la parte de en medio de una pieza dental con defecto	89
Figura 53	Representación de los datos de un dieléctrico en forma matricial y en imagen en 2 dimensiones	90
Figura 54	Imagen en 3 dimensiones de la rugosidad de una mica en escala de grises	91
Figura 55	Imagen en 3 dimensiones de la rugosidad de una mica a color y con resalte del defecto	91
Figura 56	Imagen en 3 dimensiones de la rugosidad de una mica a color, con resalte de defecto, tomada desde otro ángulo	92
Figura 57	Imagen en 3 dimensiones de la rugosidad de una mica a escala de grises, con resalte de defecto	92



## INTRODUCCIÓN

Actualmente los especialistas en estomatología realizan sus diagnósticos con dispositivos totalmente invasivos como: rayos X, explorador, etc., ocasionando problemas en la superficie del esmalte dental y por lo tanto al paciente, además no detecta los defectos menores a 150 micras.

Para la detección de defectos, el especialista lleva a cabo un diagnóstico que consiste en:

*Síntomas subjetivos:* se le pide al paciente que señale el órgano dentario que presenta sensibilidad, recordando que, de todas maneras, los pacientes pueden equivocarse.

*Examen visual:* el estomatólogo comienza por realizar un examen extraoral iniciándolo por la cara del paciente, verifica si presenta potentes músculos en la mandíbula, lo que puede indicar el hábito de sobrecarga de los órganos dentarios durante la masticación, articulación temporo-mandibular (ATM), ganglios.

*Examen intraoral:* se aplica el método de exploración física: inspección, palpación, auscultación y percusión y en los casos que lo requiera se emplearán los auxiliares de diagnóstico como radiografías, estudios de gabinete y verifica la presencia de desgaste en los órganos dentarios.

*Palpación:* la palpación consiste en precisar a través del sentido del tacto o presión leve, la consistencia del tejido. *Percusión:* esta prueba, al igual que otras pruebas clínicas, se realiza primero en un órgano dentario adyacente normal para que el paciente pueda distinguir la diferencia en intensidad y sensación de dolor o molestia.

*Exploración:* se explora la superficie del órgano dentario con la punta de un instrumento afilado, la punta más pequeña es de 200 micras.

*Transiluminación:* se aplica una fuente lumínica, directamente sobre la



superficie dentaria. Se dirige el haz de luz perpendicular al plano en el cual se sospecha se encuentra la fisura, ésta impedirá el paso de la luz. Aquellos órganos dentarios sanos, incluyendo aquéllos que presentan líneas de grietas, transmitirán la luz a través de la corona. *Rayos X*: la radiología representa un área muy importante dentro del campo médico y estomatológico ya que es un componente básico dentro del diagnóstico en terminaciones cervicales de restauraciones. La radiación emitida por los rayos X es absorbida por los órganos relacionados con el tratamiento estomatológico como: ojos, cerebro, tiroides, hipófisis, pulmones, lengua, cerebro, etc.

Los instrumentos utilizados para hacer un análisis en las piezas dentales se han modernizado y se han desarrollado nuevos métodos para hacer un análisis mas preciso y encontrar defectos como: caries, desmineralización, fisuras. Los métodos invasivos modernos son: rayos X con placas digitales donde las imágenes aparecen en una computadora y el proceso es más rápido y el diagnóstico es mas claro ya que se pueden ver las imágenes en RGB o a escala de grises. Láser dental: permite diagnosticar caries y fisuras prematuras pero el tratamiento es invasivo, lo importante es que conserva gran parte de la estructura del diente. Explorador: se utiliza para detectar fisuras y caries pero sólo detecta fisuras mayores de 200 micras. Los métodos no invasivos son: tabletas reveladoras, métodos químicos, seda o hilo dental prueba de Snyder, transiluminación con fibra óptica para imágenes digitales, cámaras intraorales a color y escala de grises. Con lo anteriormente expuesto se comprueba que no existen tecnologías no invasivas que detecten fisuras menores de 200 micras.

En las áreas de diseño, automatización, arqueología, robótica inteligente, medicina se han desarrollado dispositivos que realizan una reconstrucción de objetos en tres dimensiones, esto es, el proceso de reproducir en memoria de una



computadora objetos reales, manteniendo sus características físicas como: dimensiones, volumen y forma.

Con el método de *inspección*, la digitalización de la figura ayuda a determinar dónde y en qué medida difiere el modelo computarizado y el objeto real. Estas diferencias pueden servir como guía para modificar el proceso de manufactura hasta que el modelo y el objeto real sean aceptables. Para los científicos, estos modelos computarizados ofrecen la oportunidad de estudiar y medir artefactos, usando poderosas herramientas de computación. En efectos especiales, juegos y mundos virtuales las imágenes sintéticas están jugando un papel cada vez más importante en la creación de efectos especiales para el cine. Además, los video-juegos y el hardware de juegos se están inclinando lentamente hacia los gráficos interactivos en tercera dimensión. La realidad virtual como medio para experimentar mundos simulados también crece en popularidad. En medicina la tomografía digital es de gran importancia para analizar los tejidos del ser humano, haciendo cortes a las imágenes en tres dimensiones y así proporcionar un mejor diagnóstico.

Todas estas aplicaciones requieren modelos en 3D que puedan ser tomados de la vida real. La digitalización de formas de modelos físicos será esencial para poblar ambientes sintéticos.

En el área de visión artificial existe una variedad de métodos cuyo objetivo es obtener un algoritmo que permita realizar la conexión del conjunto de puntos representativos del objeto en forma de elementos de superficie, ya sean triángulos, cuadrados o cualquier otra forma geométrica.

Por lo anterior en el Laboratorio de Métodos no Invasivos para el Diagnostico Estomatológico de la Facultad de Estomatología se está desarrollando tecnología no invasiva para la detección de defectos en la superficie del esmalte



dental. En este proyecto de investigación se realizó un sistema para localizar defectos en las piezas dentales en especial se detecta la desmineralización con un escaneo a 50 micras, lo cual ayudará al especialista a tomar mejores alternativas para un diagnóstico.

Este trabajo de Investigación está compuesto de 4 capítulos:

En el Capítulo 1 se presentan conceptos estomatológicos como: clasificación de los dientes, tipo de dientes, las enfermedades que pueden tener los dientes además se presentan las metodologías más importantes que hasta este momento se han desarrollado para este tipo de trabajos en el área de cómputo como: visión computacional, gráficos, reconstrucción, imágenes digitales, topologías.

En el capítulo 2 se plantea el problema desde un punto de vista estomatológico y se lleva a una solución tanto física como computacional, al dar una solución nos damos cuenta que existen varios problemas y que existen diferentes áreas que dan posibles soluciones.

En el capítulo 3 se presentan las herramientas aplicadas en la solución de nuestro problema. También se presenta el software desarrollado e implementado para observar la rugosidad del esmalte dental en una imagen en 2 dimensiones y en animación en tres dimensiones, que permite detectar la desmineralización temprana y muestra el lugar en donde se encuentra.

En el capítulo 4 se muestran algunos experimentos realizados con piezas dentales, en particular, con molares. El experimento se desarrolló con el sistema realizado, comparando los resultados con un microscopio metalográfico y rayos X. para garantizar la efectividad del sistema.



Posteriormente se presentan los resultados obtenidos y las comparaciones que se realizaron y finalmente se presentan las conclusiones de este sistema, además los trabajos publicados.

# CAPÍTULO 1

## **ANTECEDENTES ESTOMATOLÓGICOS Y COMPUTACIONALES**



## 1.1 Morfología de las Piezas Dentales

La estructura de las piezas dentales se compone por el odontón que es la unidad compuesta por el diente más los tejidos peridentarios, que corresponden al ligamento periodontal, hueso alveolar y a la encía. Cada pieza dental está compuesta por tres capas dentales que son: componentes mineralizados, el esmalte, la dentina y el cemento, además de una porción central de tejido conectivo gelatinoso suave llamada pulpa [1].

### 1.1.1 Tejidos Peridentarios.

- ◆ **Ligamento periodontal:** es un tejido conectivo denso, ricamente vascular, caracterizado por grupos principales de fibras de colágeno I, distribuidos en patrones predeterminados para resistir las fuerzas de la masticación. Los extremos de estas fibras se embeben en el alvéolo y el cemento a manera de fibras de Sharpey, que permiten al ligamento periodontal suspender a la pieza dentaria dentro de su alvéolo. Presenta fibras nerviosas vegetativas vasomotoras, de dolor y propioceptivas. Está situado entre la región del cemento de la raíz y el alvéolo óseo y mide 0,5mm de amplitud.
- ◆ **Hueso alveolar:** es la prolongación del maxilar inferior y superior, que se divide en compartimentos llamados alvéolos, separados entre sí por un tabique interalveolar óseo. El alvéolo presenta tres regiones que son: placas corticales, la esponjosa y la alveolar. Las placas corticales se disponen en sentido lingual y labial, formando una repisa de sostén muy resistente de hueso compacto, revestida por hueso poroso, que constituye la capa esponjosa, ésta rodea a una capa delgada de hueso compacto, el hueso alveolar cuya morfología es reflejo de la forma de la raíz suspendida en su interior. En las tres porciones del hueso alveolar hay conductos por los cuales pasan las arterias nutricias, que luego llegan al ligamento periodontal.



- ◆ **Encías:** son elementos de membrana mucosa resistente. Consisten en un epitelio plano pluriestratificado queratinizado por completo o paraqueratinizado y debajo un tejido conectivo denso con fibras de colágeno I que constituyen grupos de fibras principales que se parecen a las del ligamento periodontal.

### 1.1.2 Capas del diente

Cada pieza dental se compone de:

- ◆ *Esmalte:* es el tejido externo, duro y resistente, es la cubierta protectora de la corona anatómica, es translúcido, y su color se debe al color de la dentina subyacente. Consiste en un 98% de hidroxapatita cálcica y en un 2% de agua y material orgánico, este último principalmente conformado por glucoproteínas de alto peso molecular parecidos a la queratina, ricas en amelogeninas y enamelinas. El esmalte es producido por los ameloblastos, que lo elaboran todos los días en segmentos de 4 a 8 mm. Se adhieren entre sí segmentos en forma de bastoncillo sucesivos, con lo que forman bastoncillos de esmalte o prismas del esmalte. Los ameloblastos mueren antes de que la pieza dentaria se haya abierto paso hacia la cavidad bucal, por lo que el cuerpo no puede reparar el esmalte. Como los prismas del esmalte se formaron por unión de segmentos, entre estas uniones queda unas líneas, manifiestas desde el punto de vista histológico, llamadas líneas de Retzius. Al observarlas con luz polarizada se observan también bandas perpendiculares a las líneas de Retzius, llamadas estrías de Hunter-Schreger [1].
- ◆ *Dentina:* es el tejido intermedio más blando que el esmalte y es el segundo tejido más duro del cuerpo, es amarillenta, y su alto grado de elasticidad protege al esmalte suprayacente contra las fracturas. Se compone en un 65% de hidroxapatita de calcio, en un 25% de materiales orgánicos y en un



10% de agua. La sustancia orgánica es en su mayoría colágena I acompañada de proteoglicanos y glucoproteínas.

La dentina es producida por los odontoblastos, que se ubican entre la dentina y la pulpa dentaria, y que conservan su relación con la dentina durante toda la vida del diente y ésta puede autorrepararse. La dentina presenta los canalículos dentarios, que contienen las prolongaciones citoplasmáticas de los procesos odontoblásticos. La dentina también se forma en segmentos de 4mm a 8mm, por lo que se presentan al microscopio líneas, llamadas líneas de Owen, análogas a las líneas de Retzius [1].

- ◆ **Cemento:** Corresponde a un tejido óseo especial, sin irrigación ni inervación. Se compone en un 55% de hidroxapatita cálcica y en 45% de agua. Se restringe a la raíz del diente y en su región apical presenta los cementocitos, que lo elaboraron y que se encuentran en los osteocitos del hueso, esta región del cemento se denomina cemento celular. La región coronal del cemento carece de cementocitos y se denomina cemento acelular. Ambos cementos presentan cementoblastos. Las fibras colágenas del ligamento periodontal (fibras de Sharpey) se encuentran embebidas en el cemento y se unen al alvéolo, fijando el diente al alvéolo. El cemento se puede resorber por células del tipo de los osteoclastos conocidas como odontoclastos. Este proceso se observa en la exfoliación (caída de los dientes deciduales).
- ◆ **Pulpa:** es el tejido intermedio (normalmente llamado *nervio*), blando de color rosado, engloba los vasos sanguíneos y terminaciones nerviosas. Está compuesta de tejido conectivo gelatinoso rico en proteoglicanos, con extensa inervación e irrigación. La pulpa se comunica con el ligamento periodontal a través del agujero apical o foramen apical, pequeña abertura situada en la punta de cada raíz. Vasos y nervios entran a través de estas aberturas. Suele subclasificarse la pulpa en tres zonas concéntricas



alrededor de un núcleo central: la odontoblástica que corresponde a una sola capa de odontoblastos, la zona libre de células y bajo ésta la zona rica en células, que consiste en fibroblastos y células mesenquimatosas, y rodea el centro de la pulpa. El centro de la pulpa se parece a la mayor parte de los otros tejidos conectivos laxos, salvo que carece de células adiposas. Las fibras nerviosas son simpáticas, de acción vasomotora, y sensitivas, que transmiten sensaciones de dolor [1].

En la figura 1 se muestran los tejidos y las capas de las piezas dentales.

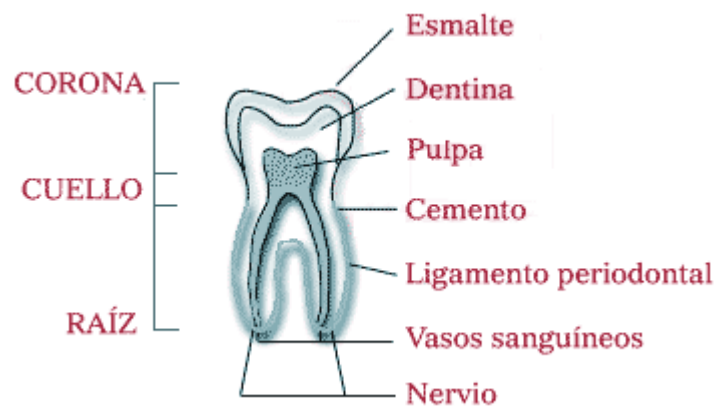


Figura1. Estructura interna de una pieza dental

### 1.1.3 Dentición

Hasta los 8 o 9 años de edad, los seres humanos sólo poseen 20 dientes, la llamada *dentición de leche* o temporal, que será sustituida por un total de 32 piezas que constituirán la dentición definitiva, existiendo tres grupos de dientes con funciones específicas [2].

➤ *Dentición de leche*

- ◆ Incisivo central (6-8 meses)
- ◆ Incisivo lateral (8-10 meses)
- ◆ Canino (16-20 meses)
- ◆ Primer molar (12-16 meses)
- ◆ Segundo molar (12-16 meses)



➤ *Dentición permanente*

- ◆ Canino (9-12 años)
- ◆ Incisivo central (7-9 años)
- ◆ Incisivo lateral (7-9 años)
- ◆ Primer premolar (10-12 años)
- ◆ Segundo premolar (10-12 años)
- ◆ Primer molar (6-7 años)
- ◆ Segundo molar (11-13 años)
- ◆ Tercer molar (17 años o más)

➤ *Tipos de dientes definitivos.*

Por la forma de la corona y por su función, hay cuatro tipos de dientes:

- ◆ *Incisivos* (8 piezas): dientes anteriores con borde afilado, cortan los alimentos
- ◆ *Caninos* (4 piezas): con forma de cúspide puntiaguada, también llamados colmillos, desgarran los alimentos.
- ◆ *Premolares* (8 piezas): poseen dos cúspides puntiaguadas, desgarran aplastan los alimentos.
- ◆ *Molares* (12 piezas): cúspides anchas, trituran los alimentos.

En la figura 3 se muestra de forma gráfica la clasificación de la dentición.

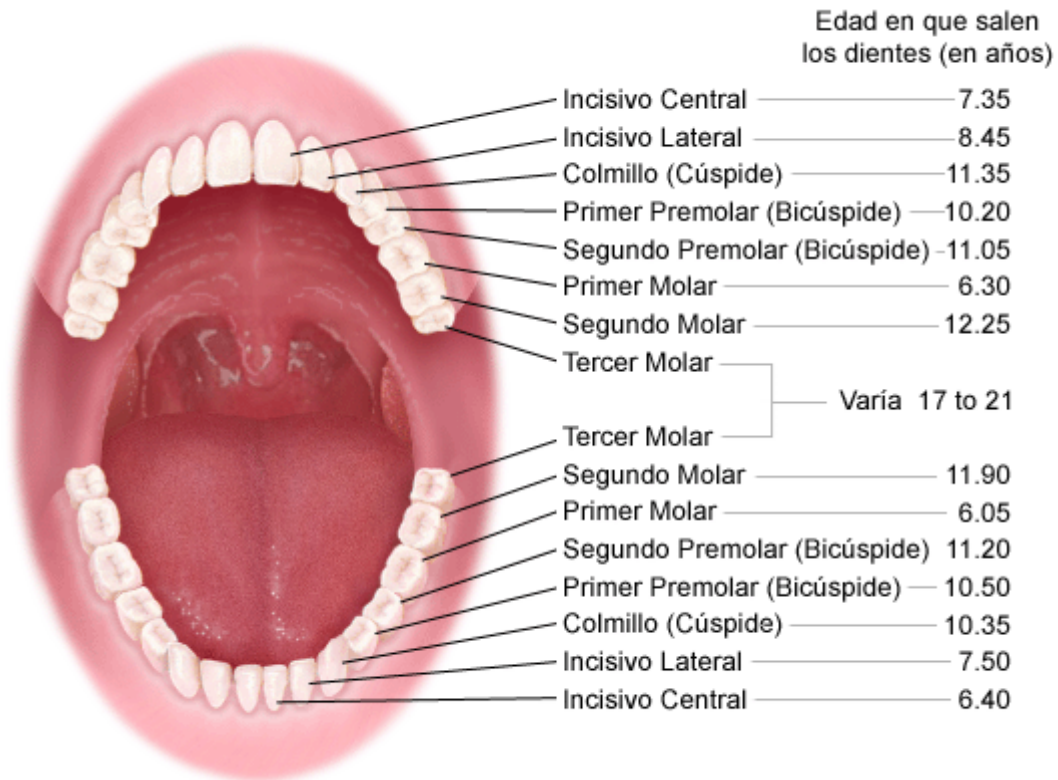


Figura 2. Clasificación de las piezas dentales según su localización

### 1.1.4 Enfermedades Dentales

- ◆ *Caries*: es la destrucción del tejido dental causada por los ácidos liberados por bacterias depositadas en la superficie de los dientes [3], ver figura 3.



Figura 3. Caries

- ◆ *Placa bacteriana*: es la masa de gérmenes dañinos que se encuentran en la boca y se fijan a los dientes. Algunos tipos de placa bacteriana causan la caries dental. Otros tipos de placa causan enfermedades de las encías como: encías rojas, hinchadas o sangrantes (gingivitis) [4]. Si la enfermedad de las encías es ignorada, los tejidos que mantienen a los dientes en su lugar se destruyen. La placa dental difícilmente puede ser vista, a menos que esté teñida, como se muestra en la figura 4.

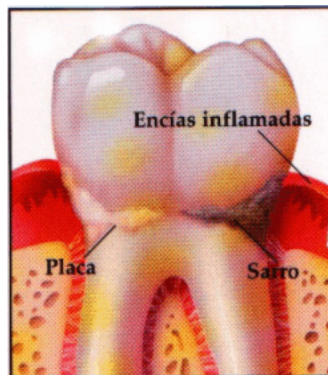


Figura 4. Placa Bacteriana

- ◆ *Gingivitis*: es la inflamación de las encías causada por los depósitos de la placa bacteriana, al producir toxinas que producen inflamación del tejido gingival, ver figura 5. Existen distintos grados de gingivitis. En general, se manifiestan por encía enrojecida, inflamada, hipersensible y sangrante, puede llegar a la periodontitis o piorrea con pérdida progresiva del hueso de la encía y del anclaje del diente [3].



Figura 5. Gingivitis



- ◆ Pulpitis: es la inflamación dolorosa de la pulpa dentaria. La provocan principalmente la caries y traumatismos dentarios [3].
- ◆ Bruxismo: es el rechinar de dientes. Se produce más frecuentemente durante el sueño y puede ser causado por distintas patologías [3].

## 1.2 Métodos Utilizados para el Diagnóstico

### Estomatológico

A pesar de que estos métodos se han modernizado siguen siendo invasivos, a continuación se describirán estos métodos.

#### 1.2.1 Rayos X

La radiología representa un área muy importante dentro del campo médico y odontológico ya que es un componente básico dentro del diagnóstico.

En el área de la endodoncia, la toma de radiografías es de vital importancia durante las diferentes etapas del tratamiento, por lo tanto, es esencial obtener radiografías de alta calidad diagnóstica para llevar a cabo una terapia endodóntica exitosa [4].

El uso de la computadora en la odontología ha marcado un gran cambio en la práctica. De esta manera, se han desarrollado equipos de alta tecnología en el área de la radiología que permiten tanto el almacenamiento [5], como, la visualización digital de las radiografías intraorales. También el desarrollo de sistemas computarizados como la tomografía ofrece una visión tridimensional para el estudio de la morfología de los dientes con fines didácticos y también de investigación.



### **1.2.1.1 Composición de la Película**

La película radiográfica está compuesta por una emulsión y una base. La emulsión está compuesta de cristales de haluro de plata que son fotosensibles y una matriz de gelatina, la cual tiene la función de suspender estos cristales. Los cristales de haluro contienen bromuro de plata y yoduro de plata. Estos últimos poseen cristales más grandes, lo que permite usar menores dosis de radiación. Existen películas con doble emulsión, las cuales para obtener la imagen requieren menor cantidad de radiación y las que presentan una sola emulsión, logran imágenes más detalladas.

### **1.2.1.2 Características de la Imagen Radiográfica**

La radiografía se evalúa según diferentes características las cuales influirán directamente en la calidad de la imagen, estas características son:

- ◆ Densidad radiográfica
- ◆ Contraste radiográfico
- ◆ Detalle
- ◆ Velo y radiación dispersa o secundaria.

### **1.2.1.3 Evaluación de las Radiografías**

De acuerdo a la filosofía de la radio protección es necesario evaluar adecuadamente todas las radiografías, para asegurarse de que la información obtenida en ellas sea completa y beneficiosa para el paciente. Hay tres pasos que abarcan estos factores en relación al manejo del paciente:

- ◆ Reconocimiento de características
- ◆ Interpretación
- ◆ Manejo de las necesidades del paciente



#### **1.2.1.4 Radiografías Digitales**

En un pasado lejano, las radiografías servían para decidir si debía extraerse o tratarse un diente y el paciente debía esperar los resultados del cuarto oscuro, en cambio, hoy en día, el paciente está totalmente implicado en los procesos de diagnóstico, profilaxis y tratamiento debido al uso creciente de las imágenes digitales para el diagnóstico dental.

#### **1.2.1.5 Sistema de Diagnóstico con Placa Radiográfica Digital**

Con las radiografías panorámicas y las Ceph (*Orthophos Plus Ceph* ofrece telerradiografía y radiografía panorámica sobre película y *Orthophos Plus DS Ceph* telerradiografía y radiografía panorámica digital. Con 16 programas de radiografía panorámica y distintas técnicas telerradiográficas, puede hacer frente a todas las exigencias de ortodoncia y cirugía maxilofacial), las películas analógicas o los casetes guía son fácilmente sustituibles por las placas radiográficas.

El aparato de rayos X está listo para su uso y digitalizar imágenes sin necesidad de aplicar ningún tipo de conversión. Inversión justificable y necesaria para entrar en el mundo del diagnóstico digital.

#### **1.2.1.6 La Tecnología de la Placa Radiográfica en Odontología**

En la radiología médica, las placas radiográficas se utilizan de forma tan habitual como las radiografías por rayos X. Las potenciales ventajas del Dürr VistaScan y de la tecnología de placas radiográficas son:

- ◆ Latitud de exposición.
- ◆ Completa gama de escala de grises.
- ◆ Alta resolución.
- ◆ Dosis de radiación reducida.



### 1.2.2 Transiluminación

Odontólogos de Estados Unidos están utilizando ya un nuevo dispositivo, llamado Difoti (digital imaging fiber optic transillumination), que permite encontrar la caries que no es vista en las radiografías tradicionales y observar todos los rincones de la boca en el monitor de la computadora ubicada al lado del sillón del paciente. Los dentistas aseguran que hay caries que pasan inadvertidas por los rayos X, pero que con este dispositivo se pueden ver con claridad. El Difoti forma parte de toda una serie de tecnologías digitales nuevas que ayudan a llenar varios vacíos a la hora de hacer diagnósticos. La transiluminación es muy útil en el sector anterior bucal, debido a que las piezas dentales tienen un espesor bucolingual reducido. Pudiéndose identificar zonas oscuras en el esmalte debido a la refracción de la luz que ocasiona la caries o la obturación dental. Dicha técnica se puede realizar con cualquier lámpara de polimerización eliminando el gasto de un equipamiento auxiliar [5].

En el sector posterior el valor predictivo de caries dental es nulo debido al mayor espesor bucodental de las piezas examinadas. Se necesita una radiación más penetrante, como los rayos Röntgen, la radiografía periapical es el método de diagnóstico más adecuado, teniendo una relación costo beneficio aceptable. Solo son necesarias ocho radiografías periapicales. Pero la crítica a este método, es que la radiación Röntgen detecta caries de punto de contacto en la pieza dental solo cuando se ha perdido el 50% del calcio. Es decir que la lesión cariosa es observada cuando se ha perdido más de la mitad de la densidad de esmalte.

### 1.2.3 Explorador

Antes de conceptualizar el explorador como un instrumento para el diagnóstico de la caries dental, se debe saber que este instrumento es parte esencial de la “tríada odontológica”, la cual está compuesta por el espejo dental, la pinza algodонера y el explorador [5].



Este conjunto de instrumentos comprenden los elementos esenciales para el examen y diagnóstico de la cavidad bucal en una consulta odontológica.

El explorador es un instrumento empleado para el examen clínico del paciente, que permite descubrir cavidades cariosas y recidivas de obturaciones efectuadas (para detectar filtraciones en márgenes de aquellas, debe estar provisto de una punta extremadamente aguzada de aproximadamente  $0.2 \mu\text{m}$ ). Existen diferentes tipos de exploradores, según la cantidad de las partes activas:

- ◆ *Explorador de extremo simple*

Cuando la parte activa del instrumento se encuentra en un extremo del mango, siglas (SE).

- ◆ *Explorador de extremo doble*

Cuando las partes activas del instrumento se encuentran en ambos extremos del mango, siglas (DE).

## 1.2.4 Láser Dental

Existen alteraciones a nivel de esmalte muy difíciles de detectar en estadios iniciales. Con este método se puede diagnosticar lesiones iniciales y realizar tratamientos mínimo invasivos, que permiten conservar gran parte de la estructura del diente [5].

### 1.2.4.1 Historia del Láser

Albert Einstein en 1916 predice en su Teoría de la Relatividad que se puede crear una luz que fuera coherente. La luz ordinaria se considera luz incoherente porque todas las formas de energía que la conforman son diferentes, y se dispersan o viajan en todas direcciones, mientras que la luz láser está formada por energía de una misma longitud de onda lo que hace monocromática y direccional formando un haz de luz coherente donde sus ondas siempre viajan en una misma dirección. El creciente auge de modernos equipos emisores de láser de baja potencia



encuentra un sinnúmero de aplicaciones en la práctica diaria odontológica por sus efectos analgésicos, antiinflamatorios y bioestimulantes.

#### 1.2.4.2 Clasificación del Láser

Existen muchos y diversos tipos de láser que pueden ser clasificados en dos grandes grupos:

- Láser Quirúrgico. Los láseres quirúrgicos tienen una potencia muy alta y una energía termal que les da la capacidad de cortar, vaporizar o coagular los tejidos y debido a estas propiedades sus aplicaciones dentro del campo de la odontología son muchas.

Son utilizados para:

- ◆ Para tejidos blandos
- ◆ Para tejidos duros
- ◆ Para foto polimerización

- Láser Terapéutico. Los láseres terapéuticos se emiten con menor potencia que los láseres quirúrgicos y su efecto no es termal por lo que no se utiliza para cortes de tejidos.

Son utilizados para:

- ◆ Terapéuticos
- ◆ Para diagnóstico.

El efecto del láser terapéutico se debe a la interacción de la luz con los procesos metabólicos celulares por lo que a este tipo de láser se le llama bioestimulante por su excelente estimulación biológica celular.

Los láser terapéuticos tienen como medio activo el Arseniuro de Galio y Aluminio (As, Ga, Al) o el Helio Neón (He, Ne).



La profundidad de la energía de este láser en tejido óseo es de 1 cm. Mientras que en tejidos blandos su penetración puede ser de 2 a 5 cm.

#### **1.2.4.3 Diagnóstico por Medio de Láser**

Este procedimiento se basa en las propiedades de fluorescencia de los tejidos dentales, ya que valora las diferentes respuestas que tienen el tejido desmineralizado o con caries y el tejido dental sano, cuando éstos son iluminados con luz láser.

Un diodo láser genera un rayo de luz con una longitud de onda definida, que incide sobre la superficie del diente. Este rayo láser genera sobre el tejido la reflexión de luz fluorescente que será medida y evaluada por un sistema electrónico diferenciando: el tejido dental desmineralizado del tejido dental sano, mediante una señal acústica y numérica.

### **1.2.5 Cámaras Intraorales**

#### **1.2.5.1 Composición de la Cámara Intraoral Color.**

- ◆ Posee un diámetro de 2.5 cm de largo por 1 cm de ancho, lo que permite introducirlo en la boca del paciente [6].
- ◆ Cuenta con un lente de aumento color regulable.
- ◆ Posee un cable de 2 metros de largo.
- ◆ En el extremo del mismo posee una ficha de tipo telefónico que se conecta a la interfase.
- ◆ Entrada para cable RCA de VIDEO
- ◆ Entrada para fuente de electricidad que se conecta a 110 / 220 volts.

##### **1.2.5.1.1 Imagen de la Cámara Intraoral Color.**



- ◆ La imagen de una cámara se mide por líneas de resolución, la Cámara Intraoral Dental Visión cuenta con 450 líneas, al igual que las Cámaras más costosas, debido a esto, se obtiene la misma calidad de imagen.
- ◆ Esta Cámara Intraoral cuenta con un lente de aumento que permite ampliar hasta 500 veces las imágenes de una pieza dental, de esta forma las lesiones visibles se magnifican y las invisibles, como pequeñas grietas y sarro, se ven claramente.
- ◆ Cuenta con un foco manual que se gradúa sólo con un dedo, con la misma mano con la que se sostiene la cámara, y una vez ajustada se pueden tomar imágenes de 1cm a 10cm sin variarlo.
- ◆ La cámara posee un sistema de auto corrección de la imagen que se activa por diferentes intensidades de luminosidad.
- ◆ Este sistema de alta tecnología es exclusivo de la Cámara Intraoral Dental Visión la cual le permitirá filmar a su paciente con la intensidad lumínica que le sea más cómodo trabajar, sin observar alteración alguna en claridad o nitidez en la imagen.

#### **1.2.5.2 Cámara Dental Visión en Televisión.**

Es apta para utilizar en cualquier tamaño de TV sin que puedan observarse alteraciones en la calidad de imagen, desde un monitor de 9 pulgadas hasta un proyector de cine gigante sin que se produzcan modificaciones en la nitidez de la misma. Una vez conectada se tienen imágenes del paciente en forma de video [6].

#### **1.2.6 Método de Diagnóstico para la Detección de Fisuras Dentales**

Los especialistas en estomatología llevan a cabo una metodología para realizar un diagnóstico que consiste en lo siguiente:



### **1.2.6.1 Historia Dental**

Verificar si existió un antecedente previo y repetido de ajustes oclusales, que tan sólo aliviaban en forma temporaria la sintomatología, o si fue evaluado por diversos profesionales sin que éstos llegaran a un diagnóstico concluyente.

Verificar si hay o hubo antecedentes de enfermedad periodontal, con extensas pérdidas óseas en el área [3, 4].

### **1.2.6.2 Examen Subjetivo**

Se pide al paciente que señale el órgano dentario que presenta sensibilidad o dolor, recordando que, de todas maneras, los pacientes pueden equivocarse.

### **1.2.6.3 Examen Visual**

Se examina la cara del paciente, verificando si presenta potentes músculos en la mandíbula, lo que puede indicar el hábito de sobrecarga de los órganos dentarios durante la masticación. Luego verifican la presencia de desgaste, que pueden indicar antecedentes de apretamiento.

### **1.2.6.4 Examen Exploratorio**

Se explora la superficie del órgano dentario con la punta de un instrumento afilado, cuyo grosor mínimo es de 200  $\mu\text{m}$ .

### **1.2.6.5 Tinción**

Las fisuras pueden hacerse evidentes al teñirlas. Un colorante, tal como el azul de metileno, puede ser aplicado en la superficie externa del órgano dentario.



### 1.2.6.6 Transiluminación

Durante la transiluminación, se aplica una fibra óptica o una fuente lumínica similar, directamente sobre la superficie dentaria. Se dirige el haz de luz perpendicular al plano en el cual se sospecha se encuentra la fisura. La fisura impedirá el paso de la luz. Aquéllos órganos dentarios sanos, incluyendo aquéllos que presentan líneas de grietas, transmitirán la luz a través de la corona.

## 1.3 Métodos Ópticos

Los Métodos ópticos son aquellos que miden la radiación electromagnética que emana de la materia o que interacciona con ella. Se incluye todos los campos del espectro desde los rayos gamma hasta las ondas de radio. Los métodos ópticos se dividen en dos:

*Métodos espectroscópicos:* se basan en la medida de la intensidad y la longitud de onda de la energía radiante.

*Métodos no espectroscópicos:* no miden espectro y no están relacionados con las transiciones entre estados de energía característicos sino que se basan en una interacción entre la radiación electromagnética y la materia que produce como resultado un cambio en la dirección o en las propiedades físicas de la radiación electromagnética, en este grupo se encuentran refracción, la reflexión la dispersión refractiva, la dispersión, las interferencias, la difracción y la polarización. En el presente trabajo se describirá el método de reflectometría ya que es el método en el que se basa el sistema.

### 1.3.1 Método de Reflectometría

El método de Reflectometría consiste en hacer incidir un rayo de luz sobre una superficie y el reflejo muestra la cantidad de luminosidad que se encuentra sobre ésta, como se muestra en la figura 9, con este método se analiza la rugosidad de las superficie [7].

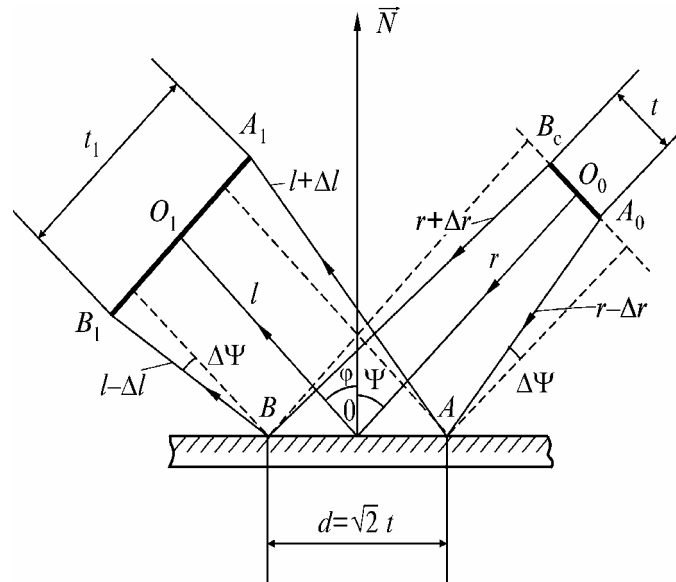


Figura 6. Modelo del método de reflectometría

En base a la figura 6 se puede establecer el sistema de ecuaciones para determinar la respuesta de luminosidad de emisión y la intensidad de luminosidad de reflexión.

• **Respuesta de emisión lumínica.**

Como el haz de luz de incidencia cae sobre una superficie y la luz que se propaga no es lineal se toma el rayo central para asegurar que la luz emitida no se disperse en su totalidad.

La respuesta de luz emitida se calcula siguiendo la figura 6 dando como resultado (1), (2), (3), (4) y (5).

$$d = \sqrt{2}t \dots\dots\dots (1)$$

$$\Delta r = \frac{\sqrt{2}}{4} d \dots\dots\dots (2)$$



$$\Delta l = \frac{2t + \sqrt{2d}}{4} \dots\dots\dots (3)$$

$$\Delta \psi = \text{arctg} \left( \frac{\sqrt{2d - 2t}}{4r - \sqrt{2d}} \right) \dots\dots\dots (4)$$

$$t_1 = 2(t + l) + \sqrt{2d} \dots\dots\dots (5)$$

Con (1), (2), (3), (4) y (5) se encuentra la respuesta de de luz expresada por E (6).

$$E = \frac{I_0 \cos \psi}{r^2} \dots\dots\dots (6)$$

Donde:

$I_0$  - intensidad de caída del rayo de luz.

$r^2$  - distancia del emisor hasta la superficie.

$\psi$  - ángulo de inclinación del emisor con respecto a la superficie.

$\cos \psi$  - la forma de cono con la que se está trabajando.

• **Intensidad de luminosidad de reflexión.**

La intensidad de reflexión viene dada por (7).

$$\Phi = \bar{\rho} I_0 S \frac{2(r+l)C}{2r+d \cos \psi} \left[ 1 - \pi^2 h^2 \cos \psi \left( \frac{1}{\lambda_2^2} - \frac{1}{\lambda_1^2} \right) \right] \dots\dots\dots (7)$$

Donde:

$\lambda_1, \lambda_2$  - valores fronterizos del espectro del haz

S - superficie

C - está dada por (8)

$$C = 2 \left[ \frac{32\sqrt{2dr}}{(8r^2 - d^2)^2} - \frac{\text{tg} \psi \sin \left( \text{arctg} \frac{d - \sqrt{2t}}{2\sqrt{2r - d}} \right)}{r^2} \right] \dots\dots\dots (8)$$

Para minimizar (7) se utiliza (9) y (10)



$$C_1 = \frac{2(r+l)C}{2r+d \cos \psi} \dots\dots\dots (9)$$

$$C_* = C_1 \pi^2 \cos^2 \psi \left( \frac{1}{\lambda_2^2} - \frac{1}{\lambda_1^2} \right) \dots\dots\dots (10)$$

Obteniendo (11)

$$\Phi = \bar{\rho} I_0 S(C_1 - C_* h^2) \dots\dots\dots (11)$$

Donde:

$\bar{\rho}$  - es el grado de polarización.

Finalmente la ecuación de conversión de intensidad de luminosidad esta dada por (12):

$$U = k \bar{\rho} I_0 S(C_1 - C_* h^2) \dots\dots\dots (12)$$

Donde:

$k$  - es el coeficiente de Transferencia del convertidor.

### 1.3.1 Mecanismo Automático

De acuerdo al modelo matemático del método de reflectometría se desarrolló un mecanismo automático, el cual consta de 4 motores a pasos, dos colocados en el plano cartesiano X, Y, como se observa en la figura 7 y los otros colocados en el tracto Emisor-Receptor (E-M), para hacer los movimientos de los ángulos tanto del emisor como del receptor, éste se puede observar en la figura 8. El mecanismo hace el escaneo de la superficie con los ángulos óptimos para tener un mejor análisis [8]

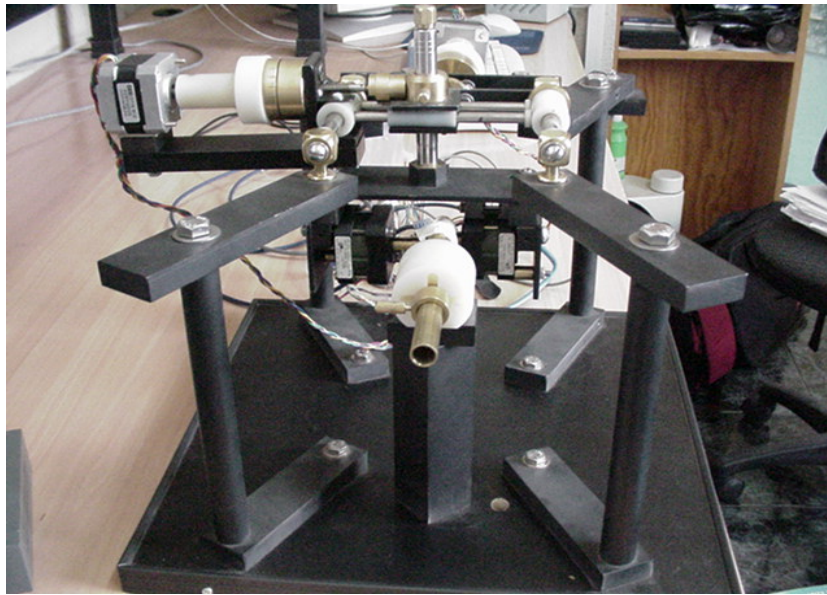


Figura 7. Mecanismo automático para el análisis de las piezas dentales

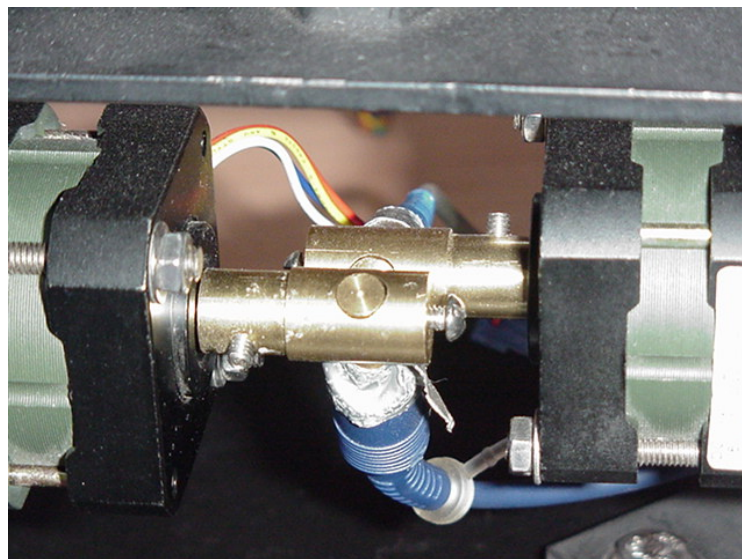


Figura 8. Tracto Emisor-Receptor

### 1.3.2 Tarjeta de Adquisición de datos.

El mecanismo se conecta a una tarjeta de adquisición de datos para enviar los resultados a una computadora haciendo una conversión de luz a voltaje analógico y posteriormente convertirlo a voltaje digital para hacer su conversión del número decimal y así el valor obtenido guardarlo en un archivo para su tratamiento

posterior [8]. En la figura 9 se muestra el diagrama de bloques del sistema de adquisición completo.

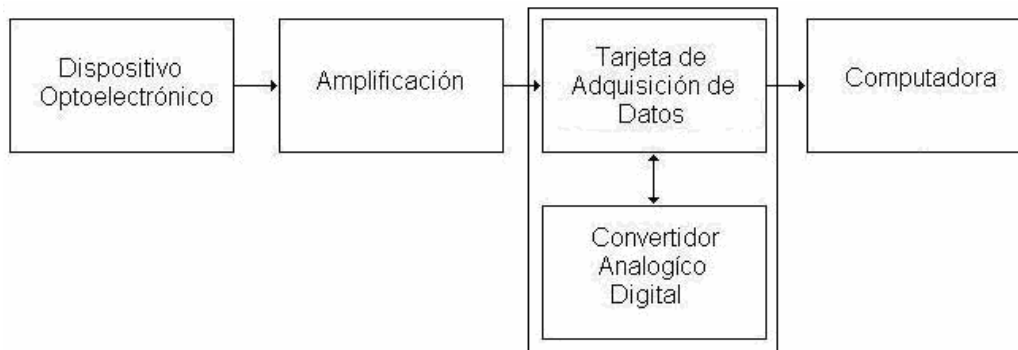


Figura 9. Diagrama de bloques del sistema de adquisición de datos

## 1.4 Métodos de Eliminación de Ruido

Los datos obtenidos por un sistema de adquisición utilizando cualquier método óptico o acústico se puede transformar a una imagen [9], ésta se puede convertir a imagen RGB o a escala de grises, estas imágenes o datos pueden presentar gran cantidad de ruido por las características propias del sistema de adquisición de datos y de los dispositivos electrónicos. Por este motivo es necesario implementar filtros que permitan tener como entrada una señal y como salida se obtendrá una señal transformada. Estas señales se caracterizan por tener una frecuencia con respecto al tiempo a estas frecuencias se les conoce con el nombre de frecuencias temporales, las frecuencias que se utilizan en imágenes digitales se le llama frecuencias espaciales, las imágenes con altas frecuencia espacial, son las que cambian periódicamente el valor de los niveles de intensidad o niveles de gris en un intervalo espacial pequeño o distancia pequeña de la imagen. Por lo tanto los niveles de gris cambian de forma abrupta de un nivel de gris a otro. Las bajas frecuencias espaciales corresponden a cambios más lentos en la variación de los niveles de gris donde los cambios ocurren gradualmente de una posición a otra de la imagen [10].



Al grupo de transformada en el dominio de la frecuencia pertenecen tres tipos de transformadas que son: Transformada de Fourier, Transformada del Coseno y Transformada de Walsh- Hadamar. Una transformada más actual y que se ha llevado con gran éxito en el tratamiento de imágenes se le conoce como Wavelet u onditas en su traducción pero solo es viable para encontrar borde o en comprimir imágenes ya que hace una linealización de valores en un rango si es que este dato se encuentra fuera de este rango.

## 1.5 Topologías

La topología es la rama de las matemáticas que estudia las propiedades del espacio que son invariantes por homeomorfismo. Se trata de propiedades no métricas, por ejemplo, una circunferencia es topológicamente equivalente a un cuadrado, aunque sus propiedades métricas sean diferentes.

Una imagen digital consta de una serie de píxeles dispuestos en una matriz rectangular cuyos elementos son números enteros, que corresponden a los niveles de cuantización en la escala de grises. Con esta disposición puede considerarse que las imágenes digitales presentan una serie de propiedades tanto métricas como topológicas, que requieren su consideración para el procesamiento posterior.

A continuación se muestran algunas propiedades de las imágenes que son importantes como:

### ❖ **Propiedades Topológicas:**

Las propiedades topológicas de las imágenes son aquellas que son invariantes a transformaciones debidas al estiramiento de una hoja de goma. Imaginemos un pequeño balón de goma con objetos pintados en él, las propiedades topológicas son aquellas que son invariantes al inflado del



balón de goma. El inflado no cambia la contigüedad de las partes del objeto ni el número de huecos en las regiones.

❖ **Vecindad**

Se dice que todo píxel  $p$ , de coordenadas  $(x,y)$  tiene cuatro píxeles que establecen con él una relación de vecindad horizontal o vertical, que son:

Horizontal:  $(x-1,y)$  y  $(x+1,y)$  Vertical:  $(x,y-1)$  y  $(x,y+1)$

Estos 4 píxeles definen lo que se conoce como *entorno de vecindad-4* y se refiere como  $E4(p)$ .

Los cuatro vecinos diagonales de  $p$  tienen coordenadas

$$(x-1,y-1), (x+1,y-1), (x-1,y+1) \text{ y } (x+1,y+1)$$

refiriéndose a ellos como  $ED(p)$ . Estos píxeles junto con los  $E4(p)$  se llaman los *vecinos-8* de  $p$  y se denotan como  $E8(p)$ .

Naturalmente, existen las excepciones dadas cuando el píxel  $(x,y)$  es un punto del borde de la imagen, en cuyo caso algunos de los vecinos definidos anteriormente no existen. En la figura 10 se puede observar de manera gráfica la vecindad de un punto [10].

$(x-1,y-1)$	$(x,y-1)$	$(x+1,y-1)$
$(x-1,y)$	$(x,y)$	$(x+1,y)$
$(x-1,y+1)$	$(x,y+1)$	$(x+1,y+1)$

Figura 10. Vecindad de un punto con coordenadas  $(x,y)$



❖ **Conectividad**

Sea  $V$  el conjunto de valores de intensidad de los píxeles que se permiten estén adyacentes; por ejemplo, si sólo se desea que exista conectividad entre los píxeles con intensidades 80,81 y 83, entonces  $V = \{80, 81, 83\}$ . Se consideran tres tipos básicos de conectividad:

1. Conectividad-4: dos píxeles  $p$  y  $q$  con valores de  $V$  están 4-conectados si  $q$  está en el conjunto  $E_4(p)$ .
2. Conectividad-8: dos píxeles  $p$  y  $q$  con valores de  $V$  están 8-conectados si  $q$  está en el conjunto  $E_8(p)$ .
3. Conectividad- $m$  (mixta): dos píxeles  $p$  y  $q$  con valores de  $V$  están  $m$ -conectados si  $q$  está en el conjunto  $E_4(p)$  o  $q$  está en  $E_D(p)$  y  $E_4(q) \cap E_4(q)$

Un píxel  $p$  es contiguo a otro píxel  $q$  si están conectados. Se puede definir la adyacencia -4, -8, dependiendo del tipo de conectividad especificada. Dos subconjuntos imagen  $S1$  y  $S2$  son contiguos si algún píxel de  $S1$  es contiguo a algún píxel de  $S2$ .

Un camino desde el píxel  $p$  con coordenadas  $(x,y)$  hasta un píxel  $q$  con coordenadas  $(s,t)$  es una secuencia de varios píxeles con coordenadas (18),

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

donde:

$(x_0, y_0) = (x, y)$  y  $(x_n, y_n) = (s, t)$ ,  $(x_i, y_i)$  son adyacentes a  $(x_{i-1}, y_{i-1})$ , con  $1 \leq i \leq n$ .  $n$  - es la longitud del camino. Se pueden definir caminos -4, -8, dependiendo del tipo de adyacencia usada.

Si  $p$  y  $q$  son píxeles de un subconjunto de imagen  $S$ , entonces  $p$  está conectado a  $q$  en  $S$  si existe un camino desde  $p$  hasta  $q$  formado de píxeles pertenecientes a  $S$ . Dado un píxel  $p$  cualquiera de  $S$ , el conjunto de píxeles de  $S$  que están conectados a  $p$  se llama componente conectado de  $S$ . Se



deduce que dos píxeles cualesquiera de un componente conectado están a su vez conectados entre sí y que los componentes conectados distintos son disjuntos.

Un camino simple es un camino sin píxeles repetidos y un camino cerrado es un camino simple en el cual el primer píxel es un vecino del último.

Una región es un conjunto de píxeles en la cual hay un camino entre cualquier par de sus píxeles y todos los píxeles de este camino pertenecen al conjunto. Si hay un camino entre dos píxeles de la imagen esos píxeles se denominan contiguos. Alternativamente, una región es el conjunto de píxeles donde cada par de píxeles en el conjunto es contiguo. La relación “ser contiguo” es reflexiva, simétrica y transitiva y por tanto define una descomposición del conjunto (en este caso la imagen) en clases de equivalencia, que serán las regiones.

Supongamos que  $R_j$  son regiones disjuntas en la imagen que se creó con la relación ser contiguo, y además suponer para evitar casos especiales, que esas regiones no tocan los límites de la imagen. Sea la región  $R$  la unión de todas las regiones  $R_j$  y el conjunto  $R_c$  el complementario de la región  $R$  con respecto a la imagen. El subconjunto de  $R_c$  que es contiguo con los límites de la imagen se denomina el fondo y el resto del conjunto complementario  $R_c$  se denominan huecos. Si no hay huecos en una región decimos que es simplemente contigua y con huecos múltiplemente contigua. Según lo anterior, el concepto de región utiliza sólo la propiedad “ser contigua”. No obstante, suele ser habitual llamar a algunas regiones en la imagen objetos; el proceso que dice qué regiones en la imagen corresponden a los objetos de la escena, se llama segmentación de la imagen. Un sencillo ejemplo para diferenciar estos conceptos lo

constituye el texto impreso en esta misma página, donde las letras son los objetos, las áreas rodeadas por las propias letras son los huecos, por ejemplo el área dentro de la letra 'O'. El resto de las partes blancas del papel es el fondo.

Las definiciones de vecindad y contigüedad en las imágenes digitales crean algunas paradojas dignas de mención. La figura 11 muestra dos segmentos de líneas digitales con  $45^\circ$  de inclinación. Si se utiliza vecindad-4, las líneas no son contiguas en cada uno de sus puntos. Aun surge un conflicto mayor al considerar las líneas perpendiculares a la diagonal, las líneas de la parte superior derecha tienen un punto común y por tanto se interceptan, sin embargo, las líneas de la parte inferior izquierda no se interceptan puesto que no tienen ningún punto en común.

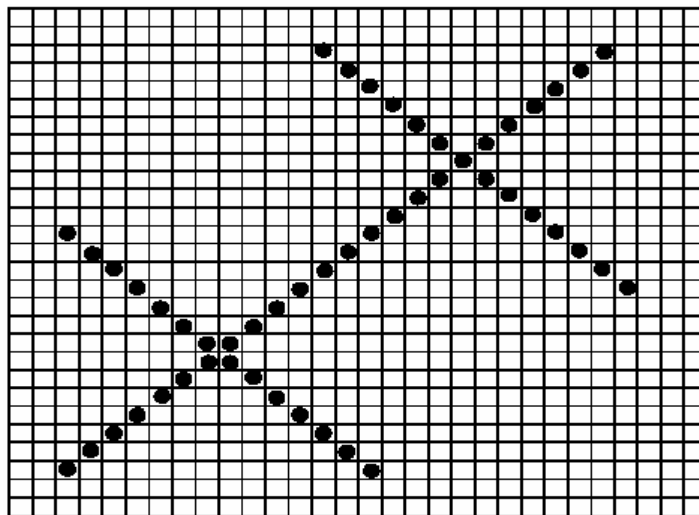


Figura 11. Líneas digitales

Por otro lado, se sabe de la geometría Euclídea que cada curva cerrada (por ejemplo un círculo) divide el plano en dos regiones, tal y como se muestra en la figura 12. Esto implica que la parte interior y la exterior de la curva constituyen una única región porque todos los píxeles de la línea pertenecen a esa única región. Esto resulta ser una nueva paradoja.



Una posible solución a estas paradojas consiste en tratar los objetos utilizando vecindades  $-4$  y el fondo vecindades  $-8$  o viceversa. Estos problemas son típicos en rejillas con celdas cuadradas, si las celdas fueran hexagonales con una geometría de panal de abejas estos problemas estarían resueltos. Sin embargo, aparecen otros, tales como la dificultad para expresar la transformada de Fourier en ellos. Además muchos dispositivos tienen una estructura de celdas cuadradas.

La frontera de la región  $R$  es el conjunto de píxeles dentro de la región que tienen uno o más vecinos fuera de  $R$ . Ésta es una definición intuitiva al considerar la frontera de una región como un conjunto de puntos en el límite de la misma. Esta definición está referida a lo que se conoce como *frontera interna* para distinguirla de la *frontera externa* que es la frontera del fondo (por tanto su complemento) de la región.

El borde es una propiedad local de un píxel y su vecindad inmediata, se trata de un vector dado por una magnitud y dirección. Existe una diferencia substancial entre frontera y borde. La frontera es un concepto global relacionado con la región, mientras el borde expresa propiedades locales de una función de imagen. La frontera y los bordes están relacionados entre sí. Una posibilidad para encontrar fronteras consiste en encadenar los puntos de borde con un gradiente elevado.

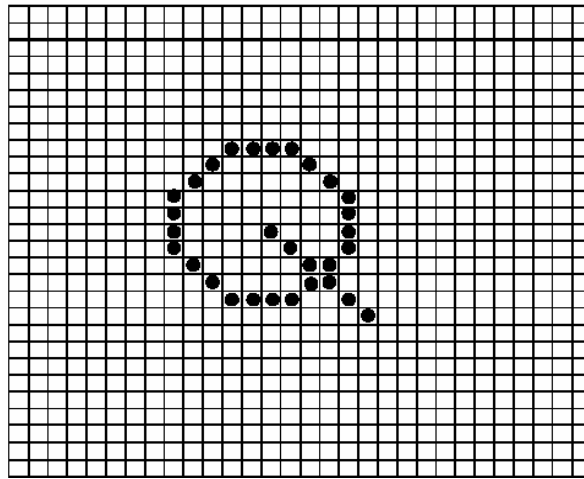


Figura 12. Conexión de dos regiones por una línea limitada por una curva cerrada sin intersección con la curva

#### ❖ **Medidas de distancias**

La distancia entre puntos con coordenadas  $(x,y)$  y  $(s,t)$  puede ser definida de varias formas diferentes:

- Distancia Euclídea:  $D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$
- Distancia D4 (llamada distancia de bloques de casas):  

$$D_4(p, q) = |x-s| + |y-t|$$
- Distancia D8 (llamada distancia de bloques de ajedrez):  

$$D_8(p, q) = \max\{|x-s|, |y-t|\}$$

## 1.6 Reconstrucción en 3 Dimensiones

La reconstrucción 3D es el proceso mediante el cual, objetos reales son reproducidos en la memoria de una computadora, manteniendo sus características físicas (dimensiones, volumen y forma). Existen dentro de la visión artificial, una variedad de métodos de reconstrucción, cuyo objetivo principal es obtener un algoritmo que sea capaz de realizar la conexión del conjunto de puntos



representativos del objeto en forma de elementos de superficie, ya sean triángulos, cuadrados o cualquier otra forma geométrica [11].

El problema de la representación y reconstrucción de formas tridimensionales ha recibido una enorme atención en investigaciones de visión en la última década. El interés surge debido a que la teoría de formas podría tener aplicaciones en una amplia variedad de campos, a saber: diseño, automatización de manufactura, mapeo de terrenos, conducción de vehículos, arqueología, restauración de obras de arte, vigilancia, robots inteligentes, entre otros. Pero además de cualquier aplicación práctica, el problema tiene mucho interés matemático y científico [11], en particular se tratará la reconstrucción en 3D por Imágenes de Rango.

### **1.6.1 Imágenes de Rango**

Las imágenes de intensidad son de uso limitado en términos de estimación de superficies. Los valores del píxel están relacionados con la geometría de la superficie sólo indirectamente. Las imágenes de rango codifican la posición de la superficie directamente, por lo tanto, la forma puede ser obtenida fácilmente. Cada píxel de una imagen de rango, expresa la distancia entre un marco de referencia conocido y un punto de vista visible en la escena. Por lo tanto, una imagen de rango reproduce la estructura 3D de una escena. Las imágenes de rango son una clase especial de imágenes digitales, conocidas como imágenes de profundidad, mapas de profundidad, mapas  $x$ ,  $y$ ,  $z$ , perfiles de superficies e imágenes 2.5D.

Las imágenes de rango pueden ser representadas en dos formas básicas: 1) es una lista de coordenadas 3D en un marco de referencia dado (nube de puntos), para lo cual no se requiere ningún orden específico, 2) es una matriz de valores de profundidad de puntos a lo largo de las direcciones de los ejes  $x$ ,  $y$  de la imagen, lo cual hace explícito la organización espacial [11].

### 1.6.2 Proceso de Reconstrucción

Cada imagen provee una descripción detallada de un objeto, en la forma en que es observado desde un punto de vista. Una sola imagen de rango no logra adquirir todos los lados de un objeto; así que, para reconstruir toda la forma de un objeto, se requieren tomar múltiples imágenes de rango como se muestra en la figura 13. [12]

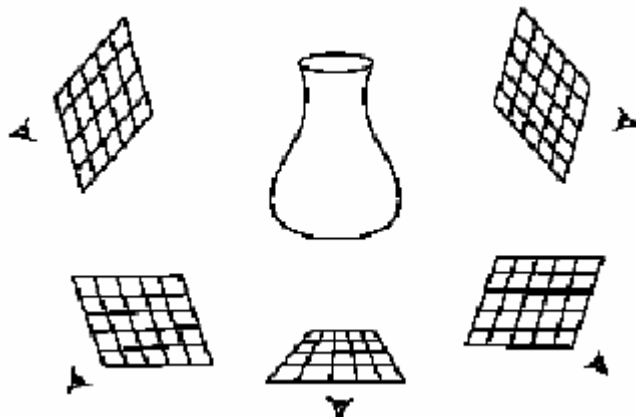


Figura 13. Imágenes de Rango Tomadas desde Múltiples Vistas

En efecto, muchos objetos son demasiado complejos para ser capturados con un pequeño número de imágenes de rango. Por ejemplo, desde direcciones mutuamente ortogonales y sus opuestos (ejemplo: las seis caras de un cubo). Debido a estas exclusiones mutuas, un objeto puede requerir un gran número de imágenes de rango para ver cada punto visible. En algunos casos, hay puntos en la superficie de un objeto, los cuales son suficientemente brillantes para reflejar la luz, casi de la misma manera que un espejo. Tales puntos sólo pueden verse cuando son observados desde ángulos particulares, requiriendo múltiples imágenes de rango para verlos todos [12]. Múltiples vistas ruidosas de un punto en una superficie, pueden contribuir a una estimación de varianza reducida de la localización de ese punto.

La primera etapa en el proceso de reconstrucción, consiste en adquirir las imágenes de las escenas en cuestión mediante la utilización de cualquier sensor



de rango. A continuación se hace una breve descripción del resto de etapas del proceso de reconstrucción [12].

◆ **Registro e integración**

Dado que una simple vista de rango puede contener únicamente los puntos en la escena que son visibles al sensor, es necesario unir múltiples vistas de rango en un único conjunto de puntos, para obtener el modelo completo de la escena. El registro consiste en obtener una transformación (translación y rotación) que permita llevar un par de vistas de rango al mismo sistema de coordenadas. La integración es el proceso donde se remueve cualquier dato redundante, que sea visible en ambas vistas, o se promedian los puntos comunes.

◆ **Segmentación**

El proceso de segmentación consiste en agrupar puntos en una imagen de rango dentro de regiones con propiedades homogéneas. Dado que estas propiedades buscan definir conjuntos de puntos pertenecientes a la misma superficie, esto conlleva a que la segmentación y el ajuste de superficies estén muy relacionados.

◆ **Ajuste de superficies**

El objetivo del ajuste de superficies es dar el mejor estimativo de la verdadera superficie descrita por un conjunto de puntos segmentado. Incluye determinar la topología de la superficie y ajustar una superficie paramétrica a esta topología.

### 1.6.3 Aplicaciones

Las aplicaciones de la digitalización y reconstrucción de formas 3D son de gran alcance e incluyen manufactura, simulación virtual, exploración científica, medicina, negocios, entre otros [12].

### ◆ Inspección

Después que un constructor ha creado un modelo computarizado de una pieza, ya sea por medio de la digitalización de la forma de un modelo físico, o a través de un diseño CAD interactivo, éste posee una variedad de opciones para fabricar esta parte, como un prototipo funcional y como punto de partida para su producción en masa. Finalmente, las dimensiones de la parte fabricada deben caer dentro de algunas tolerancias de funcionamiento. En este caso, la digitalización de la figura puede ayudar a determinar dónde y en qué medida difieren el modelo computarizado y la forma de la parte real. Estas diferencias pueden servir como guía para modificar el proceso de manufactura hasta que la parte sea aceptable, ver Figura 14.

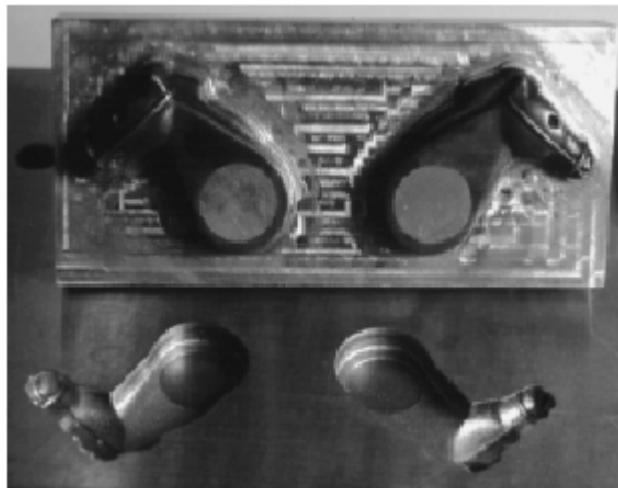


Figura 14. Molde de Fabricación [12]

### ◆ Efectos especiales, juegos, y mundos virtuales

Las imágenes sintéticas están jugando un papel cada vez más importante en la creación de efectos especiales para el cine. Además, los video-juegos y el hardware de juegos se están inclinando lentamente hacia los gráficos interactivos en tercera dimensión (3D). La realidad virtual como medio para experimentar mundos simulados también crece en popularidad. Todas estas aplicaciones requieren modelos en 3D que puedan ser tomados de la vida real o de esculturas creadas por artistas. La digitalización de formas de

modelos físicos será esencial para poblar estos ambientes sintéticos, ver figura 15.

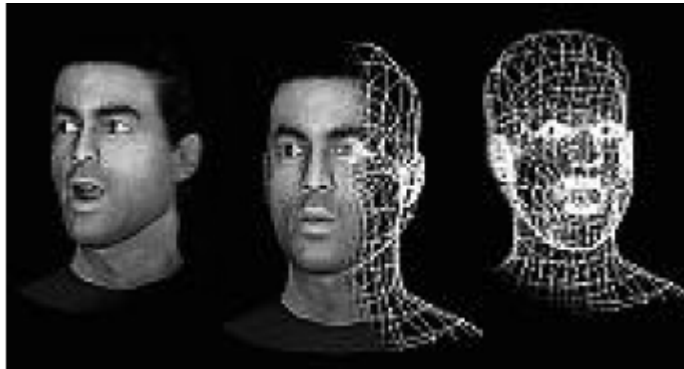


Figura 15. Animación por computador [11]

#### ◆ Obras de museo

Las obras de museos representan objetos únicos en su clase, que atraen el interés de científicos y público en general en todo el mundo. Tradicionalmente, para apreciar estos objetos, ha sido necesario visitar museos potencialmente distantes u obtener imágenes no interactivas o secuencias de video. Digitalizando estas piezas, los curadores de los museos pueden hacerlas disponibles para una visualización interactiva. Para los científicos, estos modelos computarizados ofrecen la oportunidad de estudiar y medir artefactos remotamente, usando poderosas herramientas de computación, como se muestra en la figura 16.



Figura 16. Piezas de Museo [13]



#### 1.6.4 Digitalización de Imágenes de Rango

Existen técnicas de visión 3D, que pueden ser clasificadas en métodos pasivos y activos. Los métodos pasivos se limitan al estudio de la información visual de una, dos o más vistas sobre la escena. La visión activa trata de imitar un sistema biológico, ya que se mejora el resultado de la percepción visual, empleando mecanismos que reaccionan y controlan los sensores visuales y factores externos como la iluminación, [11]. En el encuentro de la comunidad internacional de visión, en Chicago en junio de 1997, se definió la visión activa como “El control activo de todos los parámetros de la cámara” [11]. En general, las aplicaciones industriales que se realizan con visión 3D no pertenecen exclusivamente a uno de los tipos enumerados, sino que suelen ser combinaciones de dos o más de ellos. La clasificación en métodos activos y pasivos que se seguirá en esta investigación, corresponde a la que se presenta frecuentemente en la bibliografía de visión 3D, [10, 11, 12], y es mucho más sencilla, ya que no se tiene en cuenta si se modifican o no, los parámetros internos de la cámara o la posición, sino simplemente, si se modifica la iluminación. Así se referirán como técnicas de visión activa aquellas en las que se actúa con algún tipo de energía externa sobre el objeto a medir, con el fin de capturar de forma más sencilla sus dimensiones y las técnicas de visión pasiva serán aquéllas en las que no se emplea ninguna fuente de energía externa [11]. Los sensores de rango activo aprovechan una variedad de principios físicos, la figura 17 resume los métodos de adquisición de rango activo. Entre las técnicas de detección más comunes están la triangulación, radar/sonar, interferometría Moiré y enfoque/desenfoque activo. La triangulación usa un proyector de luz y una cámara de intensidad, que es colocada a cierta distancia del proyector. El proyector emite un patrón de luz. Los patrones más comunes son el plano y el haz de luz sencillo. El radar/sonar usa una onda electromagnética o acústica corta y detecta el retorno (eco) desde las superficies. La distancia es obtenida como una función del tiempo tomado entre el golpe de la onda con la superficie y su retorno. Los sensores Moiré proyectan dos rejillas con patrones espaciados regularmente sobre la superficie y miden las diferencias de fase del patrón de interferencia



observado. Otros sensores de diferencia de fase miden el cambio de fase del haz de retorno observado, en ellos la distancia es una función de diferencia de fase. Los sensores de enfoque/desenfoque activo usan dos o más imágenes de la misma escena, que son adquiridas bajo situaciones de enfoque variantes. Una vez el mejor enfoque de imagen es determinado, un modelo enlaza valores de enfoque y distancia, produciendo la profundidad. [14, 15].



Figura 17. Taxonomía de los métodos de adquisición de rango activo

## 1.7 Lenguajes de Programación.

Un lenguaje de programación es una serie de comandos que permiten codificar instrucciones de manera que sean entendidas y ejecutadas por una computadora. Un intérprete es aquel lenguaje que no trabaja en código máquina en forma directa, sino que va traduciendo cada instrucción. Obviamente son mucho más lentos que los lenguajes de alto nivel que trabajan ejecutando instrucciones directamente en código máquina. Un compilador permite traducir las instrucciones del lenguaje contenidas en el código fuente (instrucciones) a código máquina, de manera que el programa no necesita interpretar o convertir cada instrucción. Debido a esto es mucho más veloz que un intérprete y por supuesto mucho más profesional.

A partir de C y Pascal; los lenguajes se dividen en:



- Estructurados: son aquellos que en su codificación usaban una estructura jerárquica de procedimientos y funciones.
- No estructurados: como el Basic cuya codificación se basaba en líneas de programación, permitiendo al programador "saltar" de una línea de instrucción a otra, haciendo que el código fuera algunas veces no entendible y muy difícil de mantener (modificar) porque no seguía una estructura.

Basic evolucionó, primero con el ahora primitivo GW Basic, teniendo su máxima expresión con el Quick Basic del D.O.S. 5.0, el cual ya incluía algunos conceptos más de avanzada y más "aggiornados" a lo que eran sus contrapartes estructuradas. DBase fue el gran desarrollo para base de datos de los años 80. Bajo la firma Ashton Tate, empresa que dio origen a un interprete de bases de datos muy sencillo y poderoso: dBase II. Luego vinieron el dBase III+ que hizo furor, y la etapa de la decadencia para dBase: el dBase IV, ya bajo la dirección de Borland. Así surgió el gran compilador Clipper, de Nantucket Corp., en su versión Autumm 86 que permitía generar ejecutables libres de royalties y sin runtimes, incluso tomando en forma directa el código dBase.

A principios de la década pasada evolucionan los lenguajes de programación, en forma profesional y aplicaciones de alto nivel, el lenguaje esencial era C. Para el aprendizaje se usaba Pascal, que permitía inculcar el concepto de programación estructurada. Basic, era un lenguaje utilizado, no en pocas ocasiones en forma profesional, aunque con ciertas limitaciones; su reinado estuvo en los años 80.

A medida que los años van pasando el concepto de Bases relacionales empieza a decaer relativamente, surge entonces una variante que se aplica a todos los lenguajes: la orientación a objetos. Ya no sólo se habla de programación estructurada, sino que los módulos de programación son vistos como objetos, las



estructuras representan objetos y/o funciones que se adaptan en forma general a procesos específicos es la maximización de la programación modular. El modelo de objetos engloba los conceptos de encapsulación, herencia y poliformismo, el cual se aplica a los datos y al tipo de bases de datos que almacena la información. La orientación a objetos significa la agrupación de entidades de datos de forma global, de tal manera que puedan ser interpretados de una forma común por una misma estructura de programación.

Windows 95 marca el comienzo del fin de la programación DOS y por lo tanto de los lenguajes basados en éste. Este proceso no fue enérgico, hasta este momento, estamos viviendo esta etapa. Todavía hay numerosos y excelentes sistemas desarrollados bajo entorno DOS ejecutándose pero cada vez son menos.

Veamos que hicieron los lenguajes DOS para mantenerse en el mercado:

- Clipper trato de evolucionar hacia Windows bajo Computer Associates con el nombre de Visual Objects, el cual fue un fracaso.
- Visual Dbase 5.0 bajo la órbita de Borland también fracaso.
- Los sobrevivientes al menos en esencia son Visual Fox (Microsoft), Visual Basic (Microsoft), Delphi (Borland) y Visual C (Microsoft).

Con al llegada de Windows todo es Visual, todo es iconos, todo es botones, todo es Ventanas. Para programar en lenguajes visuales, primero hay que comprender lo que es Windows. La forma de programar los sistemas evolucionó radicalmente. Con Windows es preciso programar conservando las convenciones del mismo, guardando sus características y funcionalidades. Los sistemas hechos para Windows, deben ser tan Windows como el propio sistema operativo. La forma de programar se basa en objetos, cada uno de los cuales tiene sus Propiedades y funciones. Se basa en la programación de eventos para dichos objetos. Otro detalle es que la programación se basa en componentes (OLE, OCX, ActiveX), los cuales reducen notablemente el trabajo de la programación al proporcionar



herramientas antes impensadas en la programación DOS. Todos los lenguajes visuales ofrecen RAD (Rapid Application Development) o Wizzards; con lo cual comenzaron a prometer hacer aplicaciones en poco tiempo, incluso para inexpertos. Lejos estaban ya los tiempos en que programar era sólo usar un simple editor de textos. Ahora las herramientas de programación son poderosas. Son casi un sistema operativo, por así decirlo; con entornos de desarrollos avanzados y excelentes Debuggers.

Los paradigmas de la programación Windows son:

- Borland Delphi (la evolución del Pascal)
- Visual Fox (la evolución del Xbase)
- Visual Basic (la evolución de Basic)
- Visual C (la evolución de C)

Las incursiones cada mas innovadoras de Microsoft parecen imponer a la web como el centro de desarrollo de aplicaciones: Microsoft .NET. Una visión a la WEB y al futuro: HTML, Perl, Pithon, Java y otros.

Internet ha sido el disparador de nuevos lenguajes tales como el HTML que es el lenguaje de programación de las páginas WEB para hipertexto. El mismo constituye una codificación bastante simple, basada en marcadores (TAGs).

De la misma manera, cuando se hace necesario proveer de funciones adicionales a un servicio web se recurre a Perl o a Pithon que son lenguajes que permiten escribir scripts para ser alojados en los servidores a efectos de proveer contadores, estadísticas, rankings, etc.

Por otra parte Java, bajo la dirección de SUN, constituye la idea de la programación abierta y universal para las aplicaciones de escritorio, pero todavía los estándares visuales (C, Basic y Delphi), son demasiado poderosos como para



desplazarlos, a pesar de que Java promete también ser un lenguaje de excelentes prestaciones.

Las nuevas tecnologías WEB inundan el mercado: PHP, ASP, XML, DHTML, lo cual enriquecen la forma de manejar la información y su presentación al usuario final.

Es de esperarse que los lenguajes Visuales dominen el mundo de las computadoras durante mucho tiempo, por lo menos, mientras no evolucionen de otra manera los sistemas operativos. Los lenguajes de programación evolucionan a medida que lo hacen los sistemas operativos, nunca un lenguaje de programación determinó un Sistema Operativo, por el contrario los Sistemas Operativos determinaron los lenguajes de programación.

### 1.7.1 Lenguaje de Programación Delphi

El lenguaje de programación Delphi se divide en tres secciones:

- Compilador (con su "encadenador"): es un programa que crea el archivo ejecutable de Windows PE estilo Intel, sin ningún intérprete de por medio.
- Librería: es código que permite usar todas las capacidades de Delphi. Éstas están escrita en su totalidad en Object Pascal la cual es una librería de clases estilo MFC, llamada VCL, y está totalmente orientada a objetos.
- IDE (Ambiente de desarrollo integrado, o Integrated Development Environment): éste está realizado en Delphi, y utiliza las mismas librerías que se utilizan para compilar.

El programa principal de Delphi es un archivo de texto ASCII con extensión .DPR. Esta extensión quiere decir Delphi PProject (proyecto de Delphi).

Para cada una de las ventanas que se diseña en el IDE, Delphi crea una "unidad". Una unidad es un archivo individual (de texto ASCII) que representa en



general a un objeto, o a una agrupación lógica de funciones. En el caso de los "objetos" que son formas, Delphi también crea un archivo "DFM" (Delphi Form) para guardar la apariencia del diseño de las mismas.

Una vez que Delphi ha especificado todo lo que va a usar el proyecto, el programa inicializa la aplicación (Application.Initialize), crea la forma Form1 a partir de la definición de objeto TForm1, usando CreateForm, y la aplicación corre (Application.Run). Cualquier forma creada con CreateForm está activa (aunque podría ser invisible) hasta que el Application Run termina. Después de esto (normalmente cuando Run recibe el mensaje Terminate) el programa termina.

Run en Delphi es un método de la aplicación que hace al programa entrar en el ciclo de mensajes de Windows. El ciclo de mensajes es un ciclo estilo *Do While* que recibe mensajes de Windows (clicks de botones, teclados, movimientos del mouse) y procesa los mensajes en una manera consistente para todas las ventanas de la aplicación.

## 1.8 Imágenes Digitales.

Las imágenes digitales son fotos electrónicas tomadas de una escena, escaneadas de documentos como: fotografías, manuscritos, textos impresos e ilustraciones obtenidas por dispositivos que utilizan métodos ópticos. Se realiza una muestra de la imagen digital y se confecciona un mapa de ella en forma de cuadrícula de puntos o elementos de la figura (píxeles) [10]. A cada píxel se le asigna un valor tonal (negro, blanco, matices de gris o color), el cual está representado en un código binario. Los dígitos binarios (bits) para cada píxel son almacenados por una computadora en una secuencia, y con frecuencia se los reduce a una representación matemática (comprimida). Luego la computadora interpreta y lee los bits para producir una versión analógica para su visualización o impresión.



La resolución es la capacidad de distinguir los detalles espaciales finos. Por lo general, la frecuencia espacial a la cual se realiza la muestra de una imagen digital (la frecuencia de muestreo) es un buen indicador de la resolución. Este es el motivo por el cual dots-per-inch (puntos por pulgada) (dpi) o pixels-per-inch (píxeles por pulgada) (ppi) son términos comunes y sinónimos utilizados para expresar la resolución de imágenes digitales. Generalmente, pero dentro de ciertos límites, el aumento de la frecuencia de muestreo también ayuda a aumentar la resolución.

Los píxeles pueden verse en forma individual al aumentar una imagen por medio del zoom. Las dimensiones de píxel son las medidas horizontales y verticales de una imagen, expresadas en píxeles. Las dimensiones de píxel se pueden determinar multiplicando tanto el ancho como la altura por el dpi.

La profundidad de bits es determinada por la cantidad de bits utilizados para definir cada píxel. Cuanto mayor sea la profundidad de bits, tanto mayor será la cantidad de tonos (escala de grises o color) que puedan ser representados. Las imágenes digitales se pueden producir en blanco y negro (en forma bitonal), a escala de grises o a color.

Ahora observemos los cálculos binarios para la cantidad de tonos representados por profundidades de bits comunes:

- 1 bit  $(2^1) = 2$  tonos
- 2 bits  $(2^2) = 4$  tonos
- 3 bits  $(2^3) = 8$  tonos
- 4 bits  $(2^4) = 16$  tonos
- 8 bits  $(2^8) = 256$  tonos
- 16 bits  $(2^{16}) = 65.536$  tonos



24 bits ( $2^{24}$ ) = 16,7 millones de tonos

El rango dinámico es la diferencia tonal entre la parte más clara y la más oscura de una imagen. Cuanto más alto sea el rango dinámico, se pueden potencialmente representar más matices, a pesar de que el rango dinámico no se correlaciona en forma automática con la cantidad de tonos reproducidos. El rango dinámico también describe la capacidad de un sistema digital de reproducir información tonal. Esta capacidad es más importante en los documentos de tono continuo que exhiben tonos que varían ligeramente, y en el caso de las fotografías puede ser el aspecto más importante de la calidad de imagen.

Los formatos de archivo consisten tanto en los bits que comprende la imagen como en la información del encabezamiento acerca de cómo leer e interpretar el archivo. Los formatos de archivo varían en términos de resolución, profundidad de bits, capacidades de color, y soporte para compresión. En la tabla 1 observamos los diferentes formatos de las imágenes.

Nombre y versión actual	TIFF 6.0 (Tagged Image File Format)	GIF 89a (Graphics Interchange Format)	JPEG (Joint Photographic Expert Group)/JFIF (JPEG File Interchange Format)	Flashpix 1.0.2	ImagePac, Photo CD	PNG 1.2 (Portable Network Graphics)	PDF 1.3 (Portable Document Format)
<b>Extensión (Extensiones)</b>	.tif, .tiff	.gif	.jpeg, .jpg, .jif, .jfif	.fpx	.pcd	.png	.pdf
<b>Profundidad (es) de bits</b>	Bitonal a 1 bit; escala de grises o color de paleta de 4 u 8 bits; hasta color de 64 bits(1)	Bitonal, escala de grises o color entre 1 y 8 bits	Escala de grises a 8 bits; color a 24 bits	Escala de grises a 8 bits; color a 24 bits	Color a 24 bits	1-48 bits; color a 8 bits, escala de grises a 16 bits, color a 48 bits	Escala de grises a 4 bits; color a 8 bits; soporta hasta 64 bits para color
<b>Compresión</b>	Descomprimido sin pérdida: ITU-T.6, LZW, etc.  Con pérdida: JPEG	Sin pérdida: LZW(2)	Con pérdida: JPEG  Sin pérdida:(3)	Descomprimido  Con pérdida: JPEG	Con pérdida: Formato patentado Kodak "sin pérdida visual"(4)	Sin pérdida: Deflate, derivado de LZ77	Descomprimido Sin pérdida: ITU-T.6, LZW  Con pérdida: JPEG
<b>Estándar / patentado</b>	Estándar de facto	Estándar de facto	JPEG: ISO 10918-1/2	Especificación disponible para el público	Patentado	ISO 15948 (anticipado)(6)	Estándar de facto(7)



			JFIF: estándar de facto(5)				
<b>Gestión de color</b>	RGB, Paleta, YCbCr,(8) CMYK, CIE	Paleta	YCbCr	PhotoYCC y NIF RGB,(9) ICC (opcional)	PhotoYCC	Paleta, sRGB, ICC	RGB, YCbCr, CMYK
<b>Soporte de Web</b>	Conexión o aplicación externa	Originario desde Microsoft® Internet Explorer 3, Netscape Navigator® 2	Originario desde Microsoft® Internet Explorer 2, Netscape Navigator® 2	Conexión	Aplicación Java™ o aplicación externa (10)	Originario desde Microsoft® Internet Explorer 4, Netscape® Navigator 4.04, (pero aún incompleto)	Conexión o aplicación externa
<b>Soporte de metadatos</b>	Conjunto básico de rótulos etiquetados	Campo de texto libre para comentarios	Campo de texto libre para comentarios	Gran conjunto de rótulos etiquetados	A través de bases de datos externas; no posee metadatos inherentes	Conjunto básico de rótulos etiquetados más rótulos definidos por el usuario.	Conjunto básico de rótulos etiquetados
<b>Comentarios</b>	Acepta imágenes y archivos múltiples(11)	Se puede reemplazar por PNG;  Soporte de entrelazado y transparencia a través de la mayoría de los navegadores Web	JPEG progresivo ampliamente soportado por los navegadores Web(12)	Proporciona múltiples resoluciones de cada imagen; amplio soporte de la industria, pero aplicaciones actuales limitadas	Proporciona 5 ó 6 resoluciones diferentes de cada imagen; futuro incierto	Puede reemplazar a GIF	Preferido para imprimir y ver documentos de páginas múltiples; uso intensivo por parte del gobierno
<b>Página inicial</b>	Página inicial no oficial de TIFF	Especificación GIF	Página inicial JPEG	Página inicial FlashPix	Página inicial PhotoCD	Página inicial PNG	Página inicial PDF

Tabla 1. Diferentes tipos de formato de imágenes.

## 1.9 Gráficos

El término Gráficos 3D por Computadora (3D Computer Graphics) se refiere a trabajos de arte gráfico que fueron creados con ayuda de computadoras y Software especial 3D. En general, el término puede referirse también al proceso de crear dichos gráficos, o el campo de estudio de técnicas y tecnología relacionada con los gráficos 3D .



Los gráficos 3D difieren de los Gráficos 2D en que es guardado en la computadora una representación virtual en tres dimensiones de los objetos, con el propósito de realizar cálculos y generar imágenes.

OpenGL y Direct3D son dos APIs (Application Programming Interface – Interfase de Programación de Aplicaciones) para la generación de imágenes 3D en tiempo real. Muchas placas de vídeo modernas proveen algún grado de aceleración por hardware basado en estas APIs, frecuentemente habilitando el despliegue de complejos gráficos tridimensionales en tiempo real.

El proceso de creación de gráficos 3D por computadora puede ser dividido en estas tres fases básicas:

- Modelado.- La etapa de modelado consta de ir dando forma a objetos individuales que luego serán usados en la escena. Existen diversas técnicas de modelado; Constructive Solid Geometry, modelado con NURBS y modelado poligonal son algunos ejemplos. Los procesos de modelado puede incluir la edición de la superficie del objeto o las propiedades del material (color, luminosidad, difusión, especularidad, características de reflexión, transparencia u opacidad, o el índice de refracción), agregar texturas, mapas de relieve (bump-maps). Este proceso puede incluir algunas actividades relacionadas con la preparación del modelo 3D para su posterior animación. A los objetos se les puede asignar un esqueleto, una estructura central con la capacidad de afectar la forma y movimientos de ese objeto. Esto ayuda al proceso de animación, en el cual el movimiento del esqueleto automáticamente afectará las porciones correspondientes del modelo. El modelado puede ser realizado por programas dedicados (Lightwave, Rhinoceros 3D, Moray, un componente de una aplicación como Shaper, Lofter en 3D Studio o por un lenguaje de descripción de escenas como en POV-Ray. En algunos casos, no hay una distinción estricta entre estas fases; en dichos casos, el



modelado es solo una parte del proceso de creación de escenas, por ejemplo, con Caligari trueSpace.

- Composición de la escena.- Esta etapa involucra la distribución de objetos, luces, cámaras y otras entidades en una escena que será utilizada para producir una imagen estática o una animación. Si se utiliza para animación, esta fase, en general, hace uso de una técnica llamada Keyframing, que facilita la creación de movimientos complicados en la escena. Con la ayuda de esta técnica en lugar de tener que corregir la posición de un objeto, su rotación o tamaño en cada cuadro de la animación sólo se necesita marcar algunos cuadros clave (keyframes). Los cuadros entre keyframes son generados automáticamente, lo que se conoce como Interpolación.

La iluminación es un aspecto importante de la composición de la escena que contribuye al resultado estético y a la calidad visual del trabajo terminado.

El proceso de transformar la representación de objetos, como el punto medio de coordenadas de una esfera y un punto en su circunferencia, en una representación poligonal de una esfera, se conoce como tessellation. Este paso es usado en el rénder basado en polígonos, donde los objetos son descompuestos de representaciones abstractas primitivas como esferas, conos, etc., en las denominadas mallas, que son redes de triángulos interconectados. Las mallas de triángulos son populares ya que está probado que son fáciles de 'renderizar' usando Scanline rendering.

- Renderizado.- Se llama al proceso final de generar la imagen 2D o animación a partir de la escena creada. Esto puede ser comparado con tomar una foto o filmar la escena en la vida real, después que se terminó de armar. Generalmente se buscan imágenes de calidad fotorrealista, y para este fin se han desarrollado muchos métodos especiales. Las técnicas van desde el renderizado de alambre (wireframe rendering), pasando por el



renderizado basado en polígonos, hasta las técnicas más modernas como: Scanline Rendering, Raytracing o Radiosity.

El Software de renderizado puede simular efectos cinematográficos como lens flare, profundidad de campo o motion blur (desenfoque de movimiento). Estos artefactos son, en realidad, un producto de las imperfecciones mecánicas de la fotografía física, pero como el ojo humano está acostumbrado a su presencia, la simulación de dichos efectos aporta un elemento de realismo a la escena. Se han desarrollado técnicas con el propósito de simular otros efectos de origen natural, como la interacción de la luz con la atmósfera o el humo.

Este proceso también requiere gran procesamiento, ya que requiere simular gran cantidad de procesos físicos complejos. La capacidad de procesamiento se ha incrementado rápidamente a través de los años, permitiendo un grado superior de realismo en los renderizados. Los estudios de cine que producen animaciones generadas por computadora hacen uso, en general, de lo que se conoce como Render farm (granja de rúnder) para generar imágenes de manera más rápida.

Los gráficos 3D por computadora modernos cuentan con un modelo de reflexión llamado Phong reflection model, que no debe ser confundido con Phong shading, que es algo completamente diferente.

Este modelo de reflexión y las técnicas de sombreado que permite, se aplican solo a rúnders basados en polígonos, por ejemplo, raytracing y radiosity no lo utilizan.

Las técnicas de reenderezado de reflexión populares son:



- Flat shading: una técnica que sombrea cada polígono de un objeto basado en la normal del polígono y la posición e intensidad de una fuente de luz.
- Gouraud shading: es una rápida técnica de sombreado de vértices usada para simular superficies suavemente sombreadas.
- Texture mapping: Es una técnica para simular un gran nivel de detalle superficial, aplicando imágenes (texturas) sobre los polígonos.
- Phong shading: es utilizado para simular brillos especulares y superficies sombreadas suaves.
- Bump mapping: es una técnica de perturbación utilizada para simular superficies rugosas.

### **1.10.1 APIs de Gráficos 3D**

Los gráficos 3D se han convertido en algo muy popular, particularmente en juegos de computadora, al punto que se han creado APIs especializadas para facilitar los procesos en todas las etapas de la generación de gráficos por computadora. Estas APIs han demostrado ser vitales para los desarrolladores de Hardware para gráficos por computadora, ya que proveen un camino al programador para acceder al hardware de manera abstracta, aprovechando las ventajas de la tarjeta de video.

Las siguientes APIs para gráficos por computadora son particularmente mas utilizadas:

- OpenGL
- Direct3D (un derivado de DirectX)
- RenderMan



A pesar de haber muchos paquetes de modelado y animación 3D, los cuatro que se han ganado la mayor popularidad son:

- ❖ Alias Wavefront's Maya - es utilizado por muchos de los estudios de efectos visuales más importantes en combinación con Renderman, el motor de render fotorrealista de Pixar.
- ❖ Discreet's 3D Studio Max - es el líder en el desarrollo de 3D en la industria de juegos y usuarios hogareños.
- ❖ Newtek's Lightwave 3D - es el favorito entre los entusiastas, y es utilizado en muchas de las mayores productoras de efectos visuales como Digital Domain.

Hay otros que no se han ganado tal aceptación general, pero que no son simples juguetes. Algunos son:

- ❖ Caligari trueSpace - una aplicación 3D integrada, con una interfase muy intuitiva. Una característica distintiva de esta aplicación es que todas las fases de creación de gráficos 3D son realizadas dentro de un único programa. No es tan avanzado como los paquetes líderes, pero provee características como simulación de fenómenos físicos (viento, gravedad, colisiones entre cuerpos).
- ❖ Cinema4d - Motor de rénder rápido, cálculo de radiosidad.

OpenGL es un estándar sobre gráficos de computadora. Desde el punto de vista del programador es un API para interactuar con dispositivos gráficos, esto ofrece: portabilidad, primitivas geométricas básicas, transformaciones de vista, modelado, tratamiento del color, eliminación de superficies ocultas, tratamiento de las luces.

Existen distintas implementaciones como:

- Microsoft OpenGL
- SGI OpenGL para Windows
- OpenGL ports y miniports específicos



- 3DFXGL
- MESA

Y tiene librerías auxiliares como:

- GLU: primitivas 3D, operaciones con matrices...
- GLUT: tratamiento de ventanas, ratón, teclado...
- GLAUX: igual que GLUT, pero sólo para Windows
- GLX: para un sistema X-Windows de Linux
- AGL, PGL, WGL: como GLX para Apple Macintosh,
- IBM OS/2 Warp y Windows NT y 95
- OpenAL, OpenNL, OpenIN: para audio, red o entrada respectivamente.

Para trabajar con estas librerías se requiere tener un entorno de desarrollo como Visual C++, C++ Builder ó Delphi.

Para instalar estas librerías se sigue los siguientes pasos:

- ❖ Headers (glut.h, glu.h, gl.h, glaux.h) en:  
C:\Archivos de programa\Microsoft Visual Studio\VC98\Include\GL
- ❖ DLLs (glut.dll, glu.dll, glut32.dll, glu32.dll) en:  
C:\Windows\System y System32
- ❖ LIBs (glut.lib, glu32.lib, glut32.lib, glaux.lib) en:  
C:\Archivos de programa\Microsoft Visual Studio\VC98\Lib
- ❖ Programación en Windows

La función main desaparece y se sustituye por:

```
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE
hPrevInstance, LPSTR lpszCmdLine, Int nCmdShow);
```

La programación pasa a ser orientada a eventos



Todo lo anterior muestra las diferentes áreas de las ciencias así como sus herramientas para lograr nuestros objetivos. Éstas se observan de la siguiente manera:

- Definiciones Estomatológicas: estructura de los dientes, enfermedades de las piezas dentales, métodos de diagnóstico de defectos o enfermedades.
- Áreas de Computación: imágenes digitales, visión computacional, graficación, animación por computadora.
- Áreas de Matemáticas: geometría, topologías, análisis numérico.
- Áreas de Física: óptica.
- Área de Electrónica: microcontroladores, circuitos electrónicos, control
- Herramientas de Computación: lenguajes de programación, lenguaje Delphi, APIS para programación en 3D como OpenGL.

# CAPÍTULO 2

## **PLANTEAMIENTO DEL PROBLEMA**



En el área Estomatológica los métodos para hacer un diagnóstico en las piezas dentales son invasivos, es decir, provocan cambios en la estructura del esmalte dental ocasionado enfermedades posteriores, a tal grado de perder los órganos dentarios. Los métodos tradicionales como: rayos X, explorador, transiluminación; han evolucionado conforme avanza la tecnología y han surgido nuevos métodos como: rayos X con conexión a la computadora, cámaras intraorales, láser dental; aun así, estos métodos siguen siendo invasivos. Los métodos usuales no detectan defectos menores a 150 micras.

Por este motivo es necesario implementar nuevas metodologías que detecten defectos menores a 150 micras y que sean no invasivos.

En el Laboratorio de Métodos no Invasivos para el Diagnostico Estomatológico de la Facultad de Estomatología de la Benemérita Universidad Autónoma de Puebla, se está desarrollando tecnología que permita:

- No deteriorar la pieza dental al hacer el análisis.
- Hacer un análisis cada 50 micras.
- Encontrar posibles defectos que presente la pieza dental.
- Sea ecológicamente limpia.

En este proyecto de tesis se plantea una metodología para analizar la rugosidad del esmalte dental, haciendo una reconstrucción en tres dimensiones de las piezas dental, mostrando la rugosidad de las 5 caras del diente así como en volumen, y con esto se podrá observar los defectos del esmalte dental.

Esta tecnología se basa en el método de reflectometría, el cual hace incidir un rayo de luz sobre una superficie y el reflejo, este reflejo muestra el comportamiento de la rugosidad de la superficie y si existe algún defecto éste se observa en los resultados de recepción, esto es factible en piezas planas ya que siempre se conserva la misma distancia entre el tracto Emisor Recepto (E-R),



Para realizar este método en el análisis de las piezas dentales se implementará un algoritmo para la posición de los ángulos tanto del emisor y receptor así como la distancia de pieza dental, este algoritmo consistirá en hacer movimientos de los ángulos tanto del emisor y receptor, el valor mas alto de los resultados dará los ángulos óptimos para el escaneo de la superficie.

Para implementar este método, se diseñará un mecanismo automático que permitirá hacer el barrido de toda la pieza dental, en donde el tracto E-R se coloca sobre los motores que representan el plano cartesiano X y Y. El tracto E-R está compuesto por un rayo láser como emisor y un fotodiodo como receptor. Para almacenar los datos del reflejo se necesita una tarjeta de adquisición de datos para tener una comunicación entre el dispositivo optoelectrónico y la computadora, esta tarjeta se basa en la tecnología de microcontroladores PIC's y la comunicación para la transmisión de datos se realiza mediante el puerto serial siguiendo el protocolo RS-232, y el movimiento de los motores para el escaneo se realiza mediante instrucciones que se envían por el puerto paralelo.

Los datos adquiridos se almacenarán en un archivo, el cual servirá para tratarlo posteriormente, este almacenamiento se realizará de forma matricial ya que una matriz representa las coordenadas e intensidad de una imagen. Estos datos vendrán con ruido generado por los dispositivos electrónicos, a estos datos se les aplicará un algoritmo que permita observar las frecuencias tanto pasa baja como pasa alta, para encontrar bordes o anomalías que estén dentro de la imagen y con esto poder eliminar el ruido que pueda tener.

A estos datos sin ruido, se le aplicará algún método de interpolación para encontrar un punto entre dos puntos con respecto a la distancia de estos, y así dar mas forma a la topología de la imagen, es decir si se toma lecturas cada 50 micras la primera interpolación dará un valor que represente la posición de 25 micras, si se realiza otra interpolación entonces se tendrá el punto de 12.5 micras, esto



hasta tener una imagen con distancia hasta de 6.25 micras, lo cual originará una matriz aumentada y una mejor definición de los posibles defectos que tenga la pieza dental.

Luego se tendría el tratamiento de datos para generar una matriz con datos confiables y precisos para realizar la imagen en tres dimensiones. La reconstrucción de las piezas dentales en 3 dimensiones se generará utilizando métricas topológicas regenerando la topografía de las 5 caras, permitiendo observar la forma de la estructura, el comportamiento de la rugosidad, posibles defectos, características propias de las piezas dentales.

A estas imágenes en 3 dimensiones se les aplicará una técnica de visión computacional para segmentar los puntos y así poder generar el volumen de las piezas dentales utilizando el espacio de tres dimensiones (X, Y, Z). A esta nueva imagen se le aplicará algoritmos de graficación para tener rotación traslación, escalamiento, así como luces, texturas; para que el escenario tenga una mejor vista y pueda tener movimiento de acuerdo a lo que se requiera, ya que así el especialista podrá observar mejor los detalles de las piezas dentales en tres dimensiones.

# CAPÍTULO 3

**SISTEMA PARA EL ANÁLISIS DE LA  
RUGOSIDAD DEL ESMALTE DENTAL**



En la figura 18 se observa el diagrama de bloques en la cual se muestra la metodología para el análisis de la rugosidad de la superficie del esmalte dental que consiste en: un mecanismo de adquisición de datos, la forma de adquirir los datos, eliminación de ruido, interpolación y por último la reconstrucción en 3 dimensiones.



Figura 18. Diagrama de bloques del sistema completo

### 3.1 *Dispositivo Mecánico*

El mecanismo automático se diseñó para hacer un barrido en los ejes X y Y, de tal forma que pueda cubrir toda la pieza dental y pueda adquirir datos a menos de 150 micras [8]. Este mecanismo consta de dos motores que se colocan conforme al plano cartesiano sosteniendo al tracto emisor receptor. El tracto emisor-receptor se compone de un rayo láser como emisor [16] y un fotodiodo como receptor, estos elementos se colocan en un motor a pasos cada uno, con la finalidad de determinar el ángulo óptimo tanto del emisor como del receptor ya que la superficie de las piezas dentales es no isotrópicas. El procedimiento del mecanismo es el siguiente:

- se colocan los motores del plano artesiano fuera de los bordes de las piezas dentales.
- se programa la distancia del movimiento de los motores.

- cuando el mecanismo está en una posición que genere un dato diferente de 0 entonces se activan los segundos motores tanto del emisor como del receptor. Primero se activa el motor del emisor, se coloca en un ángulo de  $20^\circ$  y después el motor del receptor se mueve de forma consecutiva a partir de los  $20^\circ$  hasta  $42^\circ$ , una vez terminado se mueve un grado el motor del emisor y el motor del receptor se coloca en la posición inicial y empieza a moverse de manera consecutiva, esto se realiza hasta que el motor del emisor cubre hasta los  $42^\circ$  en un solo punto. Estos datos se almacenan en un archivo y se agarran los datos mas altos y con mayor frecuencia, a éstos se les saca la moda y con los valores mas altos obtenemos el ángulo del emisor y del receptor. Se posicionan los motores en los ángulos óptimos, se adquiere el dato y se almacena. Y ahora se recorre el eje X o Y según sea y se vuelve a repetir lo anterior, todo esto se realiza hasta obtener todos los datos de la pieza dental.

En la figura 19 se observa el mecanismo automático y en la figura 20 se muestra el tracto emisor-receptor.

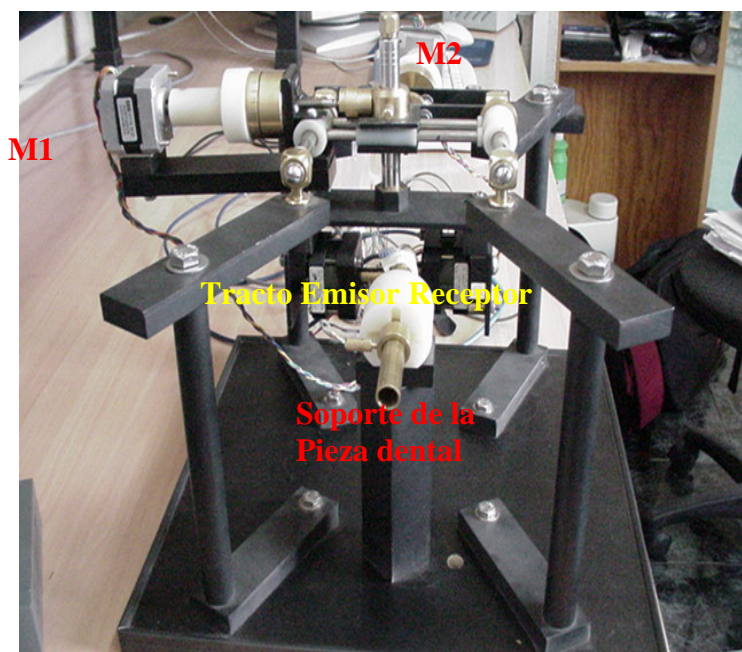


Figura 19. Mecanismo automático para la adquisición de datos

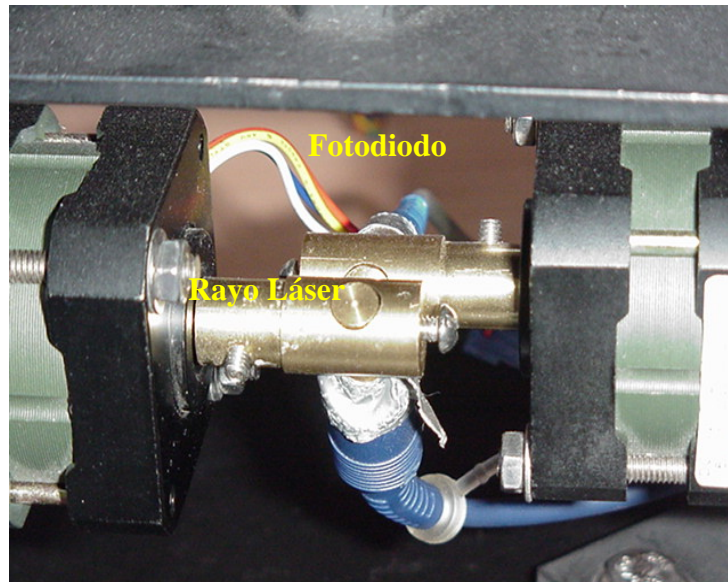


Figura 20. Tracto Emisor-Receptor

### 3.2 Adquisición de Datos

Adquirir los datos es fundamental para este trabajo, en la figura 21 se muestra el diagrama de bloques de la adquisición, éste consiste en un dispositivo optoelectrónico en el cual el receptor adquiere el dato milivoltios y se amplifica para tener el dato en voltios, este dato es adquirido por una tarjeta de adquisición de datos que se basa en microcontroladores PICS, esta tarjeta envía al convertidor analógico digital para convertir el dato con voltaje analógico a voltaje digital, luego la tarjeta trata a este dato para hacer una conversión a decimal y el resultado lo manda a la computadora utilizando el protocolo RS-232, estos datos recibidos por la computadora se almacenan en un archivo de tipo texto.

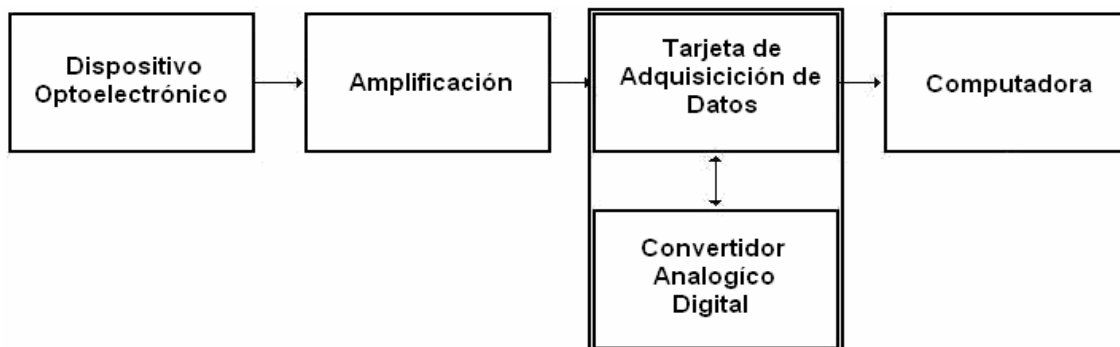


Figura 21. Diagrama de la adquisición de datos



El archivo con los datos adquiridos, se llena de la siguiente manera:

- Se coloca como etiquetas los números enteros 10, 20, 30, 40, 50 para la diferencia de las caras de las piezas dentales, es decir los datos reales que siguen delante de las etiquetas son las matrices de las caras de las piezas dentales.
  - 10 = Cara Vestibular
  - 20 = Cara Lingual o Palatina
  - 30 = Cara Distal
  - 40 = Cara Mesial
  - 50 = Cara Oclusal
- Los números enteros 60, 70, 80, 90, son etiquetas para la parte del paciente es decir:
  - 60 es el tipo de diente
    - 60 + 1001 = molar
    - 60 + 1002 = premolar
    - 60 + 1003 = diente Temporal o de leche
    - 60 + 1004 = diente Permanente
    - 60 + 1005 = canino
  - 70 es el estado del diente
    - 70 + 1011 = diente sano
    - 70 + 1012 = diente con caries
    - 70 + 1013 = diente figurado
    - 70 + 1014 = diente canino
  - 80 sexo del paciente
    - 80 + 1021 = el paciente es de sexo femenino
    - 80 + 1022 = el paciente es de sexo masculino
  - 90 es la edad del paciente.

En la Figura 22 observamos un ejemplo de esto.



10	1.386701	2.480437	2.402313	2.050755	2.011693	1.093736	1.074205	0.800771	2.499968	2.070286	1.953100	1.503887	1.191391
	2.070286	1.640604	2.070286	0.917957	1.132798	1.328108	1.328108	2.070286	1.230453	2.109348	1.445294	2.070286	2.480437
	2.246065	1.367170	2.402313	2.460906	1.699197	2.070286	1.249984	2.402313	2.324189	1.132798	2.480437	1.933569	2.187472
	2.324189	2.324189	1.816383	2.207003	0.820302	1.289046	1.582011	2.109348	2.460906	2.499968	2.363251	1.699197	1.718728
	2.402313	2.441375	1.289046	1.425763	2.460906	1.347639	2.499968	1.640604	2.304658	1.445294	2.460906	1.992162	2.558561
	1.484356	1.523418	1.562480	1.562480	2.363251	2.343720	1.542949	1.542949	1.757790	2.285127	1.562480	1.738259	1.816383
	1.699197	2.460906	2.519499	2.382782	1.582011	1.386701	2.441375	2.441375	2.499968	2.421844	1.796852	1.308577	1.230453
	2.441375	2.382782	0.839833	1.542949	0.761709	2.324189	2.441375	2.402313	1.777321	2.539030	2.343720	1.386701	2.460906
	2.187472	1.542949	2.460906	1.777321	1.542949	2.558561	2.480437	2.480437	1.738259	2.480437	1.757790	2.109348	2.480437
	2.324189	1.542949	2.519499										
20	2.187472	1.601542	2.128879	1.679666	1.874976	1.777321	1.874976	2.363251	2.519499	2.460906	1.972631	2.167941	1.699197
	2.558561	2.636685	1.874976	2.539030	2.578092	2.558561	1.992162	2.187472	1.933569	1.953100	2.167941	2.324189	2.148410
	1.562480	2.187472	2.402313	1.582011	2.480437	2.324189	2.148410	1.386701	2.421844	2.402313	2.558561	2.402313	0.976550
	1.874976	0.761709	2.304658	2.304658	2.324189	0.859364	2.167941	1.132798	1.718728	1.718728	2.421844	1.503887	2.050755
	2.382782	1.425763	1.718728	1.542949	1.796852	2.441375	2.382782	2.187472	1.640604	2.480437	1.503887	2.402313	2.421844
	2.187472	1.914038	1.542949	1.464825	1.835914	2.363251	1.484356	1.523418	1.855445	2.480437	1.601542		
30	2.011693	2.089817	2.539030	2.499968	2.363251	1.933569	1.855445	2.285127	1.933569	2.031224	2.578092	2.324189	2.246065
	1.953100	2.226534	2.558561	2.519499	2.539030	2.578092	2.480437	1.953100	1.933569	2.480437	1.660135	2.460906	2.402313
	2.539030	2.480437	2.441375	2.460906	2.402313	1.640604	2.539030	2.011693	1.269515	2.089817	2.421844	1.249984	0.878895
	1.953100	2.011693	1.972631	2.363251	2.343720	2.031224	1.562480	2.441375	2.421844	1.796852	1.503887	1.699197	2.226534
	2.187472	2.499968	1.816383	1.796852	1.816383	2.167941	2.109348	1.855445	2.324189	2.324189	2.402313	1.582011	2.480437
	1.777321	2.480437	2.109348	1.835914	1.855445	1.621073	1.835914	2.363251	2.382782	2.265596	2.324189	1.874976	2.441375
	2.265596	2.031224	2.363251	1.718728	2.558561	1.972631	2.265596	2.519499	1.914038	1.933569	2.031224	2.558561	2.578092
	2.011693	2.578092	2.441375	2.304658	2.421844	2.402313	2.285127	2.402313	1.855445	2.343720	2.421844	1.933569	2.539030
	2.519499	2.285127	2.421844	2.499968	1.796852	1.699197	1.699197	2.402313	2.480437	2.363251	1.601542	2.421844	2.050755
	2.031224	2.382782	0.937488										
40	1.972631	2.031224	2.343720	1.367170	2.539030	1.386701	2.402313	2.480437	2.480437	2.246065	2.382782	2.460906	2.207003
	1.562480	1.816383	2.480437	2.441375	2.285127	1.679666	2.265596	1.835914	2.148410	2.480437	2.382782	1.992162	2.499968
	1.914038	2.363251	2.304658	2.187472	2.519499	2.167941	1.874976	1.914038	2.460906	2.246065	2.324189	2.441375	2.226534
	1.757790	2.070286	1.874976	2.519499	1.914038	2.441375	1.874976	1.992162	2.207003	2.441375	2.402313	2.050755	2.050755
	2.011693	2.421844	2.148410	1.992162	1.874976	2.519499	2.343720	2.265596	2.285127	2.382782	2.148410	2.480437	1.874976
	1.562480	2.187472	2.460906	1.953100	2.402313	2.421844	2.421844	2.460906	1.601542	2.011693	2.382782	2.207003	1.523418
	2.011693	1.386701	1.855445	2.421844	1.562480	1.484356	2.265596	1.601542	2.128879	2.519499	1.816383	1.953100	2.382782
	2.441375	1.757790	1.933569	1.933569	2.460906	2.207003	2.480437	2.226534					
50	1.718728	1.855445	2.539030	2.343720	1.914038	2.519499	2.519499	1.777321	1.914038	2.519499	2.343720	499968	2.539030
	2.324189	2.050755	2.519499	2.480437	2.519499	1.972631	2.539030	2.558561	2.519499	2.480437	2.578092	2.499968	1.894507
	2.089817	2.578092	1.972631	1.992162	2.089817	2.558561	2.460906	2.558561	2.558561	2.167941	2.031224	1.835914	1.992162
	2.050755	2.578092	2.324189	2.480437	2.578092	1.718728	1.835914	2.128879	2.441375	2.089817	2.558561	1.796852	1.484356
	1.738259	2.148410	1.699197	1.249984	2.499968	2.070286	2.363251	1.249984	2.421844	1.855445	2.441375	1.601542	2.499968
	2.343720	1.484356	2.480437	2.343720	1.738259	1.699197	1.933569	2.070286	2.421844	1.894507	2.519499	2.539030	1.874976
	2.441375	2.324189	2.480437	2.539030	2.499968	1.894507	2.421844	2.402313	1.718728	1.757790			
60	1002												
70	1011												
80	1022												
90	21												

Figura 22. Archivo de adquisición de datos de un premolar, sano, el paciente es de sexo masculino y tiene 21 años de edad

### 3.3 Imagen en 2 Dimensiones

El archivo de datos genera una matriz de datos que representa la imagen de la rugosidad del esmalte dental por cada cara de la pieza dental, para observar estos datos en imagen en 2D se realiza una conversión a mapa de bits a 256 colores a escala de grises, es decir, cada valor obtenido tiene un rango de 256 tonalidades



en escala de grises para ser representado en el diagrama de flujo 1 se observa el diagrama de flujo de esta conversión.

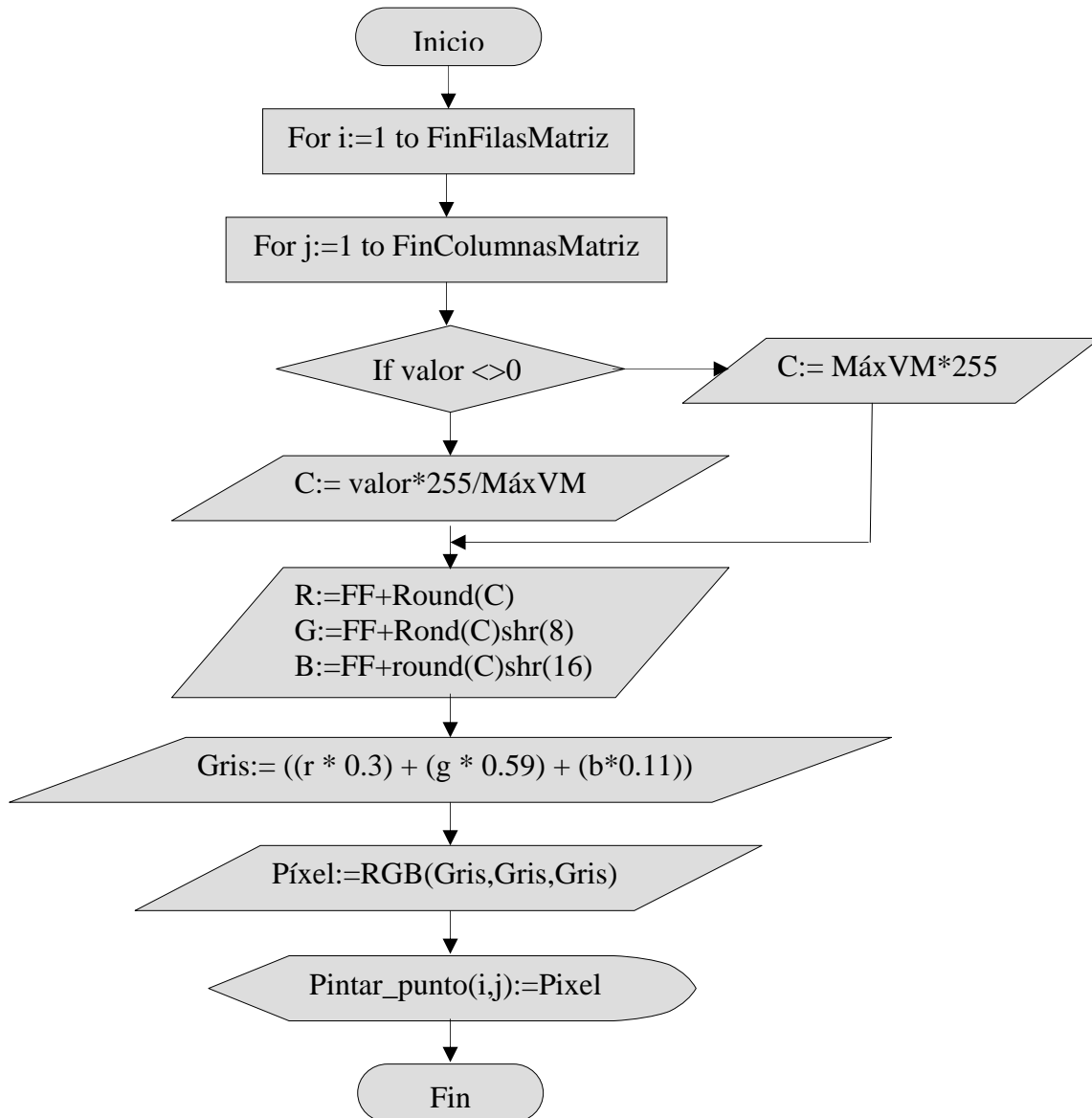


Diagrama de Flujo 1. Conversión de la matriz de datos a Imagen en 2D a escala de grises

El diagrama de flujo 1 muestra la conversión de una matriz de datos a una imagen en 2D a escala de grises a 256 tonalidades. Esto inicia con dos ciclos que realizan todo el recorrido de la matriz, se toma el valor que se tiene en la posición  $i$  (filas),  $j$  (columnas), con este valor se pregunta si es diferente a 0, si lo es entonces se multiplica el valor por 255 que es el máximo de los colores y el resultado se divide entre el valor máximo de la matriz, si el valor es igual a 0



entonces el valor sólo se multiplica por 255 para obtener un color oscuro o si no entonces sólo se deja 255. Con este nuevo valor etiquetado como  $C$  se obtiene el RGB, donde  $R$  es igual a la concatenación del número hexadecimal FF con el redondeo de  $C$ ,  $G$  es igual a la concatenación del número hexadecimal FF con el redondeo de  $C$  con corrimiento a la derecha en el 8avo bit y  $B$  es igual a la concatenación del número hexadecimal FF con el redondeo de  $C$  con corrimiento a la derecha en el 16avo bit, con estos tres valores obtenemos la intensidad del color en escala de grises es decir es el valor de intensidad que va a tener el píxel y por último pintamos en el área del mapa de bits. Esta imagen muestra algunas anomalías de la rugosidad dental pero se toma en cuenta que las intensidades pueden ser pequeñas por lo que se puede tener una imagen muy oscura o muy clara por lo que se utilizan algunos filtros para manipular la imagen y así se pueda observar algunas anomalías.

### **3.4 Eliminación de Ruido**

Los datos obtenidos por el sistema de adquisición de datos, pueden ser almacenados con alguna anomalía, lo cual puede suceder por las características propias del mecanismo automático, la tarjeta de adquisición, la luz del ambiente, hasta algún movimiento extraño de la tierra. Estos datos pueden ser observados en la imagen de 2 dimensiones o en 3 dimensiones, ya que puede haber discontinuidades muy grandes en los datos, las Imágenes obtenidas pueden contener ruido, por lo que es necesario eliminarlo para hacer un tratamiento posterior a la imagen. El ruido se puede eliminar mediante filtros, donde un filtro es un mecanismo que opera sobre una determinada señal de entrada para producir una señal de salida diferente. Ahora bien, una señal puede ser de dos tipos: una es la de alta frecuencia temporal, ésta es la que cambia periódicamente su valor en un corto período de tiempo, siendo la más conocida y la otra, la señal espacial, es decir, consideremos una imagen como la señal a tratar, se dice que una imagen con alta frecuencia espacial cambia periódicamente el valor de los niveles de intensidad o niveles de grises en un intervalo de tiempo espacial pequeño o



que es lo mismo en distancias pequeñas de la imagen, las bajas frecuencias espaciales corresponden a los cambios mas lentos en la variación de los niveles de grises. Con esto la transformación de la imagen se realiza para después operar con ella en el *Espacio de Frecuencias*. Existen una variedad de filtros que trabajan sobre el espacio de frecuencias como: Transformada de Fourier, Transformada del Coseno, Transformada de Walsh-Hadamard, Transformada Wavelet donde este filtro está revolucionando en el tratamiento de imágenes sobre todo en la comprensión de éstas y su formalidad parte del filtrado espacial, las tres transformadas anteriores trabajan bajo el filtrado espacial pero el mejor resultado en el tratamiento de imágenes es la transformada de Fourier tanto para la compresión como para el tratamiento de imágenes.

Para la solución del problema en este trabajo, se implementó la transformada de Fourier.

**Transformada de Fourier:**

En (1) se observa la fórmula matemática de la transformada de Fourier que se utilizará en la imagen y en (2) se observa la transformada inversa de Fourier [10].

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-2\pi i(ux + vy)) dx dy \dots\dots\dots (1)$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp(2\pi i(ux + vy)) dudv \dots\dots\dots (2)$$

Donde:

f(x,y) Es la Función Imagen.

F(x,y) Coeficiente de peso.

u y v Son frecuencias espaciales.

$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty}$  Son una suma generalizada de exponenciales complejas.

La expresión (2) se puede observar como una expansión de la función imagen f(x,y) en forma de una suma generalizada de exponenciales complejas. Para cada par de valores de la frecuencias espaciales u y v se tiene una exponencial en la



suma generalizada, dicha exponencial está multiplicada por el coeficiente de peso  $F(u,v)$ , estos pesos son obtenidos por (1) por lo tanto la transformada de Fourier de  $f(x,y)$  puede verse como los coeficientes de peso de la expansión de la función de intensidad en  $f$  en una suma de exponenciales.

### **3.5 Interpolación**

La interpolación en una imagen es el cálculo del valor de intensidad en una posición cualquiera como una función de los píxeles que lo rodean, es decir se encuentra un punto intermedio entre dos píxeles, esto es para dar forma o para observar el comportamiento de la intensidad de colores de un píxeles, la técnica utilizada para este trabajo es sacar la media entre dos puntos y el valor obtenido se coloca en medio de los puntos con el valor de intensidad promedio, esto se hace con 4 vecinos para generar una matriz interpolada y se pueda observar de una mejor manera la forma de la rugosidad, ver figura 23.

Como los datos obtenidos vienen dados con una cierta distancia de separación entre punto y punto, la interpolación permite recrear otros puntos intermedios para acotar las distancias iniciales. Si los datos están dados con una distancia de separación de 50 micras, al aplicar una interpolación la separación será de 25 micras, si se aplica la segunda interpolación la distancia será de 12.5 micras, pudiéndose observar el comportamiento de la rugosidad a menor distancia entre puntos.



	$P(x, y+1)$	
$P(x-1, y)$	$P(x, y)$	$P(x+1, y)$
	$P(x, y-1)$	

Figura 23. Método de Interpolación

La forma de realizar la interpolación es tomar el valor del punto  $p(x, y)$ , luego se toma el valor del punto  $p(x+1, y)$ , los valores de estos puntos se suman y se dividen entre 2 y este nuevo valor se coloca en medio de los dos puntos, una vez terminado, se realiza el mismo procedimiento pero ahora con el punto  $p(x, y)$  y con el punto  $p(x-1, y)$ , así hasta que se cubran los 4 vecinos, es decir primero se realiza por filas y luego por columnas hasta obtener la interpolación de la matriz completa. En el apéndice A se muestra este algoritmo [10].

### 3.6 Representación de la rugosidad en 3D

La matriz de datos obtenida por el dispositivo de adquisición representa el comportamiento, dirección y forma de la rugosidad del esmalte dental, la metodología para representarlo en tres dimensiones es necesario conocer cuáles son los puntos o píxeles que se pueden interconectar y así dibujar la rugosidad, la metodología implementada es llamada 8 vecinos o tablero de ajedrez la cual consiste en tomar un punto y los 8 puntos que están a su alrededor, en la figura 24 se muestra esta técnica. Este método consiste en tomar un punto de inicio con las coordenadas X-Y y obtener la métrica que hay en ese punto con respecto a los 8 vecinos. Con este método se tienen 8 métricas las cuales se almacenan en un arreglo, este arreglo se ordena y se toma el valor de la cuarta posición ya que es la mitad del tamaño del arreglo. El valor obtenido se le suma 0.5 y se le resta 0.5 y queda un rango, ahora se obtiene la distancia del punto inicial con los que están a

su alrededor y si el valor está dentro del rango entonces se traza una línea y si está fuera del rango no se traza nada. De esta forma se observa el comportamiento de los puntos, ahora bien, si se desea observar otros comportamientos sólo basta con tomar cualquier posición del arreglo y tomar la métrica. En el apéndice B se muestra el algoritmo de este método.

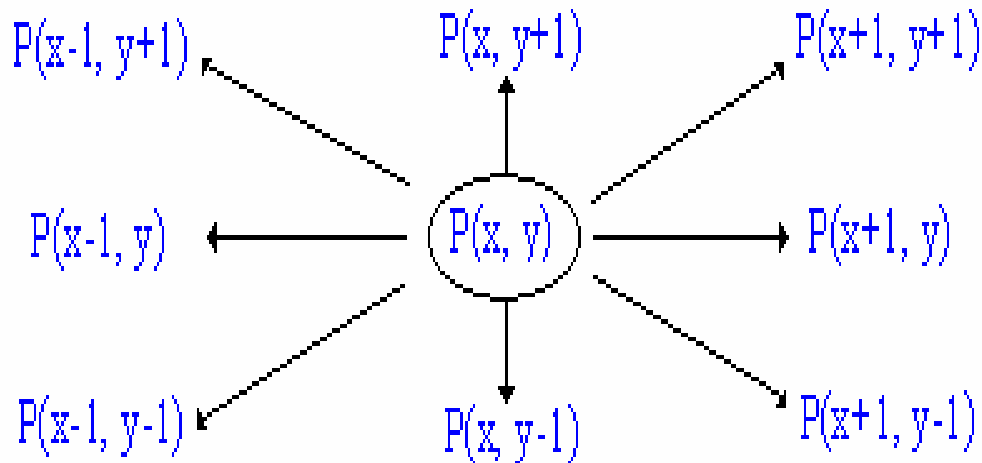


Figura 24. Método de los 8 vecinos

### 3.7 Reconstrucción del Volumen.

La reconstrucción de la rugosidad se realiza por cada cara de la pieza dental, para representar el volumen de la pieza dental es decir que las 5 caras estén unidas y se pueda observar la pieza dental en tres dimensiones, se utiliza una técnica para reconstrucción de imágenes segmentadas, esta técnica consiste en formar un cubo en el espacio de 3 dimensiones y se pinta una imagen por cada cara del cubo, sólo es necesario realizar un ajuste hacia la imagen para que las caras den la forma de diente. En la figura 25 se observa el método del cubo y en apéndice A se muestra el pseudocódigo desarrollado.

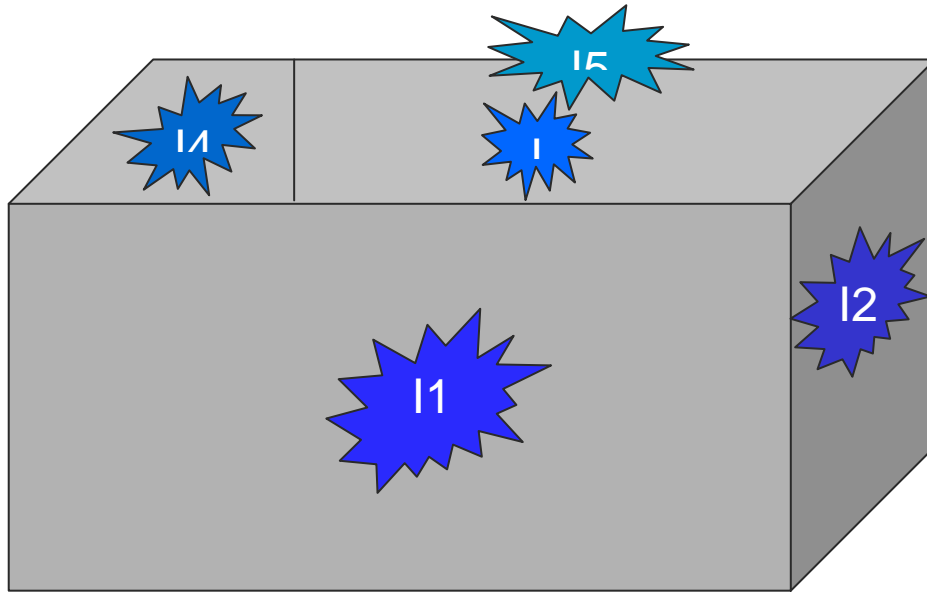


Figura 25. Método del Cubo

### **3.8 Software de Representación de Datos**

Se diseñó un software para la representación de los datos obtenidos por el sistema de adquisición de datos de la rugosidad, ver figura 26, este sistema consta de una pantalla principal la cual lleva por nombre “*Detección y Localización de Defectos en la Superficie del Esmalte Dental en 3 Dimensiones*”, tiene tres iconos iniciales los cuales son:

- 1) Abrir archivo de datos de la rugosidad del esmalte dental
- 2) Abrir archivo de datos de la rugosidad de materiales dieléctricos
- 3) Acerca de este sistema.

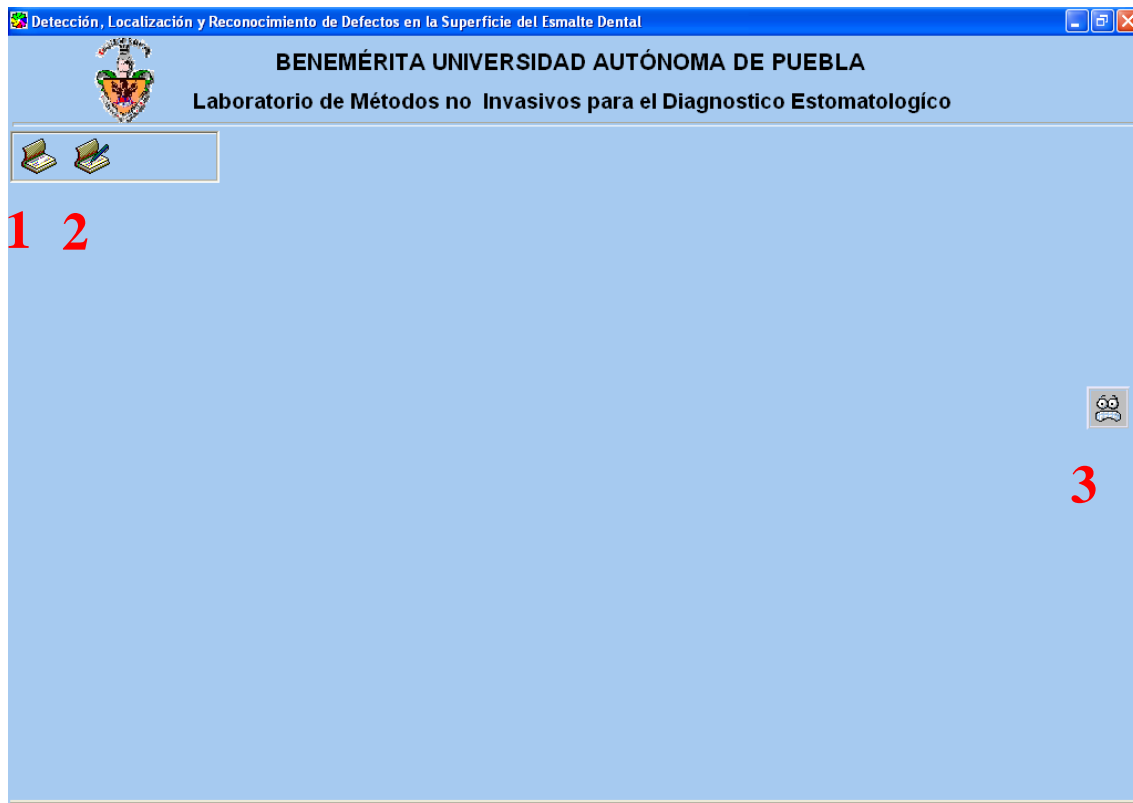


Figura 26. Pantalla principal del sistema de representación de datos

Al presionar el icono de Abrir archivo de datos de la rugosidad del esmalte dental, se presenta una nueva pantalla como se observa en la figura 27, la cual se divide en tres módulos:

- Características de las piezas dentales:  
En este módulo se observan las características de la pieza dental analizada por el sistema de adquisición las cuales son:
  - Tipo de pieza dental
  - Estado de la pieza dental
  - Sexo del paciente
  - Edad del paciente.
- Representación matricial de los datos adquiridos.  
En este módulo se presentan los datos obtenidos por el mecanismo automático donde se le presenta al especialista el número total de datos que se obtuvieron, el valor máximo y mínimo que se tiene en esta matriz,

además se presenta una serie de pestañas que indican el número de caras del diente así como el nombre de cada una de ella, al presionar la pestaña aparecen los datos en forma matricial de la cara, el total de datos de la cara, valores máximos y mínimos de la cara y la figura en forma de muela cambia de posición según sea la cara que se esté observando.

- Representación de imagen en dos dimensiones de los datos adquiridos. En esta zona se presentan los datos representados en imagen en escala de grises a 256 tonos de grises, esto para que el especialista observe la forma de la pieza dental y si existe alguna anomalía es posible que sea observable. El usuario puede aumentar la imagen y trasladarse sobre ella y observar las intensidades.

En la figura 28 podemos observar estas tres zonas.

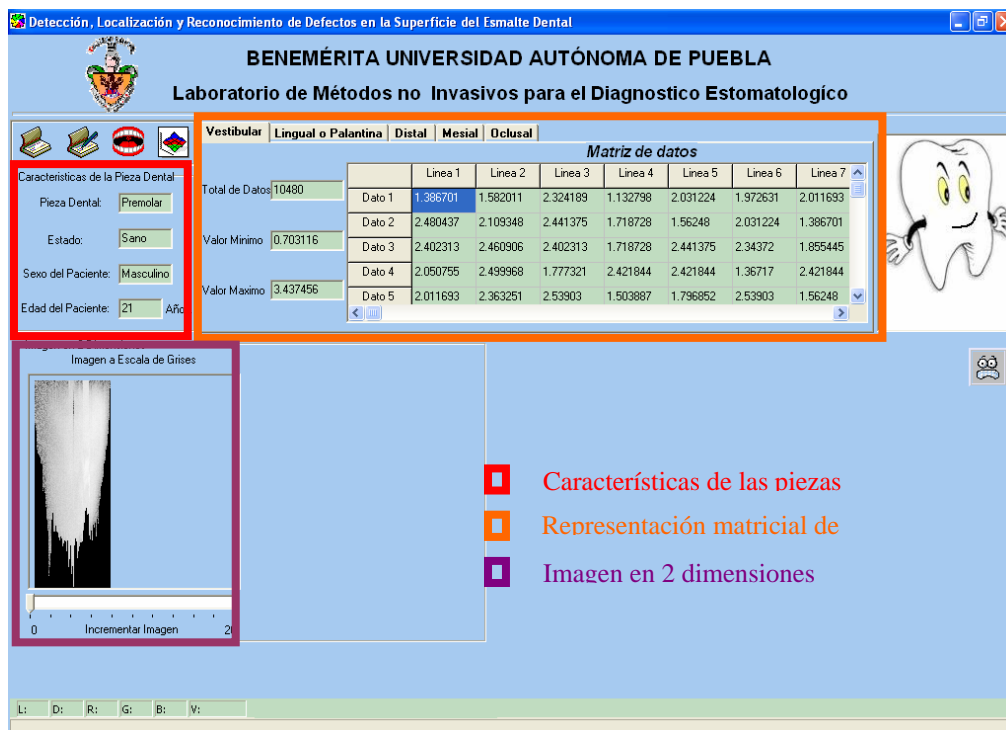


Figura 27. Módulo de la rugosidad de piezas dentales

También se presentan dos iconos los cuales el icono 1, muestra una imagen de rayos X de las piezas dentales esto se observa en la figura 27. El icono

2 presenta una nueva pantalla donde se observas la imagen en tres dimensiones, figura 28.

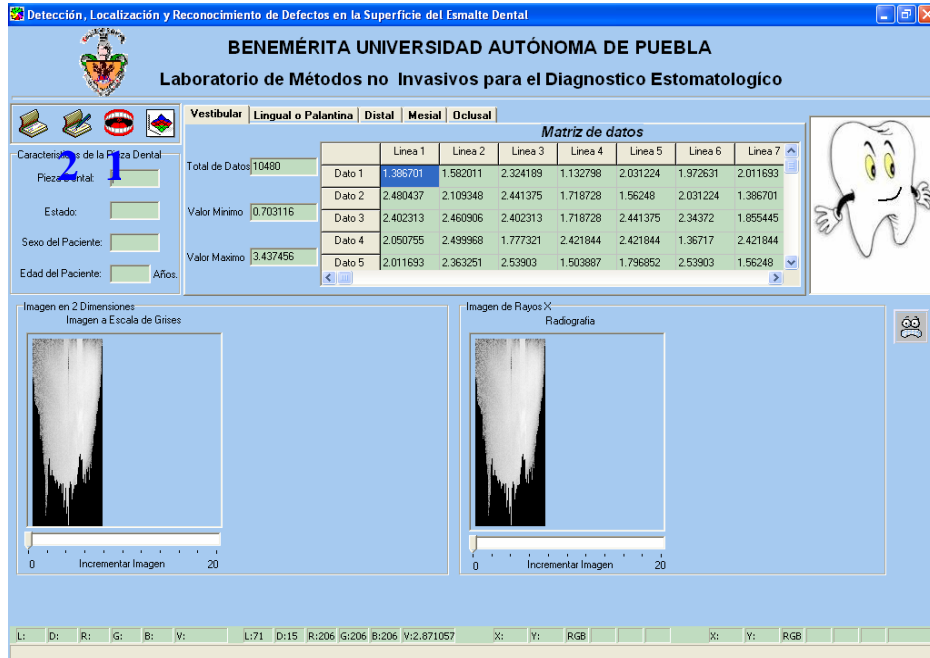


Figura 28. Imagen en 2D y con radiografía

La imagen de la radiografía también se presenta en escala de grises, para observar algún defecto es necesario aplicar filtros para tener una mejor resolución.

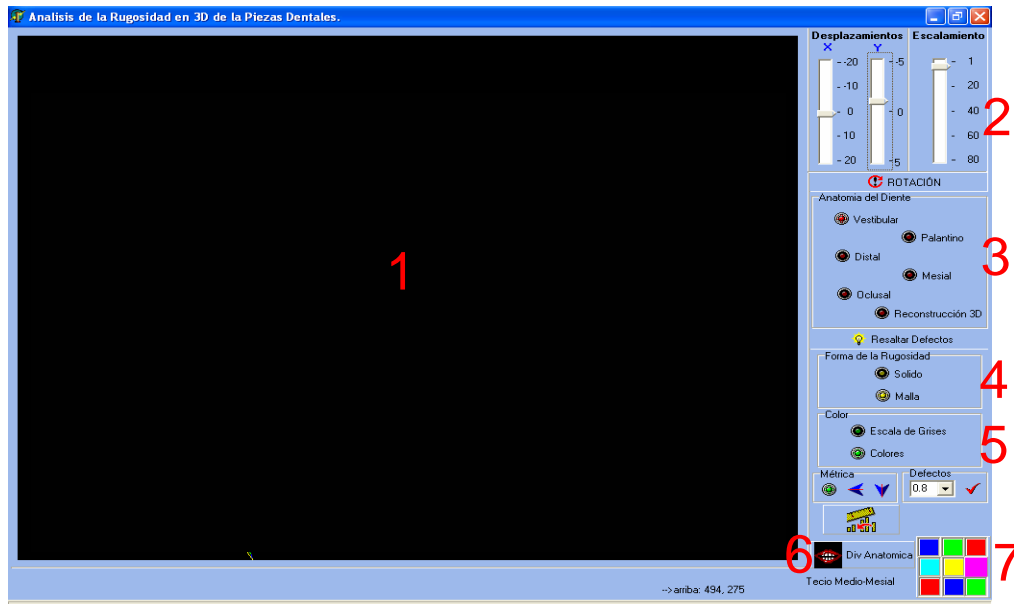


Figura 29. Interfaz de la animación en 3 dimensiones



La interfaz de reconstrucción de la rugosidad del esmalte dental en 3 dimensiones está compuesta de 7 módulos los cuales son:

- 1) *Área de representación:* en esta zona el usuario observa la figura en 3 D
- 2) *Área de Animación:* se tienen 2 barras de desplazamiento para desplazar la figura sobre los ejes X, Y, se tiene una barra de desplazamiento para escalar la imagen, además un botón que indica la rotación.
- 3) *Anatomía del diente:* en esta área se tiene 4 o 5 botones para mostrar la imagen de cada cara de la pieza dental el número de botones depende del tipo de pieza dental. Y un botón para formar la reconstrucción de la pieza dental utilizando todas las caras del diente.
- 4) *Forma de la rugosidad:* en esta área se tienen dos botones uno para representar la rugosidad en forma de sólido y el otro botón para representar la rugosidad en forma de mallas.
- 5) *Color:* hay dos botones para indicar si la figura se observa en escala de grises o a colores.
- 6) *Métricas:* el usuario puede indicar si la conexión de puntos para la representar la rugosidad es con una métrica baja, una métrica media o con una métrica alta. Con la finalidad de observar la diferencia de la conexión de puntos.
- 7) *División anatómica:* esta zona es muy importante para el especialista ya que indica la posición del posible defecto que se encuentra en la pieza dental, cada casilla del panel de colores indica el nombre de los tercios en que se dividió la misma.

Cuando se presiona el icono de Abrir archivo de datos de la rugosidad de materiales dieléctricos se presenta un módulo, éste se divide en dos zonas como se observa en la figura 30. Donde sólo se presenta el área de los datos obtenidos en forma matricial y los datos en imagen en 2 dimensiones.

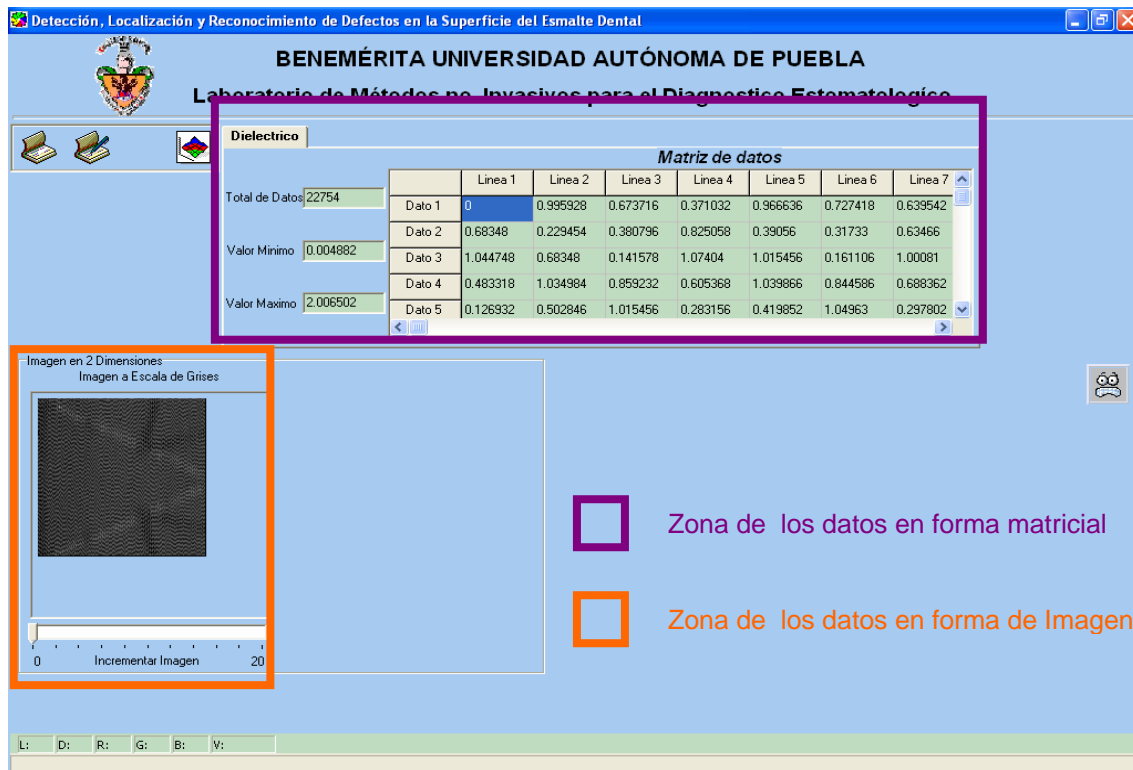


Figura 30. Módulo de la rugosidad de materiales dieléctricos

El icono de reconstrucción pasa a una interfaz donde se observa la figura en tres dimensiones, esto se observa en la figura 31.

Esta interfaz cuenta con 5 módulos los cuales son:

- 1) *Área de representación:* en esta zona el usuario observa la figura en 3 D
- 2) *Área de Animación:* se tienen 2 barras de desplazamiento para desplazar la figura sobre los ejes X, Y, una barra de desplazamiento para escalar la imagen y un botón que indica la rotación.
- 3) *Forma de la rugosidad:* en esta área se tienen dos botones uno para representar la rugosidad en forma de sólido y el otro botón para representar la rugosidad en forma de mallas.
- 4) *Color:* hay dos botones para indicar si la figura se observa en escala de grises o a colores.

- 5) *Métricas*: el usuario puede indicar si la conexión de puntos para la representar la rugosidad es con una métrica baja, o una métrica media o con una métrica alta.

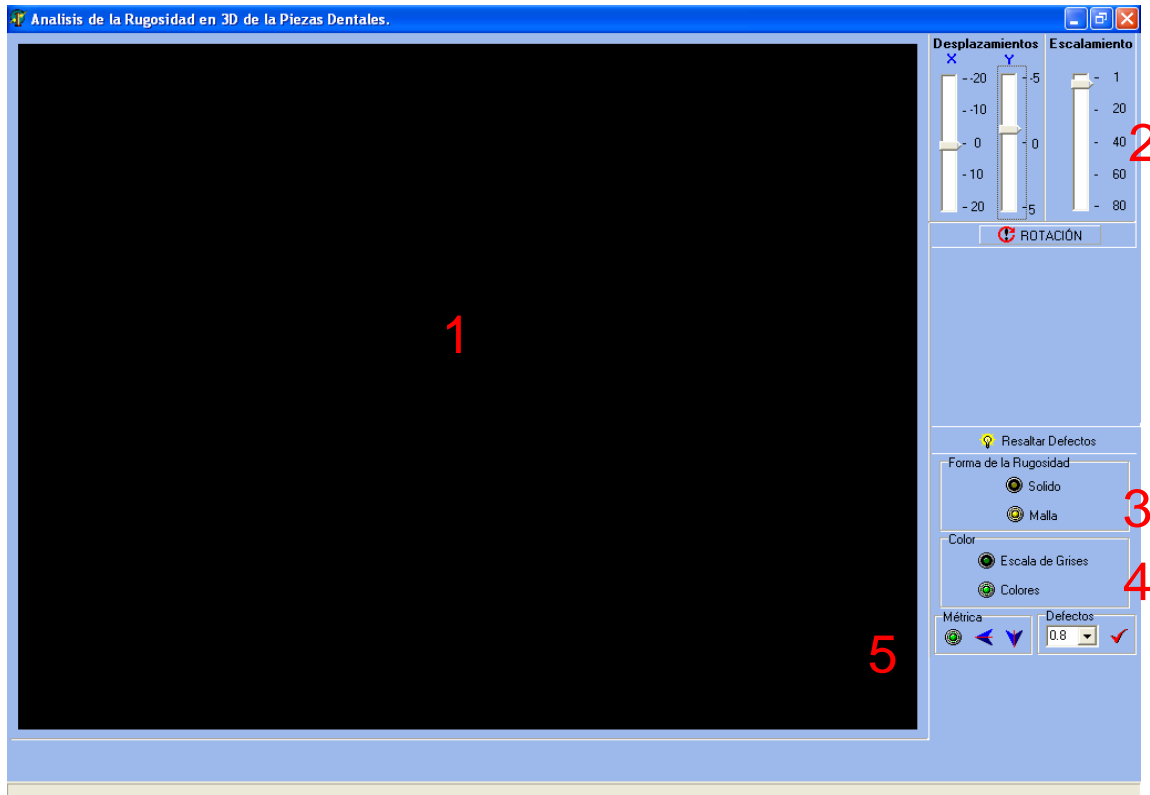


Figura 31. Interfaz de reconstrucción de la rugosidad de materiales dieléctricos en 3 dimensiones

En el apéndice B se muestra el algoritmo del funcionamiento interno del sistema.

# CAPÍTULO 4

## **EXPERIMENTOS Y RESULTADOS**

Los experimentos se realizaron con piezas dentales extraídas de las jornadas que realizaron los pasantes de estomatología de la FE de la BUAP, estas piezas fueron: incisivos laterales inferiores, premolares superiores e inferiores, caninos supernumerario y terceras molares superiores e inferiores. Las odontectomías se realizaron con un elevador de modo que la corona del diente no fuera dañada, una vez extraídas se lavaron a chorro de agua y fueron conservadas en solución fisiológica al 0.9% de cloruro de sodio, donde cada semana se realizó el cambio de solución. En la figura 32 se observan algunas muestras almacenadas estas contienen una etiqueta con un número para que se identificaran con facilidad.



Figura 32. Piezas dentales extraídas y almacenadas

El experimento se realizó de la siguiente manera:

- 1) Se tomó la pieza dental con unas pinzas de curación y se secó con campos estériles.
- 2) Se colocó en el mecanismo automático y se realizó el escaneo.
- 3) El escaneo tardó 12 horas.



- 4) El archivo resultante se utilizó con el software de representación de datos y se obtuvieron las imágenes en dos dimensiones y las animaciones en tres dimensiones.
- 5) Luego se realizó un barrido a la pieza dental con un microscopio metalográfico.
- 6) También se tomó una radiografía con rayos X a la pieza dental.
- 7) El estomatólogo analizó y comparó las imágenes del microscopio y radiografía con la animación en tres dimensiones.
- 8) El especialista le colocó a la pieza dental ácido grabador con solución del 38 % de ácido fosfórico durante 45 segundos y un minuto bajo chorro de agua.
- 9) Se volvió a analizar con el mecanismo automático.
- 10) Se realizó un barrido con un microscopio metalográfico
- 11) Se tomó una radiografía con rayos X.
- 12) Se compararon las imágenes resultantes.

### **Experimento 1.**

Se realizó un experimento con una molar, en la figura 33 se muestra en la pantalla los datos en forma matricial y en forma de imagen en 2 dimensiones a escala de grises de la cara vestibular. En la figura 34 se muestra la imagen resultante del filtro de la transformada de Fourier y las imágenes filtradas de pasa baja, pasa alta, pasa banda. En la figura 35 se presenta la inversa de la transformada de Fourier pero ya con eliminación de ruido por medio de un barrido de los contornos de la imagen de la transformada de Fourier. En la figura 36 se muestra la figura en 3 dimensiones con textura en sólido y en escalas de grises, en la figura 37 se muestra esta misma cara pero en mallas. En la figura 38 y 39 se observa la cara vestibular en sólido y en malla respectivamente. En la figura 40 se presenta el defecto de la pieza dental donde se linealizaron todos los datos para obtener un realce del defecto. En la figura 41 se presenta la localización del defecto es decir se realizó la división anatómica a la cara vestibular en 3 dimensiones.

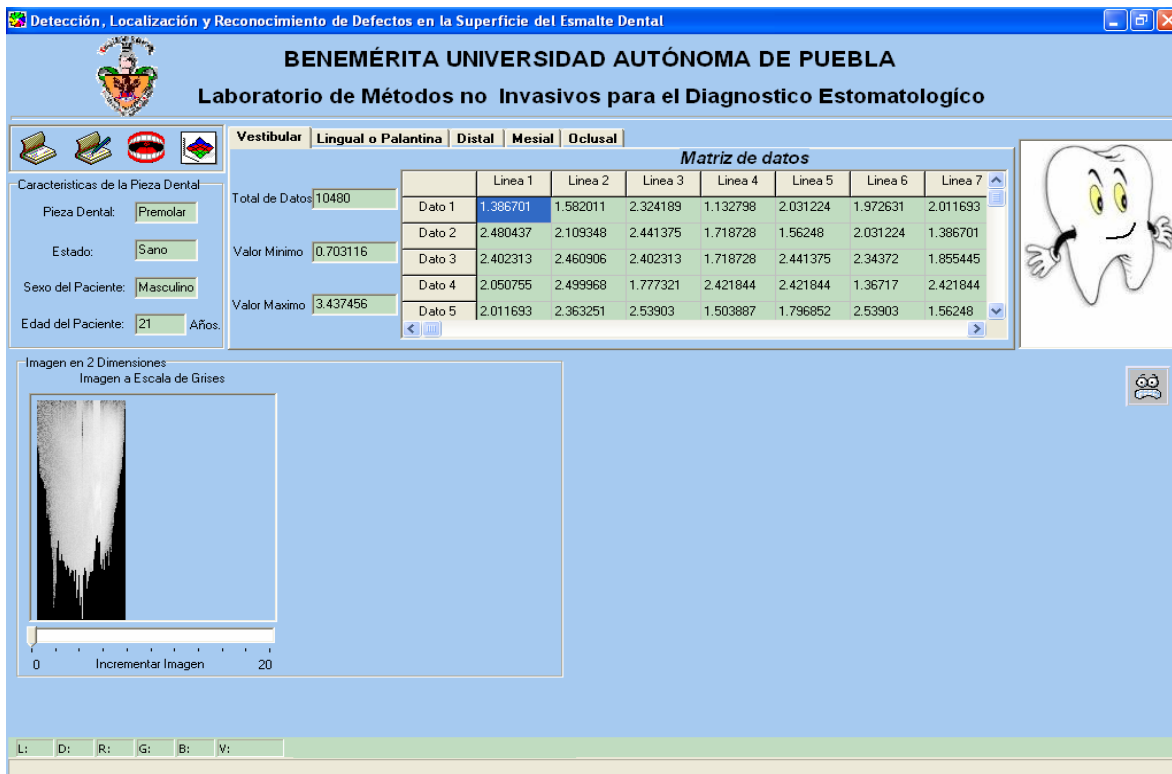


Figura 33. Datos en forma matricial y en forma de imagen de la cara vestibular de una molar

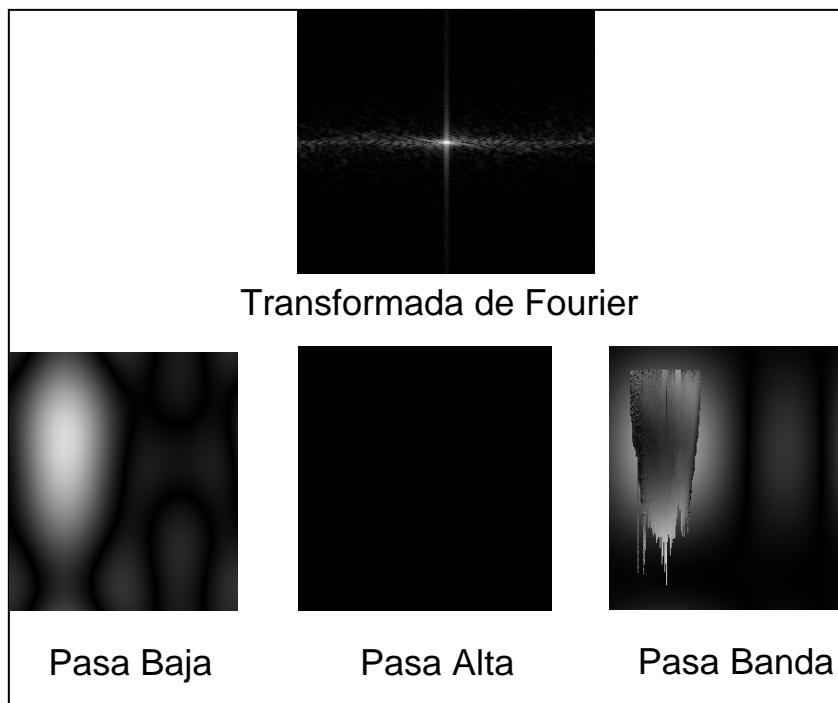


Figura 34. Imágenes del filtro de la Transformada de Fourier de la cara vestibular de una premolar

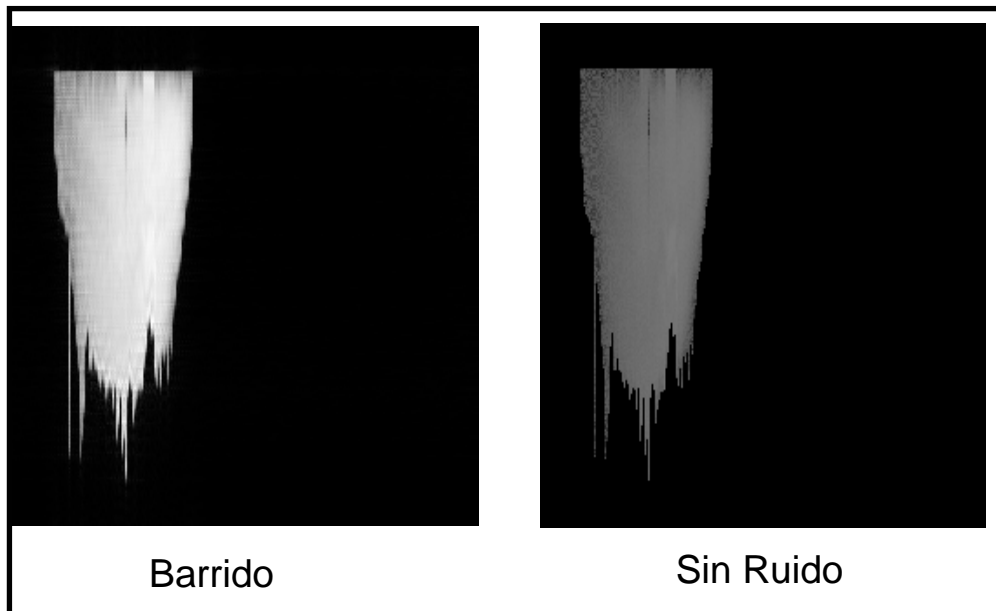


Figura 35. Imagen limpia de la cara vestibular de una molar

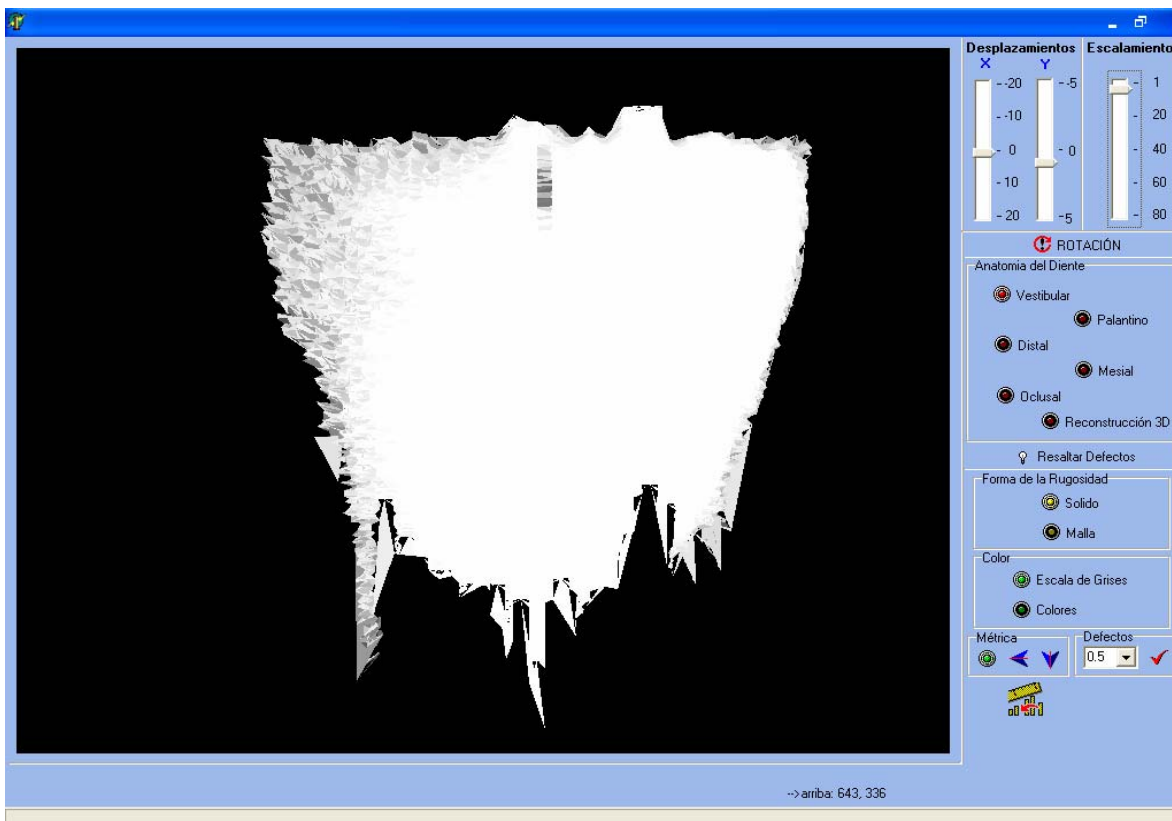


Figura 36. Reconstrucción en 3D de la rugosidad de la cara vestibular de una molar con textura en sólido a escala de grises

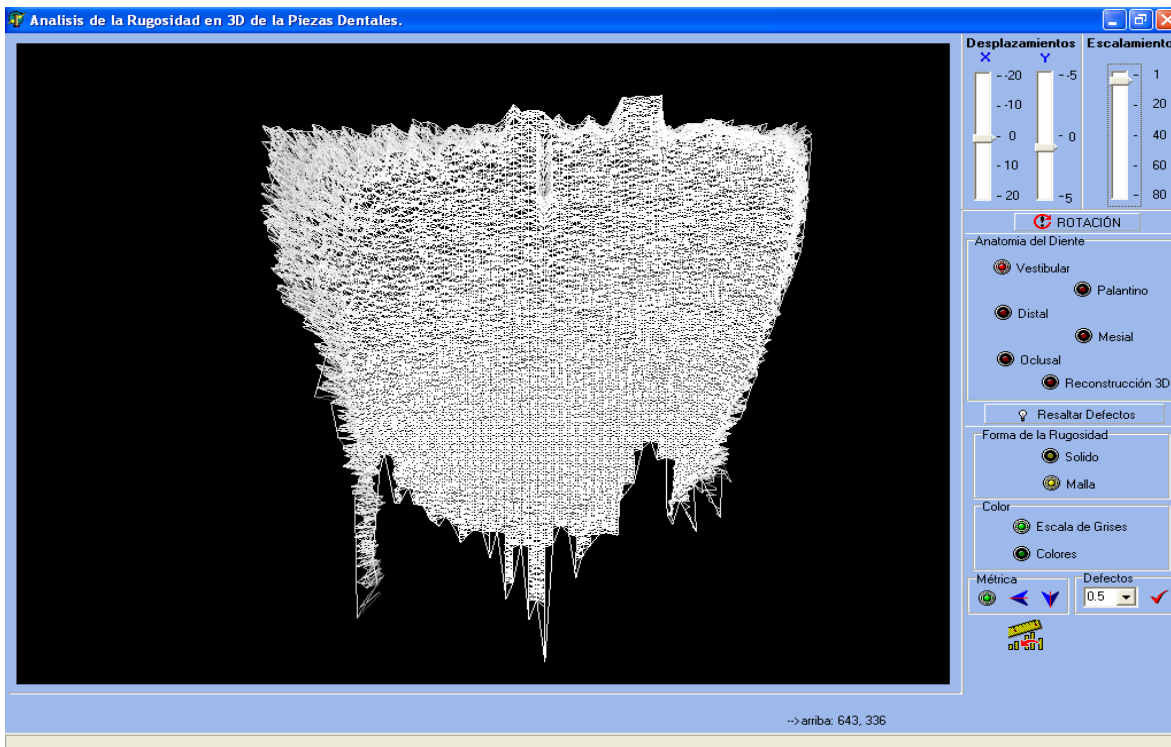


Figura 37. Reconstrucción en 3D de la rugosidad de la cara vestibular de una molar con textura en mallas a escala de grises

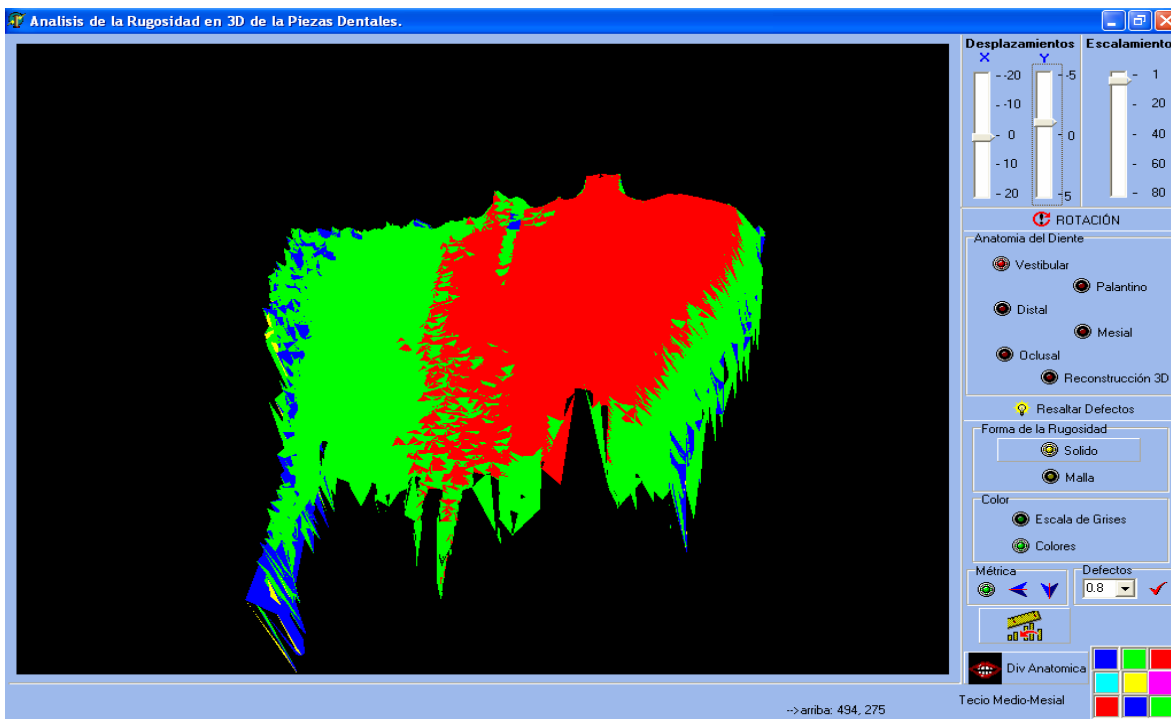


Figura 38. Reconstrucción en 3D de la rugosidad de la cara vestibular de una molar con textura en sólido a color

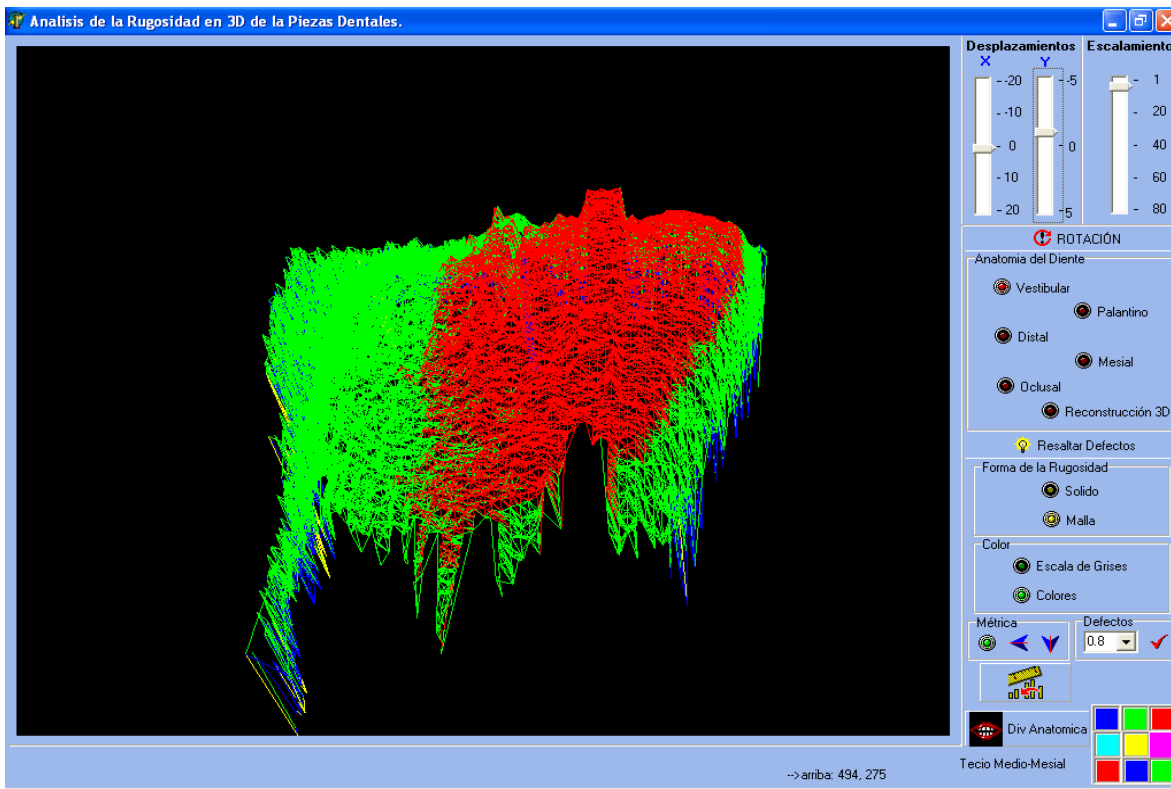


Figura 39. Reconstrucción en 3D de la rugosidad de la cara vestibular de una molar con textura en mallas a color

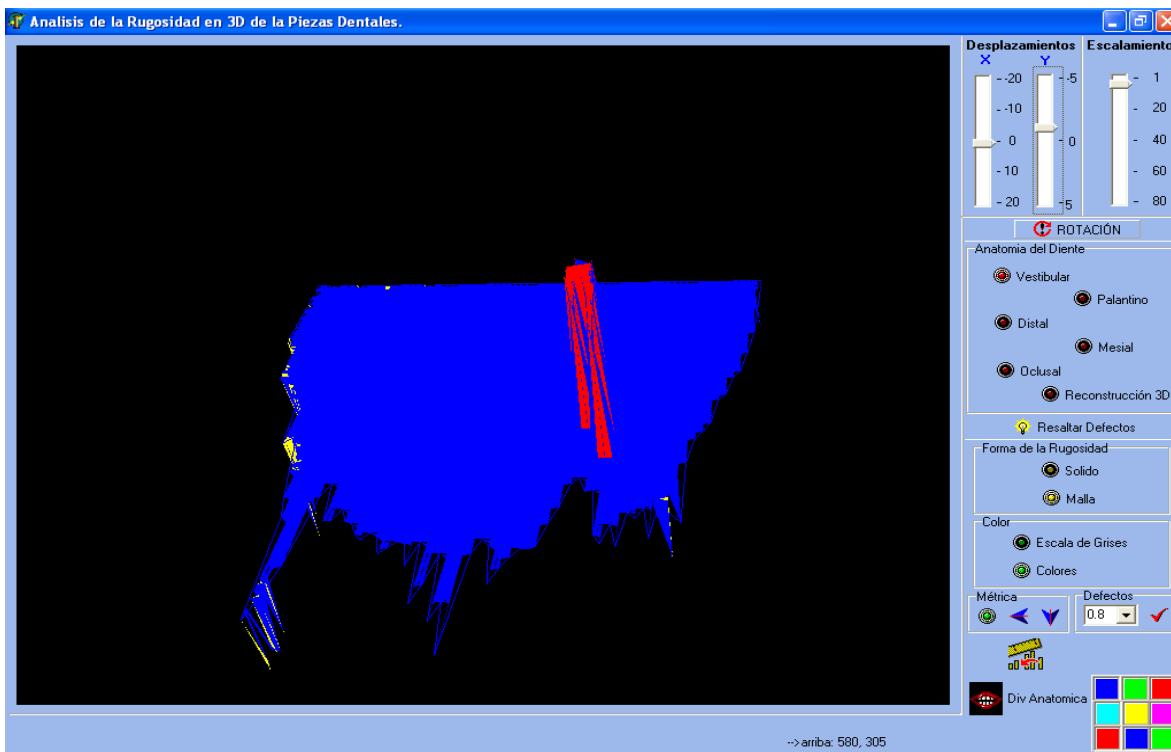


Figura 40. Resalte del defecto de la pieza dental a color

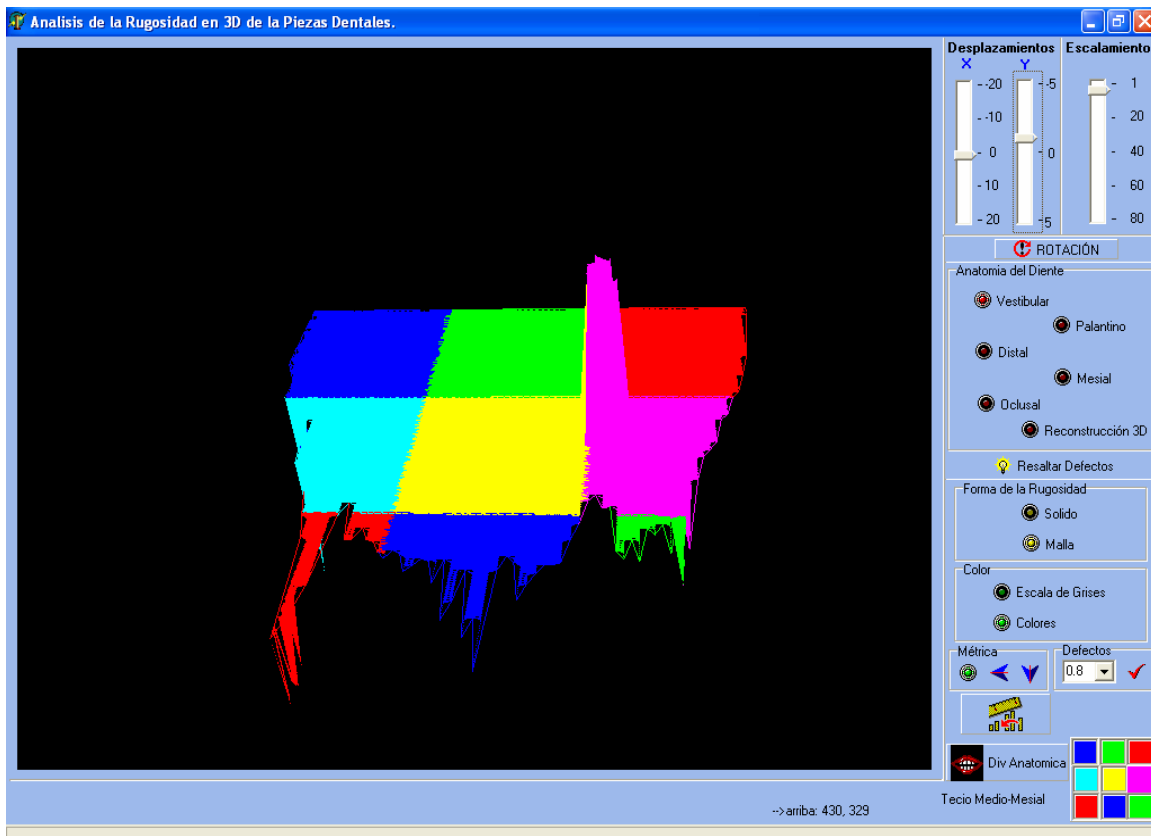


Figura 41. Resalte del defecto de la pieza dental con la división anatómica

## Experimento 2.

Este experimento se realizó con una pieza molar, la cual se colocó en el mecanismo automático por 8 minutos, los datos adquiridos se almacenaron y se le dio el tratamiento con el software desarrollado. En la figura 42 se observa la pantalla principal donde se observan los datos adquiridos, así, como la imagen en dos dimensiones. Las figuras 43 y 44 muestran la imagen en 3 dimensiones a escala de grises en diferentes perspectivas. En la figura 45 se muestra la imagen obtenida por un microscopio metalográfico, donde se observa que en las imágenes obtenidas por ambos métodos no hay anomalía.

Posteriormente se desmineralizó la pieza dental, en las figuras 46 y 47 se observa la imagen obtenida por el sistema en diferentes perspectivas en donde se

observa el defecto, dicha imagen se comprobó con el microscopio metalográfico, figura 48, en la cual se observa el defecto.

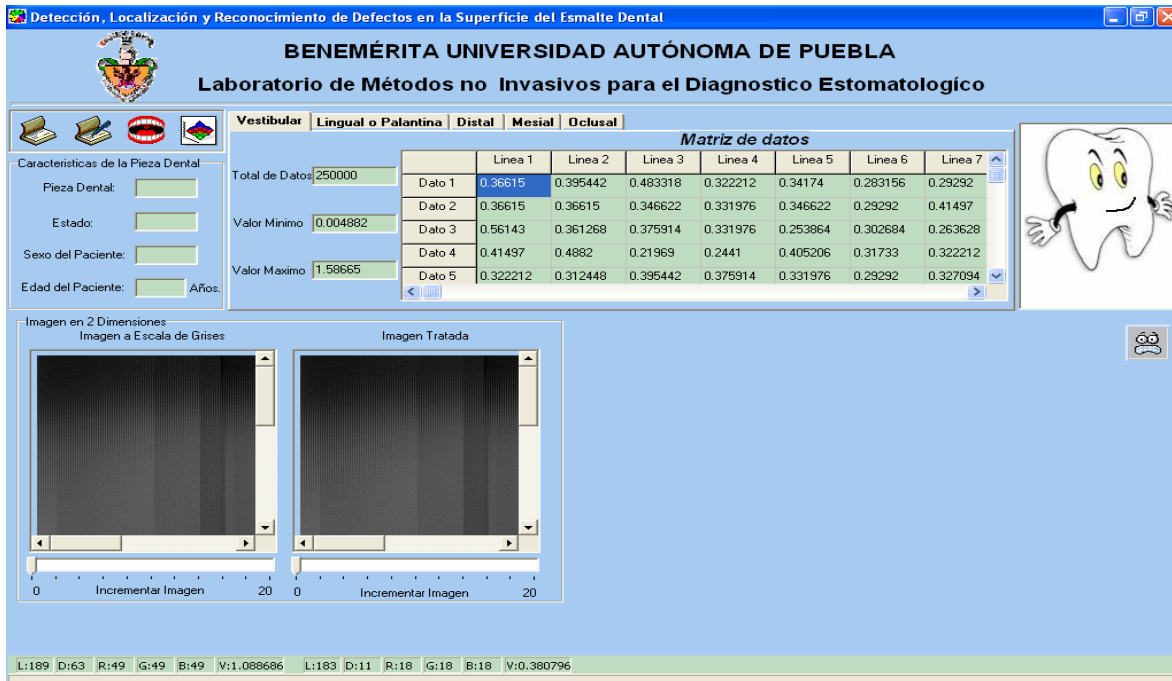


Figura 42. Datos en forma matricial y en imagen de 2 dimensiones en escala de grises de una pieza dental

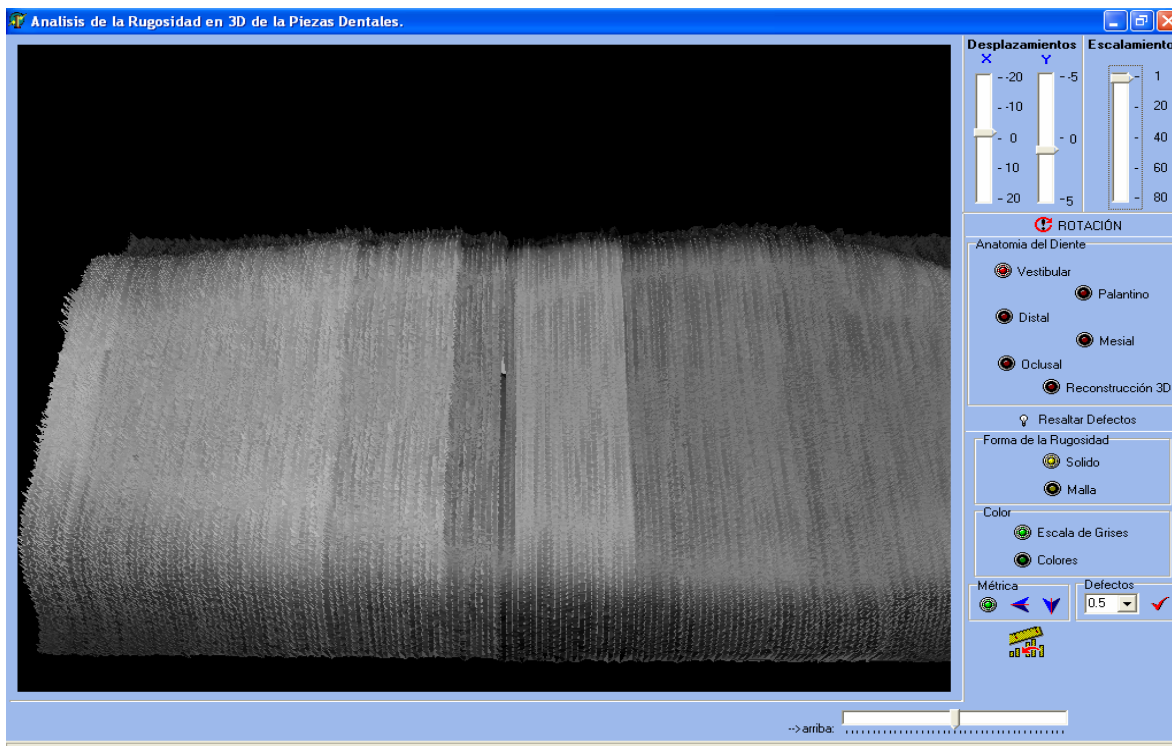


Figura 43. Reconstrucción en tres dimensiones de una pieza dental sin defecto



Figura 44. Reconstrucción den tres dimensiones de una pieza dental sin defecto con otro ángulo de visión



Figura 45. Imagen de la rugosidad de la pieza dental analizada con microscopio metalográfico

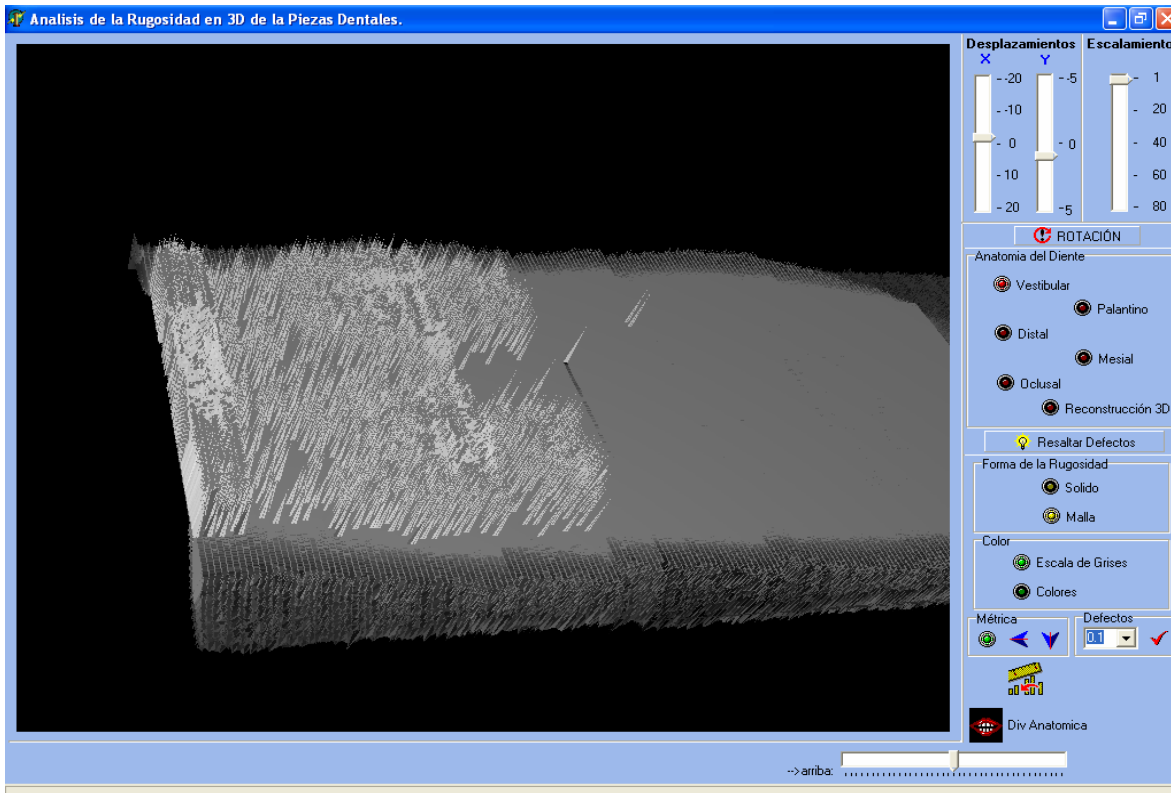


Figura 46. Realce del defecto de una pieza dental, en tres dimensiones

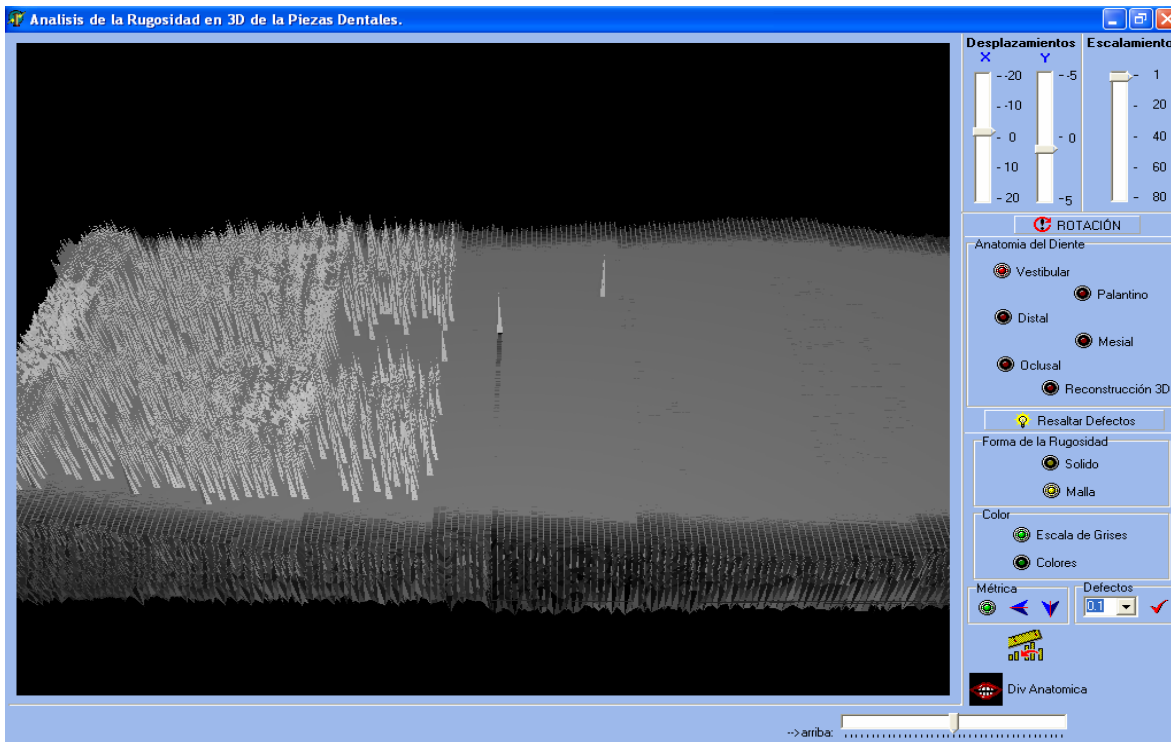


Figura 47. Realce del defecto de una pieza dental, en tres dimensiones. Observada desde otro ángulo

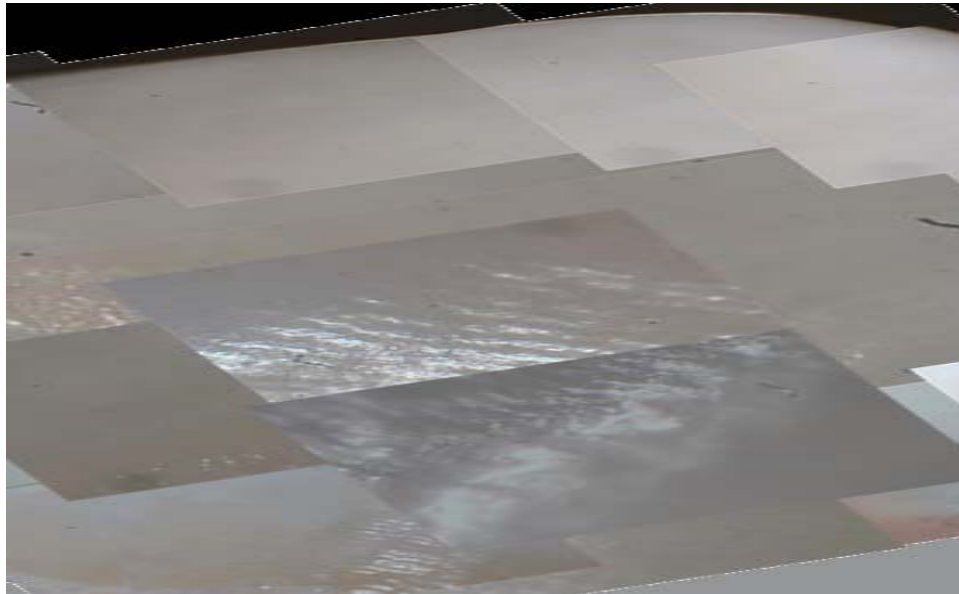


Figura 48. Imagen de una pieza dental con defecto analizada con microscopio

A continuación se presentan dos resultados obtenidos por el software de representación de datos. En la figura 49 se observa la parte central de una premolar donde no se observa el defecto y en la figura 50 se observa la desmineralización que se le realizó. En la figura 51 se muestra la parte central de otra pieza premolar y en la figura 52 observamos la desmineralización de ésta.

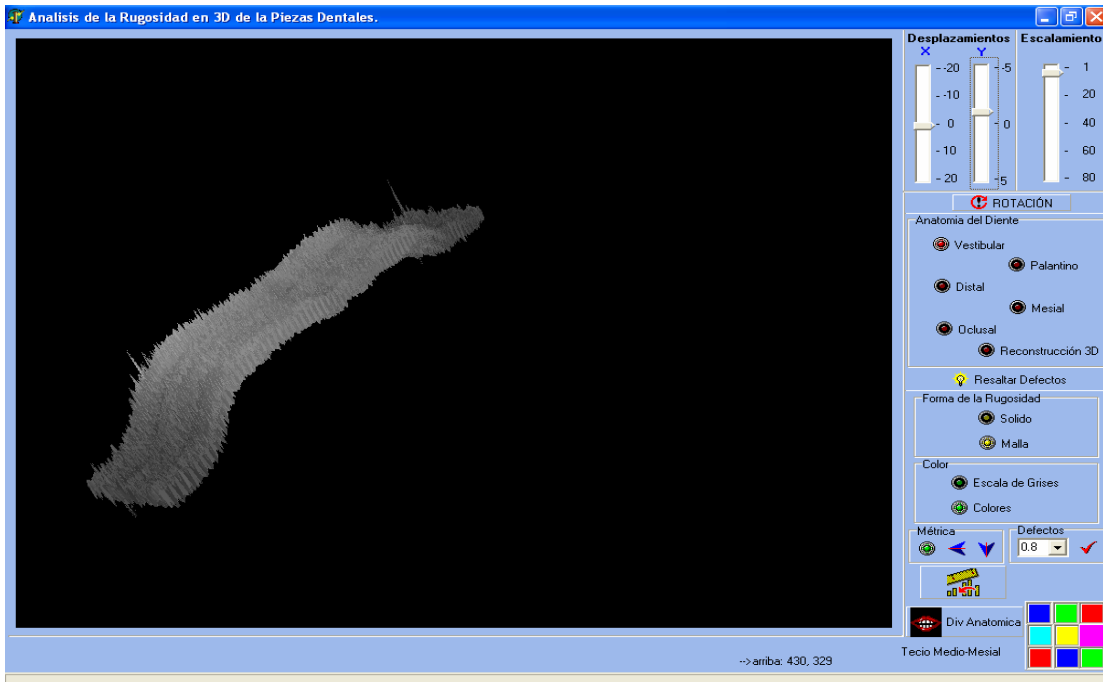


Figura 49. Imagen de la parte central de una pieza dental sin defecto

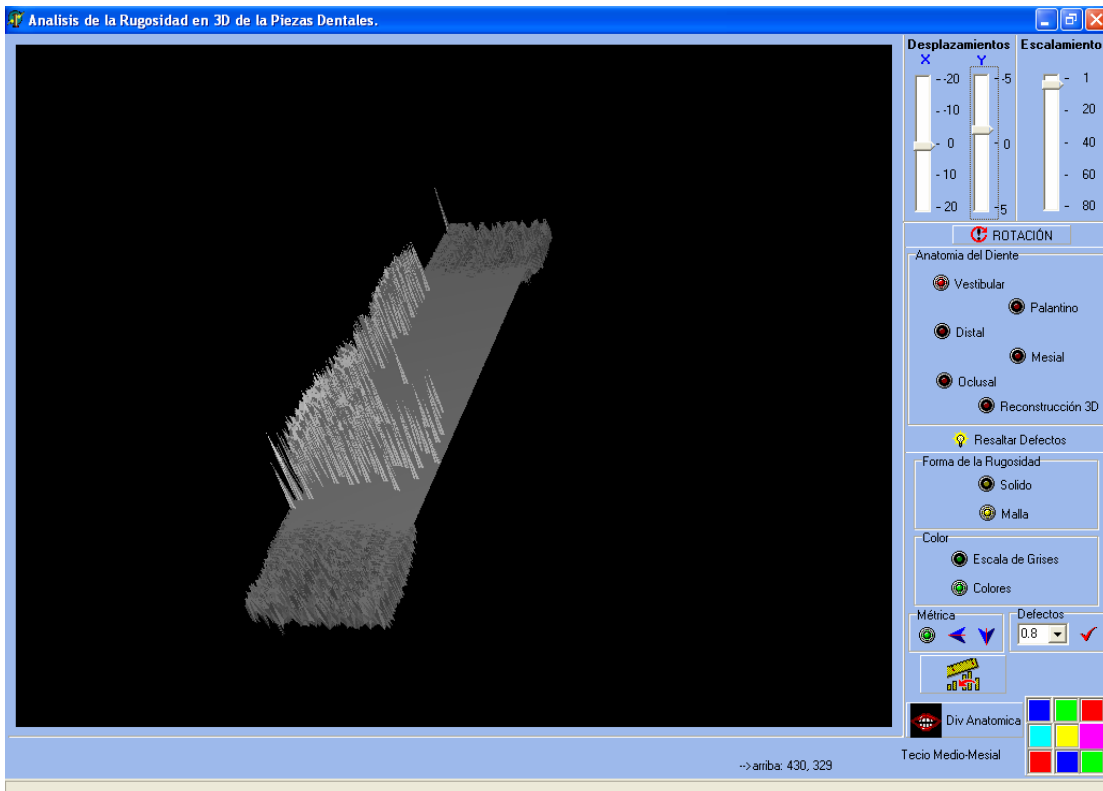


Figura 50. Imagen de la parte central de una pieza dental con defecto

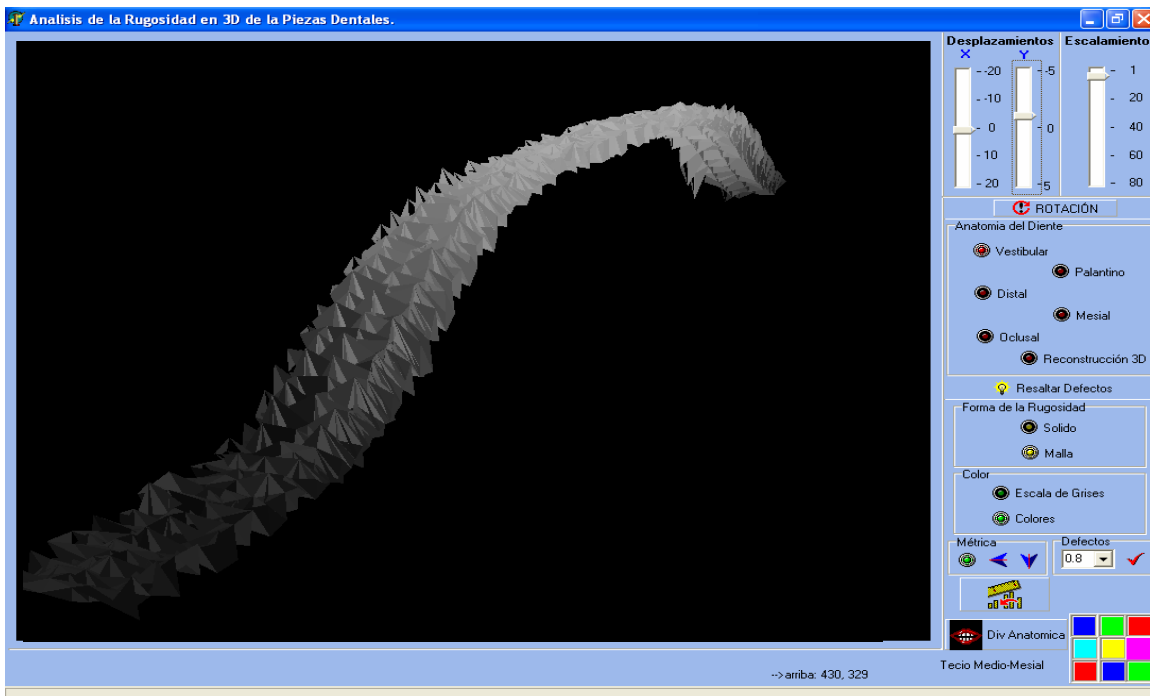


Figura 51. Imagen de una pieza dental tomada de la parte inferior hacia la superior de la parte de en medio de una pieza dental sin defecto

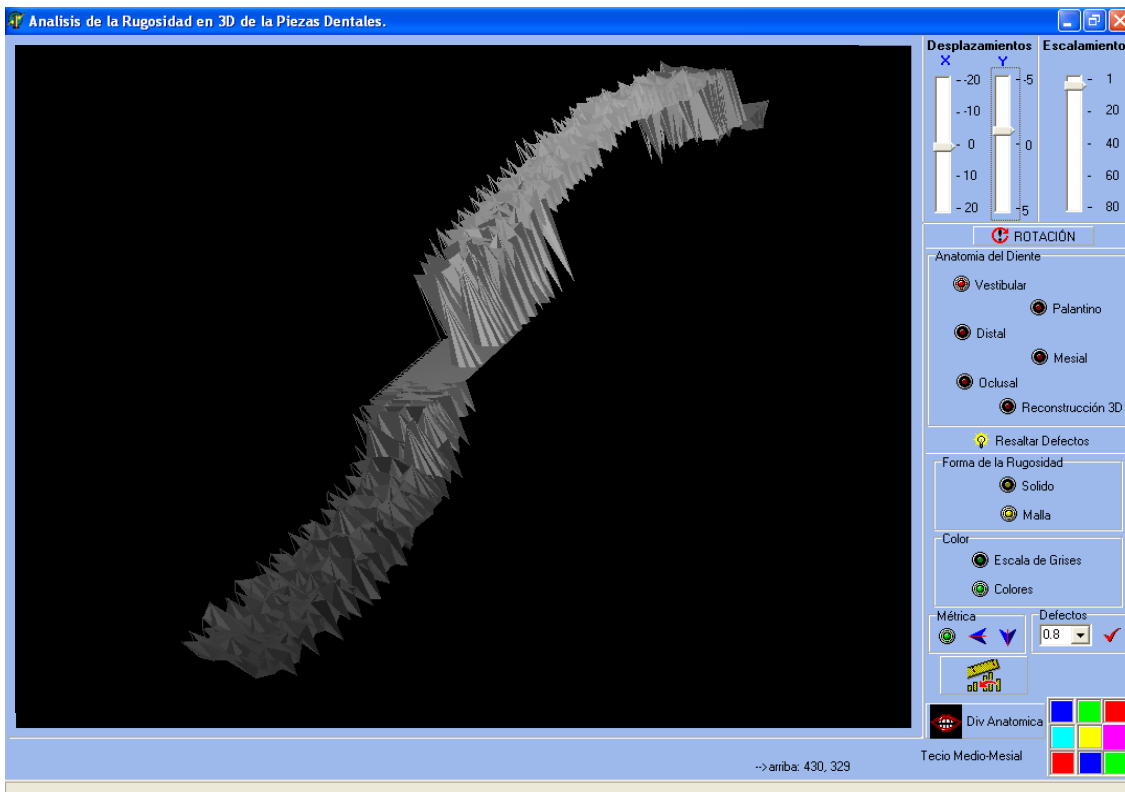


Figura 52. Imagen de una pieza dental toma de la parte inferior hacia la superior de la parte de en medio de una pieza dental con defecto

### Experimento 3.

Para validar la funcionalidad del sistema desarrollado con la metodología implementada para la detección de defectos en la rugosidad del esmalte dental se realizó una prueba con un dieléctrico el cual fue una mica, a la cual se le provocó un defecto, con la finalidad de que el sistema lo detectara. En la figura 53 se presentan los datos en forma matricial, así como la imagen en dos dimensiones a escala de grises. En la figura 54 se presenta la imagen en tres dimensiones a escala de grises. En la figura 55 se muestra la imagen en tres dimensiones de la mica a color con resalte de defecto. En las figuras 56 y 57 se muestra la mica con resalte de defecto, observada desde otro ángulo, a color y escala de grises respectivamente.

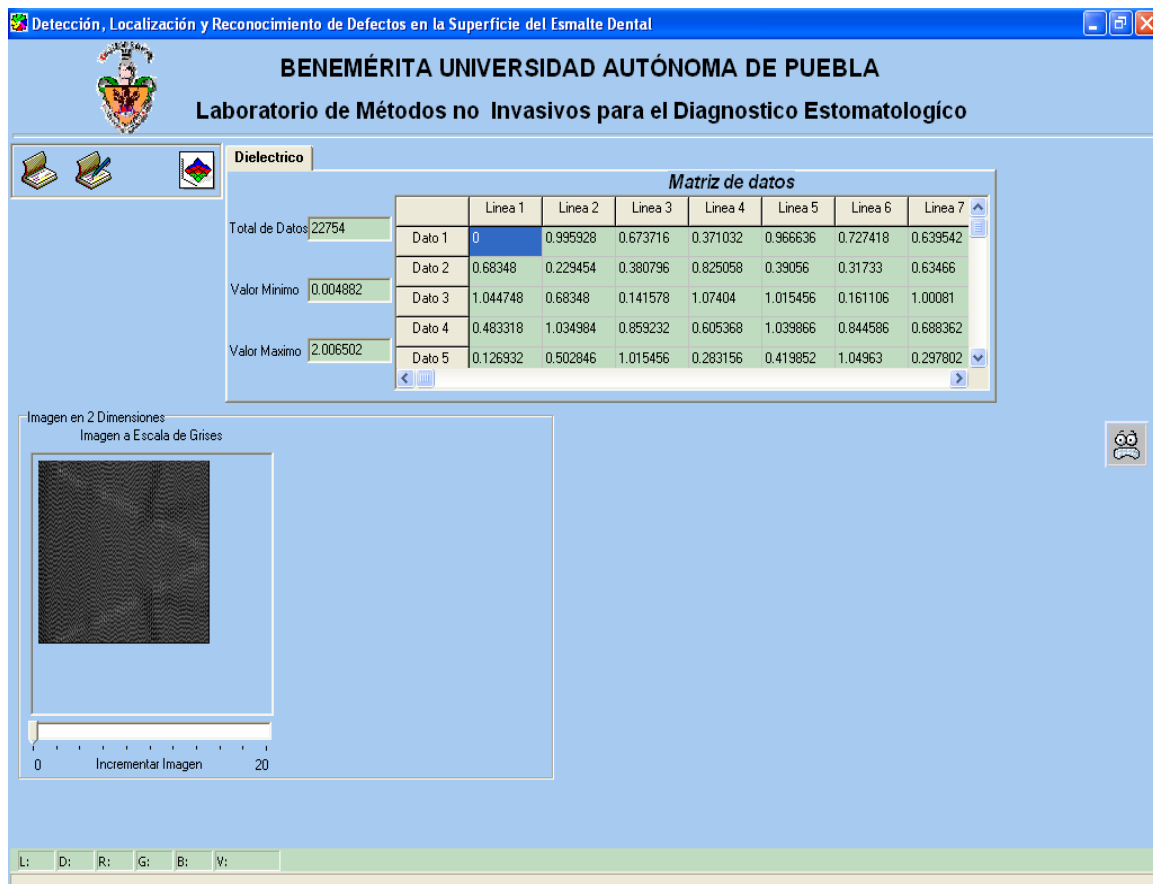


Figura 53. Representación de los datos de un dieléctrico en forma matricial y en imagen en 2 dimensiones

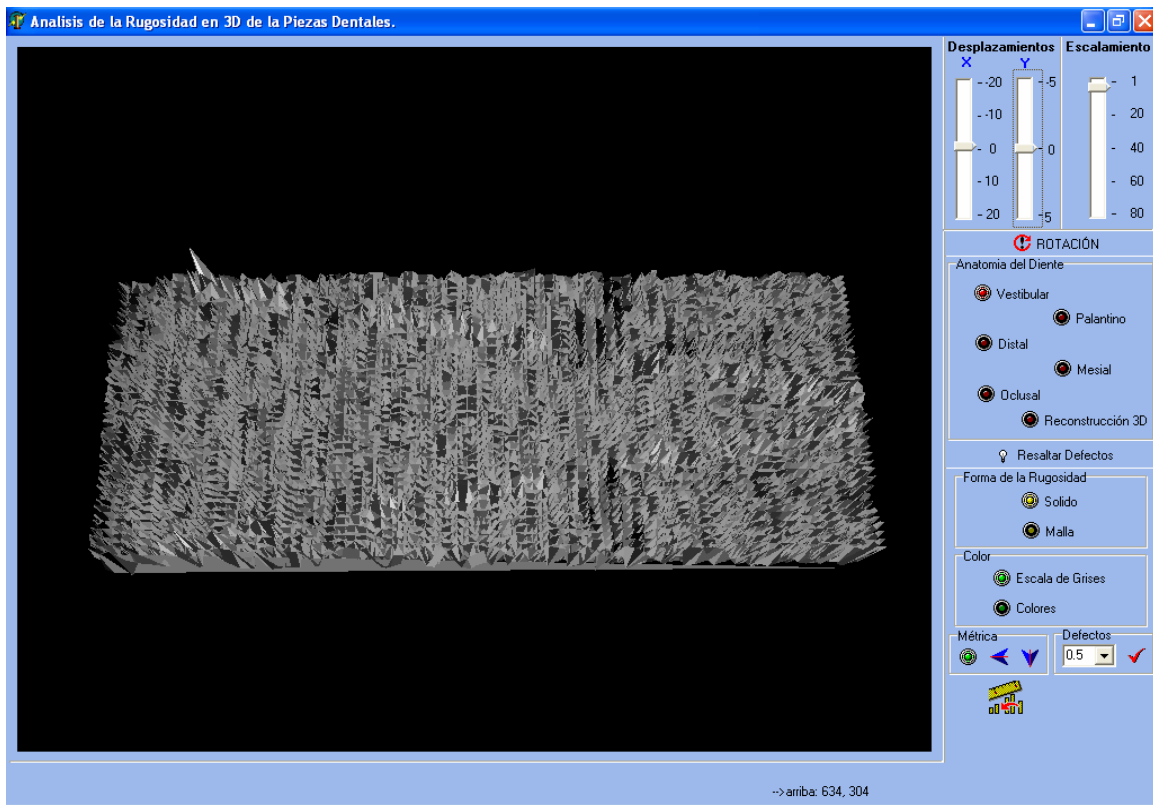


Figura 54. Imagen en 3 dimensiones de la rugosidad de una mica en escala de grises

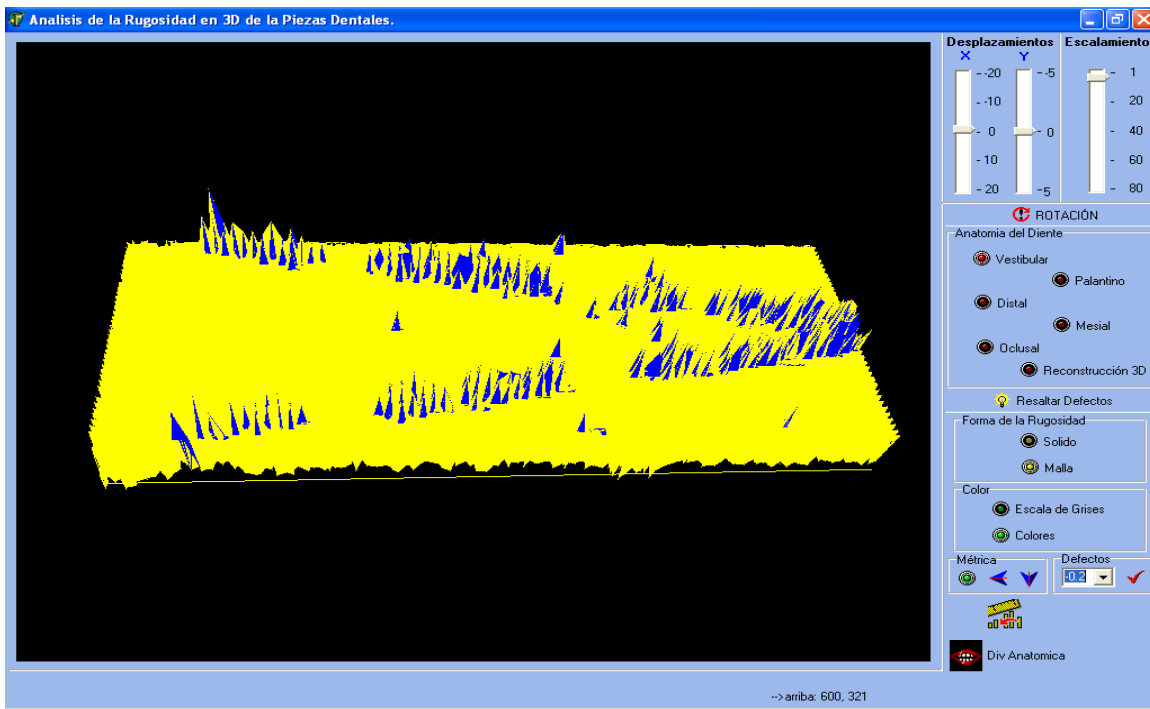


Figura 55. Imagen en 3 dimensiones de la rugosidad de una mica a color y con resalte del defecto

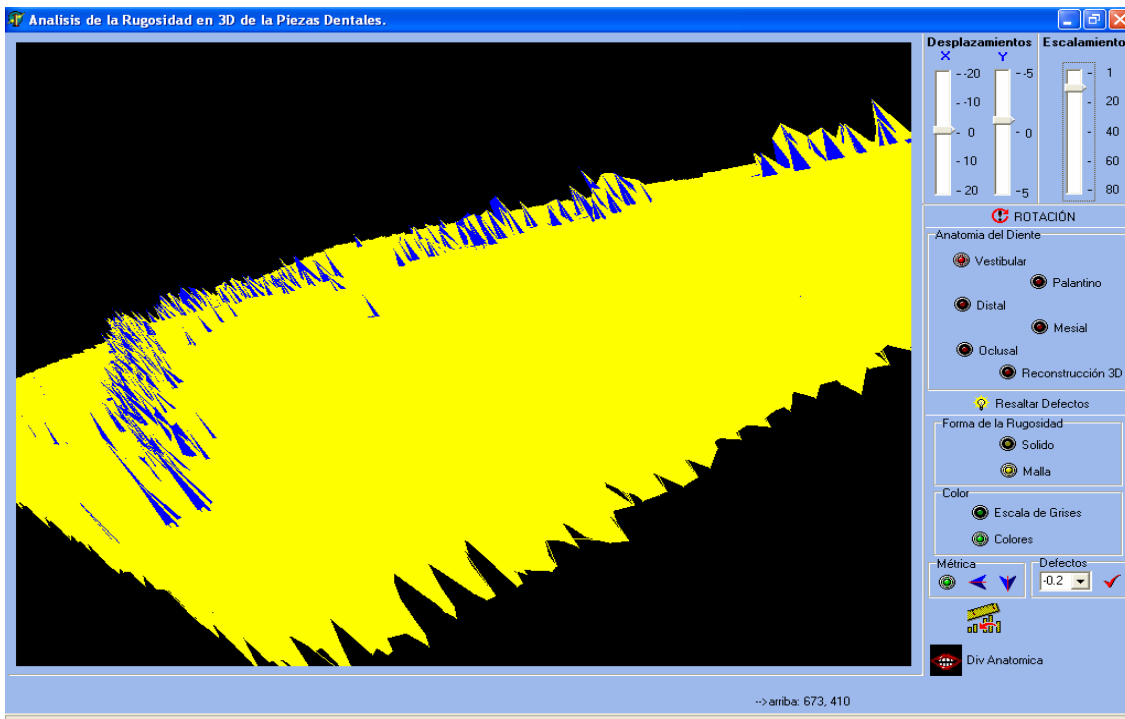


Figura 56. Imagen en 3 dimensiones de la rugosidad de una mica a color, con resalte de defecto, tomada desde otro ángulo

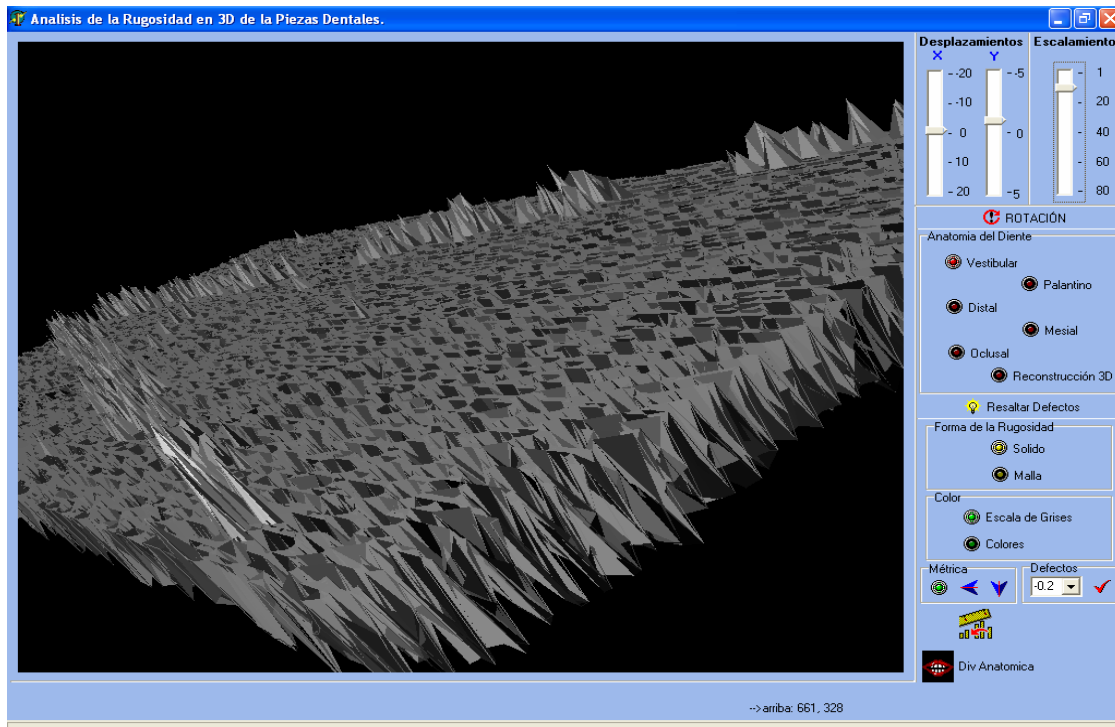


Figura 57. Imagen en 3 dimensiones de la rugosidad de una mica a escala de grises, con resalte de defecto



## CONCLUSIONES

Se desarrolló e implementó un sistema para analizar la rugosidad de la superficie del esmalte dental en forma de imagen en 2 D y en animación en 3 D, que permite al especialista en estomatología observar las piezas dentales sin defecto y con defecto en una computadora además rotar, escalar y trasladar, la imagen.

El sistema es capaz de detectar zonas de desmineralización de 50 micras en piezas dentales tomando en cuenta su geometría, además se pueden detectar defectos en materiales dieléctricos, en particular: la mica.



## **APORTACIONES**

Los métodos invasivos para el diagnóstico estomatológico son utilizados por los especialistas para encontrar defectos en las piezas dentales, muchos de estos métodos perjudican a los pacientes y con el tiempo provocan enfermedades. La aportación de este trabajo es el desarrollo e implementación de una metodología para analizar la rugosidad del esmalte dental, por lo tanto el especialista en estomatología puede observar la rugosidad de la superficies de las piezas dentales en tres dimensiones y pueda ver los defectos que puede tener la pieza dental, además la imagen la puede rotar, trasladar y escalar la pieza dental para tener mejores ángulos de observación sobre la pieza dental.



## PERSPECTIVAS

Este trabajo permitió el desarrollo de un nuevo sistema para el análisis de la rugosidad de la superficie del esmalte dental.

Para continuar con el trabajo se requiere desarrollar lo siguiente:

- 1.- detectar defectos internos en las piezas dentales mediante un sistema tomográfico.
- 2.- crear formas anatómicas propias de las piezas dentales como son fisuras y fosetas.
- 3.- mejoras en la navegación de la superficie.
- 4.- crear un sistema de reconocimiento patrones para la detección de defectos en la superficie del esmalte dental como puede ser odontogénesis imperfecta, hipoplasia, hipocalcificación y fluorosis.



## BIBLIOGRAFÍA

- [1] *Gómez de Ferraris, Campos Muños.* Histología y Embriología Bucodental. Editorial Panamericana, 4ta reimpression, Marzo 2000.
- [2] *Orban. S. N. Bhaskar.* Histología y Embriología Bucal, Editorial, Prado, 8va edición, 2000.
- [3] *Ruth Ashley, Dr. Harry R.* Anatomía y Terminología Dental, Editorial Limusa, México 1979.
- [4] *A. I. Potapov, C. S. Morokina, J. I. Cortez.* Control de la Dosis de Radiación Debido a la Exposición de Rayos X en los Órganos Dentarios. Revista Control no Invasivo y Diagnostico del Medio Ambiente y Materiales. Vol. 2, pp. 137-143, 2001. (UDK620.179.18) (en Ruso).
- [5] [http://www.dentalcolombia.com/docs/general/revision\\_laser.htm](http://www.dentalcolombia.com/docs/general/revision_laser.htm)
- [6] [http://www.camaraintraoral.com/submenu\\_limpieza\\_ref\\_esp.htm](http://www.camaraintraoral.com/submenu_limpieza_ref_esp.htm)
- [7] *Javier Caldera Miguel.* Identificación de cambios físicos en el esmalte dental utilizando el microcontrolador 8032, Tesis de Licenciatura, BUAP, 2003.
- [8] *Jesús Alberto Ramírez Popo,* ,Tesis de Maestría, BUAP, 2005
- [9] *J. I. Cortez, D. M. Bustillo, A. Grebennikov.* On the scheme of infrared reflection measurements and regularization data processing algorithm, Wseas Transactions on Computers, Issue 3, Volume 2, July 2003, ISSN 1109-2750.
- [10] *Gonzalo Pajares, Jesús M. de la Cruz.* Visión por Computador Imágenes Digitales y Aplicaciones. Editorial Alfaomega\_Ra-Ma, ISBN 970-15-0804-1. Madrid España.
- [11] *A. Myers.* Introductory literature review surface reconstruction from three Dimensional Range Data. April 1999.
- [12] *B. Curless.* New methods for surface reconstruction from range images. StanfordUniversity Stanford, 1997.
- [13] *R. Baribeau.* Colour three-dimensional modeling of museum objects.1996



- [14] *P. Boulanger*. Range image integration for direct replication of objects. 1996.
- [15] *T. Akimoto*. Automatic creation of 3d facial models. IEEE Computer Graphics Applications, 13. Pp16-22 1993.
- [16] *J. I. Cortez*. Métodos y dispositivos para la defectoscopia óptica infrarroja en materiales dieléctricos, Tesis Doctoral, San Petersburgo RUSIA, p-178, 2000.
- [14] <http://www.charrua.com/vista/sp/downloads/SUIS.ppt>
- [15] *R. Corrigan*. Scanned linear architecture improves laser projector. Laser Focus, World. 1998.
- [16] *J. I. Cortez, D. M. Bustillo, L. Cortez, M. F. Gonzáles, J. M. Caldera*. Acquisition and data with apdsed system using statistical process for the detection of defects in the surface of the dental enamel , Wseas Transactions on Biology and Biomedicine , Issue 4, Volume 1, October 2004, ISSN 1109-9518.

# APÉNDICE A

**CÓDIGOS PRINCIPALES DEL SOFTWARE DE  
REPRESENTACIÓN DE DATOS.**



## Procedimiento de Interpolación

```

*****
Procedure TForm1.Interpolación();
var
    i,k           :integer;
    dato,dato1,dato2,dato3 :real;
    med,med1      :real;
    cont,dato     :integer;
begin
*****Interpola Columnas*****
    cont:=0;
    for k:=1 to Columnas1 do
        begin
            for i:=1 to Filas1 do
                begin
                    dato1:= Cara1[i,k];
                    dato2:= Cara1[i+1,k];
                    med :=((dato1 + dato2)/2);
                    if i=1 then
                        begin
                            TransMat[i,k]:=dato1;
                            TransMat[i+1,k]:=med;
                        end
                    else
                        begin
                            TransMat[i+cont,k]:=dato1;
                            TransMat[i+cont+1,k]:=med;
                        end;
                    cont:=cont+1
                end;
            end;
        end;
    end;
*****Interpola Filas*****
    cont:=0;
    for i:=1 to Filas1 do
        begin
            for k:=1 to Columnas1 do
                begin
                    dato1:= Cara1[i,k];
                    dato2:= Cara1[i,k+1];
                    med :=((dato1 + dato2)/2);
                    if k=1 then
                        begin
                            TransMat[(i*2)-1,k]:=dato1;
                            TransMat[(i*2)-1,k+1]:=med;
                        end
                    else
                        begin
                            TransMat[(i*2)-1,k+cont]:=dato1;
                            TransMat[(i*2)-1,k+cont+1]:=med;
                        end;
                    cont:=cont+1
                end;
            end;
        end;
    end;
*****
end;

```



## Reconstrucción de la Rugosidad por Medio de Métricas

```

*****
procedure TForm2.Rugosidad_cara1();
var
  i,k,l,s,t,pas,pat           : integer;
  valor,valor1,valor2        : real;
  met,metrica,tem,c          : real;
  met1,met2,met3,met4        : real;
  r,g,b                       : real;
  r1,g1,b1                   : byte;
  gris                        : real;
  gris1                       : byte;
  Pixel                       : integer;
  punto1,punto2,punto3,punto4 : real;
  punto5,punto6,punto7,punto8 : real;
begin
  *****Pinta los puntos de la Rugosidad*****
  for k:=1 to Filas1 do//Form1.SG.RowCount do
  begin
    glBegin(GL_POINTS); // QUADS LINES LINE_STRIP
    for i:=1 to Columnas1 do//Form1.SG.ColCount do
    begin
      valor:= Cara1[k,i];
      if valor <> 0 then
      begin
        c:= (Valor * 255) / (3.5);
        r := $FF and Round(c) ;
        g := $FF and (Round(c) shr 8);
        b := $FF and (Round(c) shr 16);
        Gris := ((r * 0.3) + (g*0.59) + (b*0.11));
        GLPointSize(5.0);
        GLColor3f(Gris,Gris,Gris);
        GLVertex3f(-i/10,-k/10,valor);
      end;
    end;
  glEnd;
end;
  *****Pintamos el contorno de Arriba*****
  for k:=1 to 1 do
  begin
    glBegin(GL_QUADS); //GL_POINTS QUADS LINES LINE_STRIP
    for i:=1 to Columnas1 do//Form1.SG.ColCount do
    begin
      valor:= Cara1[k,i];
      if valor <> 0 then
      begin
        c:= (valor * 255)/ 5.0;
        r1 := $FF and Trunc(c) ;
        g1 := $FF and (Trunc(c) shr 8);
        b1 := $FF and (Trunc(c) shr 16);
        Gris1 := (r1 + g1 + b1);
        glColor3ub(Gris1,Gris1,Gris1);
        GLVertex3f(-i/10,-k/10,valor);
      end;
    end;
  glEnd;
end;
  *****Pintamos el contorno de Abajo*****
  *****Primero buscamos los ultimos datos de cada columna*****
  for k:=1 to Columnas1 do
  begin
    for i:=1 to Filas1 do//Form1.SG.ColCount do
    begin
      valor:= Cara1[i,k];
      valor1:=Cara1[i+1,k];
      if (valor<>0)and(valor1=0) then
      begin
        ultimo[k]:= valor;
        Cordei[k]:=i;
        Cordek[k]:=k;
      end;
    end;
  end;
end;

```



```

end;

end;
end;
*****Ahora si trazamos las líneas*****
glbegin(GL_QUADS);
for l:=1 to Columnas1 do
begin
valor:= Ultimo[l];
k:=Cordei[l];
i:=Cordek[l];
if valor <> 0 then
begin
c := (valor * 255)/ 5.0;
r1 := $FF and Trunc(c) ;
g1 := $FF and (Trunc(c) shr 8);
b1 := $FF and (Trunc(c) shr 16);
Gris1 := (r1 + g1 + b1);
glColor3ub(Gris1,Gris1,Gris1);
GLVertex3f(-i/10,-k/10,valor);
end;
end;
glEnd;
*****Pone el contorno Izquierdo*****
for i:=1 to 1 do//Form1.SG.ColCount do
begin
glbegin(GL_QUADS); //GL_POINTS QUADS LINES LINE_STRIP
for k:=1 to Filas1 do//Form1.SG.RowCount do
begin
valor:= Cara1[k,i];
if valor <> 0 then
begin
c:= (valor * 255)/ 5.0;
r1 := $FF and Trunc(c) ;
g1 := $FF and (Trunc(c) shr 8);
b1 := $FF and (Trunc(c) shr 16);
Gris1 := (r1 + g1 + b1);
glColor3ub(Gris1,Gris1,Gris1);
GLVertex3f(-i/10,-k/10,valor);
end;
end;
glEnd;
end;
*****Pone el Contorno Derecho*****
for i:=Columnas1-1 to Columnas1-1 do//Form1.SG.ColCount do
begin
glbegin(GL_QUADS); //GL_POINTS QUADS LINES LINE_STRIP
for k:=1 to Filas1 do//Form1.SG.RowCount do
begin
valor:= Cara1[k,i];
if valor <> 0 then
begin
c:= (valor * 255)/ 5.0;
r1 := $FF and Trunc(c) ;
g1 := $FF and (Trunc(c) shr 8);
b1 := $FF and (Trunc(c) shr 16);
Gris1 := (r1 + g1 + b1);
glColor3ub(Gris1,Gris1,Gris1);
GLVertex3f(-i/10,-k/10,valor);
end;
end;
end;
glEnd;
end;
*****Terminamos de Pintar los contornos*****
*****Obtenemos la Rugosidad*****
for k:=2 to Filas1-1 do//Form1.SG.RowCount do
begin
for i:=2 to Columnas1-1 do//Form1.SG.ColCount do
begin
valor:= Cara1[k,i];
if valor <> 0 then
begin

```



```

//*****Obtenemos los 8 Puntos Vecinos al principal*****//
punto1:=Cara1[k,i-1];
punto2:=Cara1[k,i+1];
punto3:=Cara1[k-1,i];
punto4:=Cara1[k+1,i];
punto5:=Cara1[k-1,i-1];
punto6:=Cara1[k-1,i+1];
punto7:=Cara1[k+1,i-1];
punto8:=Cara1[k+1,i+1];
*****Obtenemos el punto que nos indica cuales se conectan*****
metricas[1]:=Abs(valor-punto1);
metricas[2]:=Abs(valor-punto2);
metricas[3]:=Abs(valor-punto3);
metricas[4]:=Abs(valor-punto4);
metricas[5]:=Abs(valor-punto5);
metricas[6]:=Abs(valor-punto6);
metricas[7]:=Abs(valor-punto7);
metricas[8]:=Abs(valor-punto8);
//*****Ordenamos el Arreglo*****//
pas:=8;
pat:=8;
for t:=1 to pas do
begin
for s:=1 to pat do
begin
if metricas[s] > Metricas[t] then
begin
tem:=metricas[s];
metricas[s]:=metricas[t];
metricas[t]:=tem;
end;
end;
end;
Metrica:=Metricas[4];
valor1:= valor+Metrica;
valor2:=valor-metrica;
glbegin(GL_QUADS); //GL_POINTS LINES LINE_STRIP
*****Damos el color*****
c:= (valor * 255)/ 5.0;
r1 := $FF and Trunc(c) ;
g1 := $FF and (Trunc(c) shr 8);
b1 := $FF and (Trunc(c) shr 16);
Gris1 := (r1 + g1 + b1);
glColor3ub(Gris1,Gris1,Gris1);
*****Trazamos las lineas Hozontales y verticales*****
if (punto1 <= valor1) and (Punto1 >= valor2)and(punto1<>0) then
begin
//GLColor3f(0,0,1);
GLVertex3f(-i/10,-k/10,valor);
GLVertex3f((-1*(i-1))/10,-k/10,Punto1);
end;
if (punto2 <= valor1) and (Punto2 >= valor2)and(punto2<>0) then
begin
//GLColor3f(0,1,0);
GLVertex3f(-i/10,-k/10,valor);
GLVertex3f((-1*(i+1))/10,-k/10,Punto2);
end;
if (punto3 <= valor1) and (Punto3 >= valor2)and(punto3<>0) then
begin
//GLColor3f(0,1,0);
GLVertex3f(-i/10,-k/10,valor);
GLVertex3f(-i/10,(-1*(k-1))/10,Punto3);
end;
if (punto4 <= valor1) and (Punto4 >= valor2)and(punto4<>0) then
begin
//GLColor3f(0,1,1);
GLVertex3f(-i/10,-k/10,valor);
GLVertex3f(-i/10,(-1*(k+1))/10,Punto4);
end;
*****Trazamos la lineas Diagonales*****
if (punto5 <= valor1) and (Punto5 >= valor2)and(punto5<>0) then
begin
//GLColor3f(1,0,0);
GLVertex3f(-i/10,-k/10,valor);

```



```

    GLVertex3f((-1*(i-1))/10,(-1*(k-1))/10,Punto5);
end;
if (punto6 <= valor1) and (Punto6 >= valor2)and(punto6<>0) then
begin
    //GLColor3f(1,0,1);
    GLVertex3f(-i/10,-k/10,valor);
    GLVertex3f((-1*(i+1))/10,(-1*(k-1))/10,Punto6);
end;
if (punto7 <= valor1) and (Punto7 >= valor2)and(punto7<>0) then
begin
    //GLColor3f(1,1,0);
    GLVertex3f(-i/10,-k/10,valor);
    GLVertex3f((-1*(i-1))/10,(-1*(k+1))/10,Punto7);
end;
if (punto8 <= valor1) and (Punto8 >= valor2)and(punto8<>0) then
begin
    //GLColor3f(1,1,1);
    GLVertex3f(-i/10,-k/10,valor);
    GLVertex3f((-1*(i+1))/10,(-1*(k+1))/10,Punto8);
end;
glEnd;
end;
end;
//edit1.Text:=FloatToStr(metrica);
end;
*****

```



## Reconstrucción de la Pieza Dental en 3 Dimensiones

```

*****
procedure TForm2.Reconstrucción_Pieza_Dental();
var
    i,k      : integer;
    valor    : real;
    c        : real;
    r,g,b    : integer;
    gris     : real;
    Pixel    : integer;
Begin
*****Una Cara*****
GIPushMatrix();
  GIPushMatrix();
    glTranslatef(-1.0,-1.0,1);
    glscalef(0,0,1);
    cubo1();
  GIPopMatrix();
*****Segunda Cara*****
  GIPushMatrix();
    glRotatef(0.0,0.0,0.0,180);
    glRotatef(0,180,0,0);
    glTranslatef(-1.0,-1.0,5);
    Cubo2();
  GIPopMatrix();
*****Tercera Cara*****
  GIPushMatrix();
    glTranslatef(-1.0,-1.0,1);
    //glRotatef(0,45,1,0);
    //glRotatef(1,45,0,0);
    glRotatef(90,0.0,1.0,0.0);
    glRotatef(10,1.0,0.0,0.0);
    glTranslatef(-1.0,-1.3,0);
    cubo3();
  GIPopMatrix();
*****Cuarta Cara*****
  GIPushMatrix();
    glTranslatef(-2.0,-1.0,2);
    glRotatef(0,90,0,0);
    cubo4();
  GIPopMatrix();
GIPopMatrix();
end;
*****

```

# APÉNDICE B

## **DIAGRAMA PRINCIPAL DEL SOFTWARE DE REPRESENTACIÓN DE DATOS**

# APÉNDICE C

## **OPENGL EN DELPHI**



OpenGL es una librería gráfica escrita originalmente en C que permite la manipulación de gráficos 3D a todos los niveles. Esta librería se concibió para programar en máquinas nativas Silicon Graphics bajo el nombre de GL (Graphics Library). Posteriormente se consideró la posibilidad de extenderla a cualquier tipo de plataforma y asegurar así su portabilidad y extensibilidad de uso con lo que se llegó al término Open Graphics Library, es decir, OpenGL. La librería se ejecuta a la par con nuestro programa independientemente de la capacidad gráfica de la máquina que usamos. Esto significa que la ejecución se dará por software a no ser que contemos con hardware gráfico específico en nuestra máquina. Si contamos con tarjetas aceleradoras de vídeo, tecnología MMX, aceleradoras 3D, pipelines gráficos implementados en placa, etc. gozaremos por supuesto de una ejecución muchísimo más rápida en tiempo real. Así esta librería puede usarse bajo todo tipo de sistemas operativos e incluso usando una gran variedad de lenguajes de programación. Podemos encontrar variantes de OpenGL para Windows 95/NT, Unix, Linux, Iris, Solaris, Java e incluso lenguajes de programación visuales como Visual Basic, Borland C++ Builder y por supuesto Delphi. En contraste con la antigua IRIS GL-library de SGI, OpenGL es por diseño independiente de plataformas y sistemas operativos como ya lo mencionamos.

Por encima de todo, OpenGL es una biblioteca estilizada de trazado de gráficos de alto rendimiento, y hay varias tarjetas gráficas aceleradoras y especializadas en 3D que implementan primitivas OpenGL a nivel del hardware. Hasta hace poco, estas avanzadas bibliotecas gráficas solían ser muy caras y sólo estaban disponibles para estaciones SGI u otras estaciones de trabajo UNIX. Las cosas están cambiando muy de prisa y gracias a las generosas licencias y el kit de desarrollo de controladores de SGI, vamos a ver más y más hardware OpenGL para usuarios de computadoras. Al contrario de Direct3D, OpenGL es un API altamente portable y sencilla. Lo de sencilla es más bien en comparación con Direct3D, ya que para poder comprender la mayor parte de las funciones de OpenGL es necesario tener un poco de conocimientos de álgebra lineal, geometría y un poco de física, aunque a final de cuentas no es tan difícil como puede parecer.



OpenGL provee prácticamente las mismas funcionalidades en cuanto a capacidades gráficas que Direct3D; incluso en algunos aspectos se comporta de una manera más eficiente, y posee una serie de características que facilitan al programador la tarea de construir la escena, como tal es el caso de las Listas de Despliegue, que son una manera de almacenar comandos de dibujo en una lista para un trazado posterior. Los Programadores de Delphi tenemos un amplio panorama en frente nuestro respecto a OpenGL, ya que de por sí las versiones de Delphi de 32 bits incluyen la DLL OpenGL32, que es la versión de OpenGL para Win32, además incluye una unidad llamada OpenGL.pas la cual es la interfaz entre la DLL y nuestras propias aplicaciones. Incluso algunas versiones de Delphi traen archivos de ayuda para esta API y sus múltiples funciones. Debemos mencionar que existen muchos programadores de Delphi que han hecho valiosas aportaciones al mundo de los gráficos, tales como: Mitchell E. James con su componente GLPanel, ó Mike Lischke, el cual es un prominente miembro del grupo JEDI (Joint Endeavor of Delphi Innovators, Grupo de Esfuerzos de Innovadores de Delphi), y que también ha escrito varias librerías para OpenGL en Delphi. Esto, por mencionar solo a algunos, a fin de cuentas existen muchos programadores reconocidos en este ámbito.

OpenGL funciona a través de una serie de librerías DLL que suelen tener variados nombres, pero que a fin de cuentas cubren funciones muy específicas y son muy sencillas de identificar, ahora veremos cuales son estas: *OpenGL.DLL (Open Graphics Library, Librería de Gráficos Abierta)*, ésta podemos encontrarla también con el nombre de OpenGL32.DLL (para el caso de los sistemas operativos de 32 bits) o bien simplemente como GL.DLL. Esta es la librería principal, que contiene la mayoría de las funciones que aquí se utilizan. Las funciones contenidas en esta librería inician con las letras gl.

*GLU.DLL (Graphics Utility Library, Librería de Utilerías de Gráficos)*, también la podemos encontraremos como GLU32.DLL. Contiene funciones para objetos comunes a dibujar como esferas, donas, cilindros, cubos, en fin, varias figuras ya predefinidas que pueden llegar a ser útiles en ciertos casos, así como funciones para el manejo de la cámara entre muchas otras. Se puede



identificar las funciones que pertenecen a esta librería porque llevan antepuestas en su nombre las siglas glu.

*GLUT.DLL (GL Utility Toolkit, Equipo de Herramientas de Utilería para el desarrollo de Gráficos).*- Esta también permite crear objetos complejos como GLU, aunque la principal función de esta librería es permitir que los programas se vuelvan interactivos, o sea que posibilita la libre creación de ventanas, así como el acceso al ratón y al teclado. Podríamos llegar a prescindir en ese aspecto de GLUT, ya que Delphi nos ofrece de manera natural crear ventanas y responder a los diferentes eventos del manejo del ratón y el teclado; sin embargo, muchos programadores (principalmente programadores de C y C++) piensan que programar aplicaciones basadas en GLUT tiene muchísimas ventajas, ya que ofrece independencia del sistema operativo, pues no es lo mismo crear una ventana basándose en las API's de Windows que crear una ventana en algún ambiente gráfico de Unix o Linux, por esto, si se usa GLUT como manejador de ventanas, no se tendría que reescribir el código para migrar la aplicación de plataforma, ya que sólo habría que conseguir la versión de GLUT para el ambiente gráfico deseado y volver a compilar nuestro programa en el nuevo ambiente. Otro componente es el llamado Frame Buffer, que no es otra cosa que el área de memoria donde se construyen los gráficos antes de mostrarlos al usuario, es decir, que nuestro programa de OpenGL escribe en esta área de memoria, y automáticamente envía su contenido a la pantalla una vez que la escena está completamente construida. La figura 1 muestra gráficamente como esta constituida la estructura de abstracción de OpenGL.

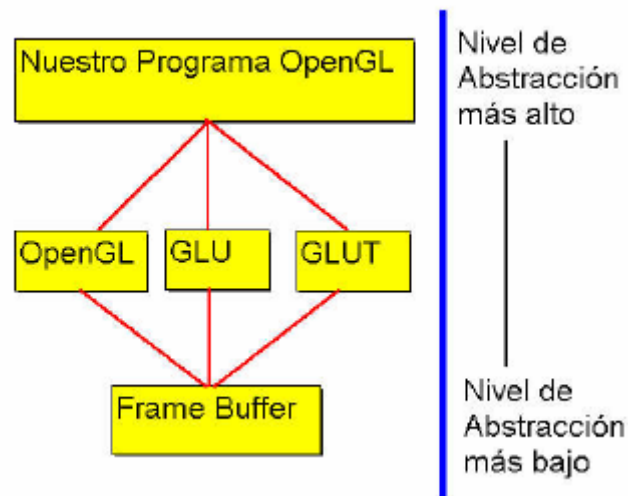


Figura 1. Niveles de Abstracción en OpenGL

OpenGL está basada en los conceptos de Matrices y Vectores y las operaciones aritméticas aplicables a estas estructuras.

En OpenGL existen básicamente tres matrices principales: Una matriz de proyección llamada `GL_PROJECTION`, la cual permite determinar la perspectiva a usar para observar la escena generada, así como el tipo de proyección a usar (esto tiene que ver mucho con cuestiones de óptica y de geometría, y abordaremos este tema en otra ocasión más específicamente, ya que es bastante extenso y complejo). Esta matriz tiene una gran relación con otro concepto muy interesante llamado clipping, que consiste en recortar u ocultar todo aquello que “está pero no se ve”, es decir lo que queda fuera del foco de nuestra cámara o nuestro campo visual activo, este es un proceso que se hace automáticamente una vez, habiendo inicializado pertinentemente esta matriz. Una matriz de Modelado llamada `GL_MODELVIEW`, donde ésta es la matriz que se usará para aplicar a la escena operaciones de rotación, traslación o escalamiento, o bien para manipular la posición y orientación de la cámara, para obtener así las animaciones. La figura 2 muestra algunos ejemplos de matrices de transformación lineal para el caso de sistemas de 3 dimensiones.

$$\begin{array}{c}
 \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Traslación} \\
 \\
 \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Escalamiento} \\
 \\
 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotación en X} \quad \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotación en Y} \quad \begin{bmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotación en Z}
 \end{array}$$

Figura 2. Matrices genéricas de transformación lineal para sistemas de 3 dimensiones.

Y una última matriz para el manejo de Texturas llamada `GL_TEXTURE`, sobre la cual también puede aplicarse las transformaciones lineales de rotación, traslación y escalamiento para manipular las texturas a utilizar en las figuras de nuestra escena, cuando estemos más avanzados explicaremos mas a detalle el manejo de texturas en nuestros programas, pero debemos aprender a gatear antes de intentar correr. Podemos hacer uso de cada una de estas matrices mediante el procedimiento `GLMatrixMode()`, el cual permite seleccionar una de estas matrices para su configuración. Para inicializar con valores cada matriz se invoca a un procedimiento llamado `GLLoadIdentity()`, el cual carga la matriz identidad, que es aquella que solo contiene valores de 1 en toda su diagonal (como se muestra en la figura 3), lo cual hace que multiplicar cualquier vector por esta matriz nos dé como resultado al mismo vector sin que haya sufrido ninguna transformación.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 3. Matriz identidad



En la figura 4 se observa el orden en que se aplican las matrices a cada uno de los vectores, y como se transforman las coordenadas hasta llegar a las coordenadas reales de la ventana.

Vectores de diversos tipos: los vectores son un conjunto de valores que permiten definir dentro de nuestra escena, desde los vértices de las figuras, hasta los colores, los materiales y las luces, entre otras muchas cosas. Existen básicamente dos formas de trabajar con vectores en OpenGL, con sus elementos como variables independientes, o manejarlos como una estructura de datos. Cuando trabajamos sus elementos de forma independiente cada vector es descompuesto en 3 ó 4 valores según sea el caso. Y cuando se maneja como una estructura estos valores están contenidos en una estructura de datos que bien puede ser un arreglo o un registro.

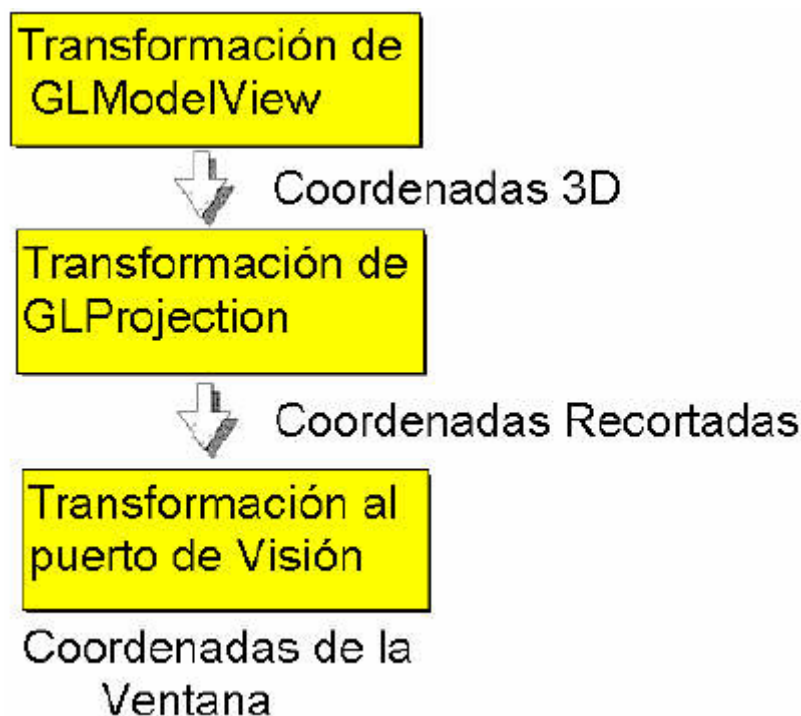


Figura 4. Secuencia de Transformación de coordenadas.

Decimos que los vectores pueden descomponerse en 3 o 4 valores, ya que en el caso de los colores éstos están compuestos por tres valores que determinan el color, codificados en RGB, y un elemento extra en algunos casos, que determina el nivel de transparencia para el objeto, cara ó vértice al



cual corresponda dicho color, a esta característica de transparencia se le conoce comúnmente como Alpha Blending. Todos estos valores (en el caso específico de los colores) tienen un dominio que fluctúa en el rango de 0 a 1.

Los vectores están compuestos por números que pueden ser de tipo real, enteros, bytes, doubles, short, etc... Aquí cabe señalar que OpenGL provee de sus propios tipos, los cuales por convencionalismos similares al que usamos en Delphi, anteponiendo una T a las clases que usamos, en OpenGL se antepone GL a la definición de Tipos que se utiliza, así porejemplo GLFloat, es el equivalente del tipo Single en Delphi, y así existen muchas equivalencias entre ambas topologías. Para declarar un vector como un vértice descomponiendo sus elementos usaríamos la instrucción GLVertex3f(X,Y,Z), donde X, Y y Z son variables o constantes de tipo GLFloat, Real, o Single. La parte final de la instrucción que usamos: 3f, indica que el procedimiento recibe como parámetro tres valores de punto flotante. Así la instrucción GLVertex3i() funcionará exactamente igual solo que con valores enteros.

Cuando se trabajan los vectores como una estructura de datos lo que se hace es pasar la dirección de memoria de esta estructura en la llamada del procedimiento, usando el operador @. Por ejemplo: Si suponemos una variable V de tipo Tvector, donde Tvector está definido como: Tvector = Array [0..2] of GLFloat; podríamos usar la instrucción: GLVertex3fv(@V); para declarar el vértice a partir de este dato estructurado. Observen como en estos casos las instrucciones para manejar estructuras terminan en: v.

Acontinuación se muestra un ejemplo donde se explicara cada línea de código:

```
Unit Unit1;
interface
uses
Windows, Forms, OpenGL, Classes, ExtCtrls, Messages, Controls, StdCtrls,Graphics,
sysUtils, Dialogs;
type
TForm1 = class(TForm)
Timer1: TTimer;
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure FormPaint(Sender: TObject);
procedure FormResize(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
{ Private declarations }
procedure WMEraseBkgnd(var Message: TWMEraseBkgnd); message
WM_ERASEBKGND;
```



```

public
{ Public declarations }
end;
var
Form1: TForm1;
RC : HGLRC;
Angulo : GLInt;
implementation
{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
    RC:=CreateRenderingContext(canvas.Handle,[opDoubleBuffered],32,0); //Primero
    Creamos un contexto...
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
    DestroyRenderingContext(RC); //Se libera el Contexto...
end;

procedure TForm1.FormPaint(Sender: TObject);
Var X,Y,Z:GLInt;
Begin
    ActivateRenderingContext(canvas.Handle,RC); // Se asocia el contexto con el
                                                manejador del Canvas
    glClearColor(0,0.2,0,0); // Ponemos un color Verde Oscuro de Fondo...
    glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT); // Algo así como
                                                            borrar la Pizarra...

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity;
    gluLookAt(0,0,3,0,0,0,1,0); // Posición y punto de vista de la cámara...
    glRotatef(30,1,0,0); //Primero hacemos una rotación de 30 grados respecto a X para
                        obtener perspectiva...
    glRotatef(Angulo,0,1,0); //Se Rota la figura en el eje Y a partir de la variable Angulo...
    GLPointSize(2.0); //Asignamos un tamaño de 2 pixeles para cada punto...
    //Ahora con tres ciclos anidados dibujamos un cubo con puntos de colores variados...
    GLBegin(GL_POINTS);
        for x := -5 to 5 do
            for y := -5 to 5 do
                for z := -5 to 5 do
                    begin
                        GLColor3f(X,Y,Z);
                        GLVertex3f(X/10,Y/10,Z/10);
                    end;
                end;
            end;
        end;
    GLEnd;
    SwapBuffers(canvas.Handle); //Copiar el Back Buffer en el canvas del formulario...
    DeactivateRenderingContext; //Libera el contexto...
end;

procedure TForm1.FormResize(Sender: TObject);
begin // Cuando se cambia de tamaño hay que actualizar el puerto de visión...
    wglMakeCurrent(canvas.handle,RC); // Otra manera de hacer el contexto dibujable
                                        cuando este ya está creado...
    glViewport(0,0,Width,Height); // Especificar un puerto de visión....
    glMatrixMode(GL_PROJECTION); // Activar matriz de proyección...
    glLoadIdentity; // Poner estado inicial en esta matriz...
    gluPerspective(35,Width/Height,1,100); // Especificar Perspectiva ...
    wglMakeCurrent(0,0); // Otra manera de liberar el contexto...
end;

```



```

Refresh; // Redibujar la escena ...
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    Inc(Angulo,4); //Rotamos el angulo de observación de la escena...
    Angulo := Angulo mod 360;
    Refresh; //y la volvemos a dibujar ...
end;

procedure TForm1.WMEraseBkgnd(var Message: TWMEraseBkgnd);
begin //Para borrar el fondo, y evitar el parpadeo ...
    Message.Result:=1;
end;

end.

```

Como se puede observar en el listado, tenemos un formulario y un componente Timer, el formulario servirá como la ventana en donde se mostrarán los gráficos; y el Timer será quién coordine nuestra animación, en esta ocasión a fin de hacer las cosas más sencillas usaremos el Timer para esta finalidad. Primeramente, se muestra un *lienzo* en el cual se plasmarán las “*obras de arte*”, por lo cual necesitamos de procesos que permitan acceder directamente a algún objeto gráfico que nos sirva como *lienzo*. En este ejemplo usamos el Objeto Canvas de nuestro formulario, pero en teoría podemos utilizar el Canvas de cualquier otra clase que contenga a éste objeto (un *PaintBox* por ejemplo). A este *lienzo* le llamaremos el *contexto*.

Como puede darse cuenta, una variable global llamada RC de tipo HGLRC, será la que represente nuestro contexto dibujable. En el evento OnCreate del formulario hemos escrito una línea de código que sirve para asociar nuestro contexto con el manejador (*handle*) del objeto Canvas del formulario, lo cual hace que nuestras animaciones se desplieguen sobre este Canvas. Del mismo modo en el evento OnDestroy del formulario debemos escribir una línea de código que libere nuestro contexto del Canvas, para evitar así *apuntadores perdidos*. Ahora, las animaciones deben construirse escena por escena, como en las películas de dibujos animados, con ligeras diferencias entre ellas para obtener calidad en nuestra animación, y a la vez debemos hacer uso de algún evento para mostrar estas escenas; así que podemos utilizar el evento OnPaint del formulario para mostrarlas una por una. Como



sabemos este evento se dispara cada vez que Windows tiene que dibujar el formulario, así que, podemos ocuparlo para que haga en él lo que deseamos que se vea. En este evento es donde escribimos el código para generar las escenas que nos interesa mostrar, vemos como con cada *frame* o escena, hacemos el contexto dibujable y al finalizar lo liberamos de nuevo. La instrucción `glClearColor()` recibe como parámetro un vector de 4 valores que determinarán el color con el que se inicializará nuestro *frame buffer* antes de escribir en él, dando como resultado un color de fondo. La instrucción `GLClear()` sirve para borrar de este *frame buffer* valores de escenas que hayan sido construidas previamente a la actual. Además, puede hacerse uso de la matriz `ModelView`, y su inicialización con `GLLoadIdentity()`, para poder aplicar las operaciones de rotación con las operaciones `GLRotate()` a los vértices que luego se definen. La operación `gluLookAt()` sirve para determinar un punto de observación (ó cámara) para la escena, podríamos obtener el mismo efecto de rotación si en vez de transformar los vértices de la escena solo movemos este punto de observación, es algo similar a lo que se hace en los clásicos juegos 3D de estilo *Doom*. Esta animación la construiremos utilizando solo puntos aislados, por lo que usamos la operación `GLPointSize()` para determinar el número de píxeles reales que ocupará cada uno de nuestros puntos, en nuestro caso 2. Y luego entre las declaraciones `GLBegin()` y `GLEnd()` que sirven para determinar los vértices de una figura, llamamos a `GLColor3f()` y `GLVertex3f()` dentro de tres ciclos anidados para dibujar todos los vértices con su correspondiente color. Nótese que se deben de declarar primero los colores que los vértices, porque, OpenGL funciona como una máquina de estados finitos, donde se deben declarar los atributos antes de las figuras.

Hasta ahora ya utilizamos la matriz de modelado para hacer las rotaciones, pero como vemos en el evento `OnResize` hacemos uso de la matriz de proyección para definir nuestro puerto de visión y perspectiva. Entender como funciona realmente la matriz de proyección es un poco complicado, por ahora sólo debemos entender que el evento `OnResize` se dispara cada vez que modificamos el tamaño de nuestra ventana, por lo cual debemos ajustar la escala de lo que estamos observando para no perder las dimensiones, es por



eso que debemos tenerlo presente. Observen como al final del código para este evento usamos el método Refresh del TForm, lo cual obliga al formulario a volver a dibujar la escena con la matriz de proyección ya ajustada. También cuando Windows trata de dibujar el formulario después de haberlo creado se dispara OnResize, lo que garantiza que, desde el principio, tendremos ajustada nuestra perspectiva sin la necesidad de modificar nosotros mismos el tamaño de nuestra ventana. El procedimiento del Timer lo único que hace es incrementar una variable global llamada Angulo y volver a dibujar la escena; la operación mod que efectuamos a esta variable es para asegurarnos que su dominio solo fluctuará entre los valores 0 y 359, que son los 360 grados de una circunferencia. Nosotros en el evento OnPaint rotamos la escena conforme a esta variable, lo que ocasiona que la escena con escena obtengamos nuestra figura en diferentes posiciones relativas, y con esto obtener la animación. El número de unidades que incrementemos a esta variable con cada escena, será lo que determine la calidad de nuestra animación a final de cuentas. Ahora como estamos utilizando el Canvas del formulario como lienzo y continuamente estamos mandando a *redibujar* la escena, forzamos a que Windows tenga que borrar el contenido del formulario anterior antes de mostrar el nuevo contenido, lo cual ocasiona un continuo parpadeo en nuestra animación, cosa que no es nada deseable. Por esto lo que tenemos que hacer es evitar que Windows se tome esta molestia, de cualquier modo nosotros mismos estamos dibujando sobre esta ventana lo que se nos va antojando, así que para qué perder ese valioso tiempo.

Para evitar que Windows borre el contenido del formulario debemos utilizar un manejador de mensajes que opere para el mensaje WM\_ERASEBKGD, el cual es enviado por Windows cada que se necesita borrar el fondo de una ventana, para evitar que Windows tome alguna acción, sólo debemos devolver como resultado un valor distinto de 0 para indicar que nosotros mismos haremos ese trabajo (en nuestro caso devolvemos un 1). Los manejadores de mensajes en este tipo de aplicaciones suelen ser bastante útiles, ya veremos después como responder a cambios en la paleta de colores y en la resolución de la pantalla por citar algunos casos.

Y con esto es suficiente para poder compilar y observar nuestra primera animación con OpenGL (Figura 5).

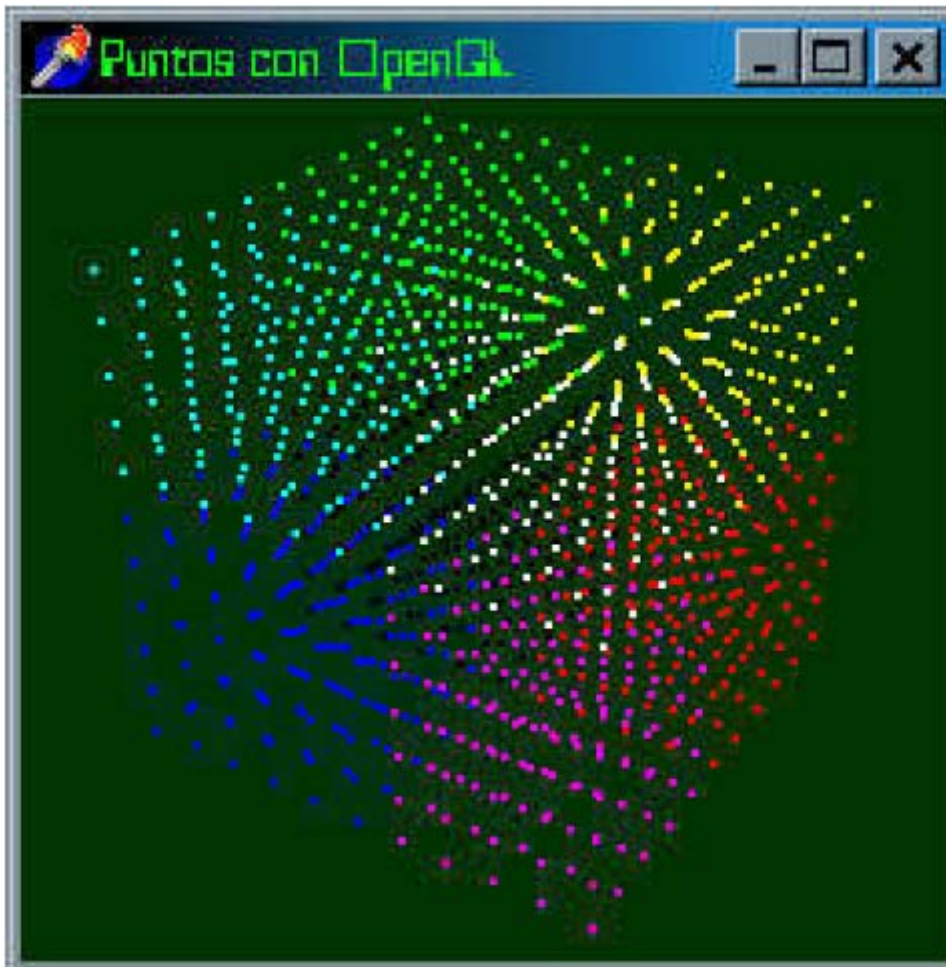


Figura 5. Aplicación de Opengl en ejecución

Con muy pocas líneas de código hemos conseguido grandes resultados, como ya es costumbre con los programas que hacemos con Delphi. Ahora algunas consideraciones extras: por alguna extraña razón las aplicaciones que hacemos con Delphi usando OpenGL ocasionan un conflicto cuando las ejecutamos desde el IDE, por lo cual es recomendable solo compilarlas y ejecutarlas desde fuera del Ambiente de Desarrollo Integrado.

Y una cosa más, recordemos que OpenGL utiliza aceleración por Hardware para optimizar el *renderizado* de las escenas, pero cuando el nuestro no provee esas características de aceleración, todo lo que debería de hacerse por hardware se hace por software.

# APÉNDICE D

## **TRABAJOS PUBLICADOS**