



CENTRO DE INVESTIGACIONES EN ÓPTICA A.C.

**“MODELADO TRIDIMENSIONAL DE PIEZAS MECÁNICAS  
MEDIANTE METROLOGÍA ÓPTICA Y ALGORITMOS  
COMPUTACIONALES”**

**TESIS**

PARA OPTAR POR EL GRADO DE:

**MAESTRO EN OPTOMECATRÓNICA**

PRESENTA:

**ING. CESAR HIDELBERTO MENDEZ TORRES**

ASESOR:

**DR. J. APOLINAR MUÑOZ RODRÍGUEZ**

*LEÓN, GUANAJUATO, A 10 DE JULIO DE 2014*

*Versión definitiva, incluye cambios sugeridos por revisores*



## Contenido

RESUMEN .....	2
JUSTIFICACION .....	2
OBJETIVOS .....	3
PROLOGO .....	4
CAPITULO I- INTRODUCCION .....	5
1.1 ANTECEDENTES .....	5
1.2 MODELADO GEOMÉTRICO DE SUPERFICIES.....	6
1.2.1 Ecuaciones paramétricas de las curvas .....	7
1.2.2 Curvas B-Splines .....	7
1.2.3 NURBS .....	8
1.3 VISIÓN POR COMPUTADORA .....	10
1.3.1 Sistema de visión.....	10
1.3.2 Cámaras .....	12
1.3.3 Procesamiento Digital de Imágenes.....	13
1.3.4 Interpretación de las imágenes para su análisis .....	14
1.4 TECNICAS DE LUZ ESTRUCTURADA.....	15
1.4.1 Proyección de un punto .....	15
1.4.2 Proyección de una línea.....	16
1.4.3 Proyección de un patrón de franjas .....	17
1.5 REDES NEURONALES.....	19
CAPITULO II.- TEORIA BASICA.....	21
2.1 MODELADO DE LA SUPERFICIE CON FUNCIONES NURBS .....	21
2.1.1 funciones base de curvas y superficies B-Spline no uniformes y racionales (NURBS) .....	22
2.1.2 Superficies NURBS .....	24
2.1.3 Selección de los pesos $W$ en la expresión para una superficie NURBS .....	25
2.1.4 Diferenciabilidad en los NURBS .....	26
2.1.5 Continuidad en los NURBS.....	27
2.2 MODELADO DEL SISTEMA DE VISIÓN.....	31
2.2.1 Distorsión de la lente.....	31
2.2.2 Modelado de la Cámara de proyección de perspectiva .....	32

2.2.3 Los parámetros extrínsecos .....	33
2.2.4 Parámetros intrínsecos.....	34
2.2.5 Transformación proyectiva.....	34
2.2.6 Calibración clásica del sistema de visión .....	35
2.3 CALIBRACION DEL ARREGLO OPTICO .....	36
2.3.1 Arreglo óptico implementado.....	36
2.4 PROCESAMIENTO DE LAS IMÁGENES PARA OBTENER LA POSICION DE LA LINEA LASER.....	40
2.4.1 Aproximación gaussiana .....	40
2.4.2 Curvas Bézier .....	41
CAPITULO III.- SISTEMA DE VISION PARA RECONSTRUCCION Y MODELADO DE OBJETOS.....	43
3.1 COMPONENTES DEL SISTEMA DE VISION .....	44
3.2 CONTROL DEL DISPOSITIVO DE VISION .....	45
3.2.1 Interfaz gráfica para la cámara.....	45
3.3 CALIBRACION DE UN SISTEMA DE VISION ARTIFICIAL .....	48
3.4 CALIBRACION IMPLEMENTADA.....	50
3.4.1 Estructura de la red neuronal.....	51
3.4.2 Implementación de la red neuronal Bézier.....	53
3.4.3 Conversión de los datos a unidades de longitud .....	55
3.5 DESCRIPCION DEL SISTEMA OPTOMECANICO PARA ESCANEEO DE PIEZAS .....	57
3.5.1 sistema mecánico y el proceso de reconstrucción.....	57
3.5.2 Control del sistema de posicionamiento .....	58
3.6 INTERFAZ ELECTRÓNICA PARA EL CONTROL DEL MOTOR .....	60
3.7 DESCRIPCIÓN DE LA TECNICA PARA RECONSTRUCCIÓN DE OBJETOS .....	62
3.7.1 Detección y procesamiento de la línea láser .....	62
3.8 MODELADO GEOMETRICO .....	64
3.8.1 Parametrización de los puntos .....	64
3.8.2 Modelado con superficies NURBS.....	67
3.9 INTEGRACIÓN DEL SISTEMA ÓPTICO Y MECÁNICO.....	70
CAPITULO IV.- RESULTADOS EXPERIMENTALES .....	72
4.1 RECUPERACIÓN DE SUPERFICIES TRIDIMENSIONALES.....	72
4.1.1 Reconstrucción de la superficie de objetos mecánicos.....	72
4.1.2 Determinación de la exactitud de la red neuronal Bézier .....	74

4.1.3 <i>Calculo de la resolución del sistema</i> .....	75
4.2 EVALUACIÓN DEL SISTEMA Y LOS DATOS EXTRAIDOS PARA EL ENGRANE.....	77
4.2.1 <i>repetibilidad del sistema</i> .....	78
4.2.2 <i>incertidumbre del sistema</i> .....	79
4.3 RESULTADOS DE LA RECONSTRUCCIÓN Y MODELADO DE OBJETOS MECÁNICOS.....	81
CONCLUSIONES.....	88
PRINCIPALES APORTACIONES.....	88
TRABAJO A FUTURO.....	89
REFERENCIAS BIBLIOGRAFICAS.....	90

## RESUMEN

En este trabajo de tesis se presenta la implementación de un sistema de visión artificial para la reconstrucción 3D mediante la proyección de un patrón estructurado y algoritmos computacionales para la recuperación y modelado de superficies complejas, se pretende realizar un proceso viable para generar formas alternativas al diseño CAD/CAM/CAE, donde la principal ventaja es el tiempo requerido para generar los objetos de ingeniería.

Para obtener el modelo geométrico que representa adecuadamente un objeto real es necesaria una expresión matemática que sea capaz de describir la forma de tal objeto. Los NURBS (Not Uniform Rational B-Splines) representan una forma viable y natural para generar modelos matemáticos de objetos de forma libre, para la aplicación de esta técnica es necesario primero extraer los puntos 3D de la superficie a modelar.

El método usado para extraer los puntos 3D es la proyección de una línea laser, éste se basa principalmente en la triangulación geométrica para recuperar las posiciones de los puntos en la escena 3D. Para implementar esta técnica se diseñó y construyó un sistema optomecánico que permite la rotación y captura de la superficie del objeto en la escena 3D, en este sistema se cuenta con un arreglo óptico compuesto por una cámara y un proyector, y un sistema mecánico de posicionamiento que son coordinados por un software para extraer la información de la línea laser proyectada sobre la topografía del objeto.

La técnica perfilométrica usada se basa en el desplazamiento que sufre la proyección cuando ésta incide sobre la superficie del objeto y que está relacionado geoméricamente con la topografía del objeto. Por medio de un modelado matemático del perfil de la proyección de luz es posible realizar un posicionamiento del haz laser con una exactitud de fracciones de pixel, permitiendo con esto una reconstrucción más precisa de las piezas bajo estudio.

Una vez extraída la nube de puntos que describe la superficie del objeto en escena se procede a la aplicación de la técnica de modelado con NURBS, ésta representara la superficie del objeto por medio de modelos matemáticos que permitirán su análisis y manipulación para fines de diseño y procesos de ingeniería inversa.

## JUSTIFICACION

En la actualidad existe una gran diversidad de industrias cuyo soporte son las tecnologías basadas en el modelado, análisis y diseño por medio de software CAD/CAM (Computer Aided Design/Computer Aided Manufacturing), anteriormente se mantenía un diseño convencional donde los objetos eran creados en software y posteriormente eran procesados para realizarlos físicamente.

mente, en la actualidad se tiene la necesidad del uso de procesos más eficientes que permitan el uso de métodos alternativos al diseño tradicional y que sean más versátiles, la ingeniería inversa es una disciplina que puede cubrir esta necesidad, donde los objetos reales son transformados en modelos de ingeniería.

Un modelo computacional de un objeto representa grandes ventajas para el mejoramiento de aspectos como la calidad, eficiencia en el diseño y el análisis para la manufactura. Con la implementación de un sistema de reconstrucción 3D se tiene un método viable y preciso para sustentar procesos de ingeniería inversa, donde se logre medir objetos reales y llevarlos a un modelo geométrico que permita explotar las ventajas de la tecnología CAD/CAM.

La técnica de modelado NURBS es comúnmente usada en procesos CAD/CAM pero también es aplicada en análisis de diseño CAE (Computer Aided Engineering), esta técnica forma parte de un gran número de estándares industriales como el IGES (Initial Graphics Exchange Specification), STEP (Standar for Exchange of Product model data), ACIS (Software núcleo para modelado geométrico) y PHIGS (Programmer's Hierarchical Interactive Graphics Systems).

En general el modelado con NURBS es altamente intuitivo y predecible, facilita la representación de formas complejas con un nivel de error muy bajo ( $\text{error} < 0.2\%$ ), lo cual asegura que el modelo se ajusta eficientemente a las dimensiones del objeto real.

## OBJETIVOS

Este proyecto tiene como finalidad la implementación de técnicas de metrología óptica y visión computacional para el modelado de piezas mecánicas, con esto se pretende la implementación de un sistema que sea capaz de modelar el contorno de piezas mecánicas para su análisis y su correspondiente aplicación en procesos industriales tales como en las áreas de calidad y prototipado de piezas mecánicas.

Como objetivos específicos podemos mencionar:

- La implementación de un arreglo óptico para la reconstrucción de piezas mecánicas que permita el análisis dimensional de las mismas.
- Implementar algoritmos de inteligencia artificial que sean capaces de reconocer patrones y que a su vez nos permita modelar adecuadamente la forma de la pieza física.
- Implementación de algoritmos computacionales para la calibración del sistema de visión artificial así también para el procesamiento digital de las imágenes y obtención de los modelos virtuales.
- Desarrollar el modelo matemático del sistema de visión artificial que permitan una correcta reconstrucción de las piezas mecánicas.

- Implementar algoritmos matemáticos para generar superficies capaces de describir las piezas escaneadas.

## PROLOGO

El presente proyecto de tesis de la Maestría en Optomecatrónica, es el producto del trabajo de investigación y desarrollo de sistemas de visión artificial aplicados en metrología, realizado bajo la supervisión del Dr. J. Apolinar Muñoz Rodríguez, de la división de Óptica , Metrología Óptica, del Centro de investigaciones en Óptica, A.C.

Esta tesis describe un sistema optomecatrónico para reconstruir objetos por medio de técnicas de luz estructurada, inteligencia artificial y un modelado geométrico. El modelo es representado por técnicas matemáticas que permiten describir cualquier parte de su superficie por medio de una función matemática. La tesis está dividida de la siguiente manera:

Capítulo 1: sección dedicada a la descripción de los conceptos básicos, tales como: procesamiento digital de imágenes, visión artificial, técnicas de luz estructurada para recuperación de formas, técnicas de modelado geométrico etc.

Capítulo 2: esta sección es dedicada a la descripción de la teoría básica referente al proyecto desarrollado, en esta parte se describen conceptos como: modelado de un sistema de visión, arreglo óptico implementado en el proyecto, calibración del sistema de visión, técnica de proyección de una línea laser y el procesamiento digital para recuperar las posiciones de ésta, modelado geométrico mediante NURBS, etc.

Capítulo 3: En esta parte se da la descripción del sistema optomecatrónico implementado, en este caso se explica el software para control del sistema de visión, los algoritmos computacionales para recuperación de forma y modelado geométricos, así como el proceso de calibración por medio de redes neuronales.

Capítulo 4: Se presentan los resultados del sistema, se da un análisis de error respecto a la recuperación de forma concerniente a un engrane de dientes rectos y otros objetos de forma libre y también se hace un análisis del error presente en el modelado de los objetos.



# CAPITULO I- INTRODUCCION

---

La geometría ha jugado un rol crucial en el desarrollo de varias disciplinas científicas y de ingeniería como astronomía, mecánica, balística, arquitectura, etc. Lo que hace a la geometría única y particularmente una rama atractiva de las matemáticas es su propiedad visual [1].

Con el advenimiento de computadoras veloces se logró realizar diseños geométricos basados en funciones matemáticas, estimulada la técnica de modelado por las ciencias en computación la geometría regreso como herramienta fundamental para la manufactura, gráficos computacionales, visión por computadora, etc.

El modelado geométrico está emergiendo rápidamente como un área de interés central en la investigación y aplicación como solución al diseño de productos para ingeniería, como campo el modelado abarca varias disciplinas, incluyendo matemáticas, ciencia en computación e ingeniería.

Modelar una superficie tradicionalmente identifica un conjunto de técnicas que pueden asemejar ciertas clases de superficies paramétricas, sujetas a condiciones particulares de forma y textura. Como técnica, el modelado se ha desarrollado en varios campos de la industria incluyendo la del automóvil, aeroespacial, construcción de barcos, entre otras.

Este trabajo tendrá un énfasis especial en el área referente a la reconstrucción de objetos mecánicos y el modelado geométrico de los mismos, como ejemplo de un objeto mecánico de interés para la industria se tiene el engrane, este elemento de transmisión que se ha implementado en infinidad de sistemas industriales, con amplio uso en los procesos de manufactura modernos y por lo cual es relevante su análisis por medio de sistemas de visión artificial.

En las siguientes secciones se dará una descripción breve de los temas referentes a este proyecto, tal como: modelado de formas por medio de NURBS, técnicas de luz estructurada para reconocimiento y recuperación de forma, el sistema de visión y el procesamiento digital de imágenes.

## 1.1 ANTECEDENTES

La optimización de procesos para describir geoméricamente, tanto fenómenos como objetos físicos a través de funciones matemáticas ha sido un tema relevante para la industria y la investigación. Dadas las necesidades de un modelado eficiente se implementaron las funciones polinomiales no uniformes y racionales, cuyas características permitieron introducir el modelado para aplicaciones en procesos de ingeniería e investigación, donde era necesario analizar el objeto o fenómeno por medio de una función matemática.

Con la creación del láser y el desarrollo de la visión artificial fue posible desarrollar una gran cantidad de aplicaciones relevantes para la metrología de objetos mecánicos. En el Centro de Investiga-

ciones en Óptica se le ha dado un desarrollo especial a la recuperación de la forma de objetos por medio de técnicas basadas en la proyección de luz estructurada, redes neuronales, visión por computadora y metrología óptica, donde para lograr una reconstrucción continua a partir de los datos discretos, se realiza un modelo del conjunto de puntos por medio de funciones polinomiales racionales.

Algunas de las aplicaciones dentro de la metrología óptica son la evaluación automática de la calidad de piezas mecánicas, el control dimensional y el prototipado rápido. Estas son algunas de las necesidades de la industria que se soportan en la generación de reconstrucciones tridimensionales de los objetos [2].

La disciplina del modelado geométrico es una interrelación de un conjunto de métodos matemáticos que nosotros usamos para describir la forma de un objeto. Cuando construimos algún modelo de un objeto, creamos un sustituto, una representación de un objeto que puede existir físicamente o ser solo un diseño.

El modelado de piezas mecánicas representa en la actualidad una parte importante del diseño CAD/CAM/CAE. Con la implementación de dispositivos que reconstruyan y a su vez modelen los objetos se tiene la posibilidad de integrar procesos más versátiles, donde el modelo del producto es probado con mayor rapidez por medio de software y donde los costos de producción se ven reducidos en gran medida.

El uso de luz estructurada, permite que el sistema de visión sea confiable y que los datos adquiridos sean fáciles de interpretar. Así mismo, la incorporación de redes neuronales a los métodos ópticos permite una automatización del proceso de reconstrucción.

Uno de los principales retos en este proyecto es la realización de la reconstrucción 3D y el modelado geométrico de objetos mecánicos en el menor tiempo posible. Por tanto para lograr tales objetivos es necesario implementar algunas técnicas de metrología óptica y algoritmos matemáticos que nos permitan dar una solución a esta necesidad.

## 1.2 MODELADO GEOMÉTRICO DE SUPERFICIES

La disciplina del modelado geométrico la usamos para describir la forma de un objeto o para expresar algún proceso físico, esto hace posible una reconstrucción rápida y exacta de objetos útiles para la industria e investigación, ésta es la base del diseño asistido por computadora y manufactura de aviones, carros y productos, mejorando la eficiencia, obtener diseños más atractivos y dar más alternativas a través de la producción personalizada.

### 1.2.1 Ecuaciones paramétricas de las curvas

La parametrización de superficies es una forma viable, rápida y estable de modelar objetos. Una función explícita  $z = f(x, y)$ , que represente una superficie en un espacio coordenado se puede expresar de forma paramétrica como:

$$x = f(u, v), y = f(u, v), z = f(u, v)$$

Esta función paramétrica define la forma del objeto respecto a un conjunto de variables  $(u, v)$ , donde las variables paramétricas son asignadas en un rango  $u, v \in [0, 1]$ ,  $f(u, v)$  representa la función base para construir la superficie del modelo, estas variables son independientes del sistema coordenado donde se muestren y permiten el ajuste de curvas a través de un conjunto de puntos, donde las propiedades intrínsecas de la forma del objeto permanecen invariantes, dando así versatilidad en el modelado de la superficie.

La forma más simple y común para modelar una superficie de manera matemática es un "parche", esta es una forma conveniente de dividir una superficie compleja en partes más fáciles de manejar.

Existen varios métodos matemáticos para el modelado de superficies entre los que se puede mencionar B-Splines, Bézier y NURBS.

Para poder modelar una superficie es necesario un polígono de control, este corresponde al grupo de puntos que describen la superficie del objeto, las funciones que forman la base matemática de la superficie parametrizada y las variables paramétricas que han de ser evaluadas para obtener la superficie del modelo.

### 1.2.2 Curvas B-Splines

Las curvas B-Spline son definidas por funciones base no uniformes, esto es, la función base que define un segmento se diferencia de otra que define otro segmento, esto nos permite interpolar uno o más puntos de control, dependiendo de la situación del modelado, la forma no racional se expresa como:

$$p(u) = \sum_{i=0}^n p_i N_{i,k}(u)$$

Para una curva B-Spline, el parámetro  $k$  define el grado de la función polinomial base y este es usualmente independiente del número de puntos control. La función base de esta técnica se expresa como:

$$N_{i,k}(u) = \frac{(u - t_i) * N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) * N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}}$$

Este es un método recursivo, donde se calculan las funciones base  $N_{i,k-1}(u)$  para poder calcular funciones de orden  $N_{i,k}(u)$ . Los  $t_i$  son llamados nodos y un conjunto de nodos componen un vector de nodos, estos relacionan la variable paramétrica  $u$  a  $p_i$  que son los puntos control donde  $i = 0, \dots, n$ .

Una curva B-Spline es constituida de segmentos de curvas, donde cada segmento es influenciado por solo unos cuantos puntos de control, esta característica permite modificar el modelo solo localmente (figura 1.1).

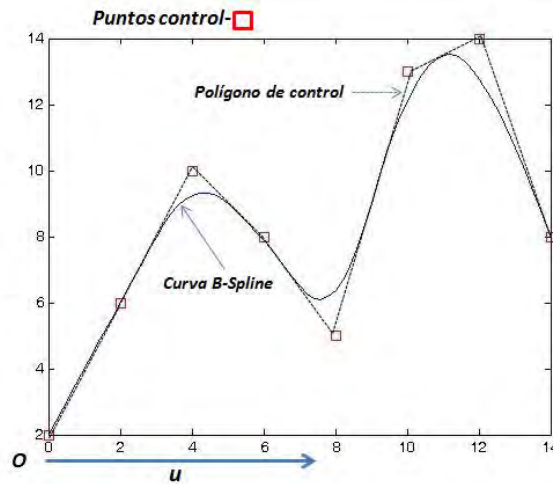


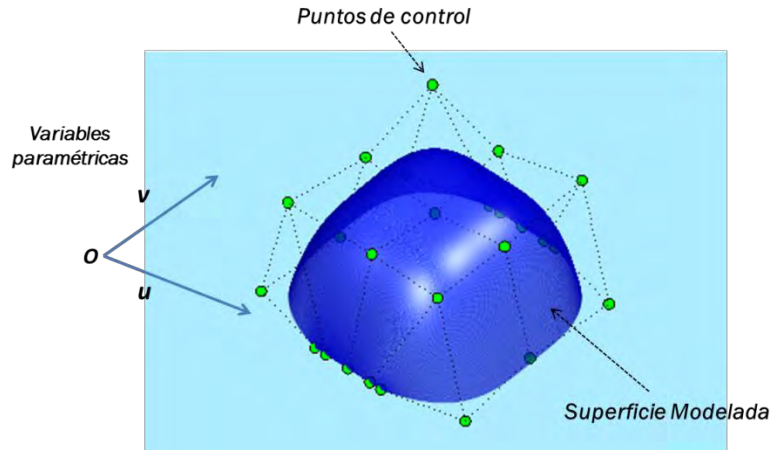
Figura.-1.1.- Grafica del ajuste de una línea B-Spline a un conjunto de puntos control.

### 1.2.3 NURBS

Las B-Splines no uniformes y racionales (NURBS) es un método que permite la construcción de formas geométricas difíciles, esta técnica se basa en la construcción de líneas que tienden a interpolar los puntos de una superficie y donde las funciones base que construyen las líneas permiten describir matemáticamente cualquier punto en la superficie.

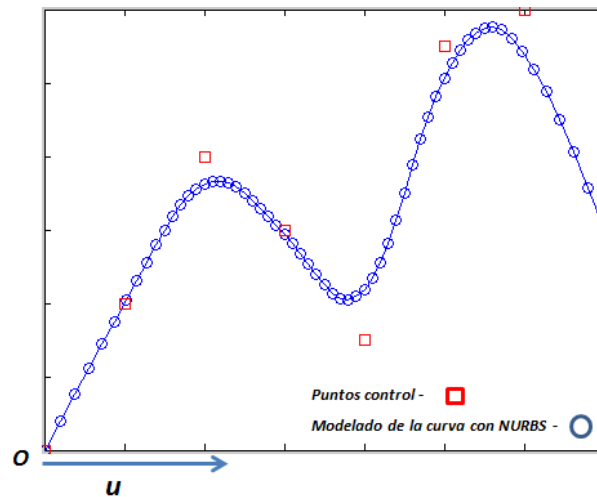
Una curva NURBS se define por su grado, un conjunto de puntos control con pesos asociados y un vector de nodos que relaciona los puntos control con las funciones base.

Las curvas y superficies NURBS son generalizaciones de las curvas B-Spline y curvas de Bézier, siendo su diferencia principal la ponderación de los puntos control que hacen a las curvas NURBS racionales, las superficies NURBS evolucionan en dos direcciones paramétricas, llamadas  $u$  y  $v$  (figura 1.2).



**Figura.-1.2.- Esquema representativo del ajuste de una superficie NURBS, evaluada en la dirección de las variables paramétricas  $u$  y  $v$ .**

Mediante la evaluación de una curva NURBS en diversos valores de la variable paramétrica  $u$  (figura 1.3), la curva se puede representar en un espacio Cartesiano de dos o tres dimensiones. Asimismo, mediante la evaluación de una superficie NURBS (figura 1.4) en diversos valores de las dos variables paramétricas  $u$  y  $v$ , la superficie se puede representar en el espacio cartesiano.

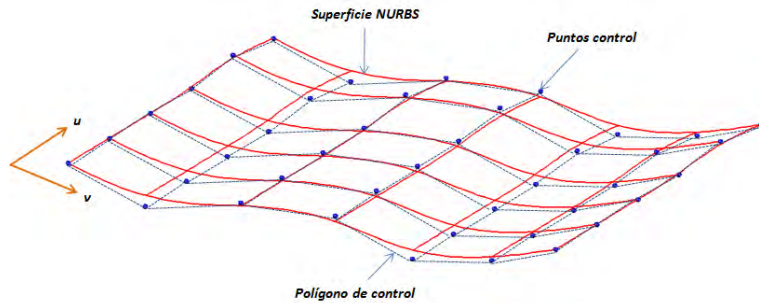


**Figura.-1.3.- Modelado de una curva con NURBS, evaluada en la dirección de la variable paramétrica  $u$ .**

Las curvas y superficies NURBS son útiles por varias razones:

- Son invariantes bajo transformaciones afines, así como de perspectiva, operaciones de rotación y traslación se pueden emplear en las curvas y superficies NURBS aplicándolas a sus puntos de control.
- Ofrecen una estructura matemática común para figuras analíticas estándar (por ejemplo, cónicas) y figuras de forma libre.

- Proporcionan flexibilidad para diseñar una gran variedad de figuras.
- Pueden ser evaluados rápidamente por algoritmos numéricamente estables y precisos.



**Figura.-1.4.- Modelado de una superficie con NURBS, evaluada en la dirección de la variable paramétrica u y v.**

## 1.3 VISIÓN POR COMPUTADORA

La visión es uno de los sentidos que permiten a los humanos percibir y entender el mundo que los rodea, mientras que la visión artificial tiende a duplicar el efecto de la visión humana por medio de la percepción electrónica y entendimiento de una imagen [3].

Dar la habilidad a una computadora de ver y analizar el mundo exterior no es una tarea fácil, debido a que vivimos en un espacio de tres dimensiones y a que los sensores de visión con los que se cuenta proveen usualmente solo de imágenes que contienen información de un espacio en dos dimensiones.

Por tanto la visión artificial es un área que concierne al entendimiento y análisis del mundo exterior, los sistemas resultantes pueden ser aplicados a gran número de procesos que requieren de un grado de autonomía y control, tal como: reconocimiento automático de objetos, cirugía remota, control dimensional, control de vehículos autónomos, etc.

En visión artificial se realiza un proceso dividido en etapas que pueden ser descritas como:

- Adquisición de las imágenes del mundo 3D
- Procesamiento digital de las imágenes
- Interpretación de las imágenes para su análisis

### 1.3.1 Sistema de visión

Para entender la visión artificial podemos describir el modelo Pin-hole de una cámara, esto con el fin de conocer los procesos ópticos que tienen lugar en el sistema de visión, en la figura 1.5 puede-

mos ver el modelo más básico de un sistema de visión, este modelo solo es una idealización de una cámara real.

Como se había comentado con anterioridad un sistema de visión artificial tiene una pérdida de información en su sensor de visión, esto contribuye a una gran pérdida de información geométrica y lo cual hace imposible analizar y reconstruir un objeto 3D solo con información obtenida de las imágenes, por tanto es necesario adaptar un modelo como el de "Pin-hole" al sistema de visión para poder llevar a cabo un análisis del objeto.

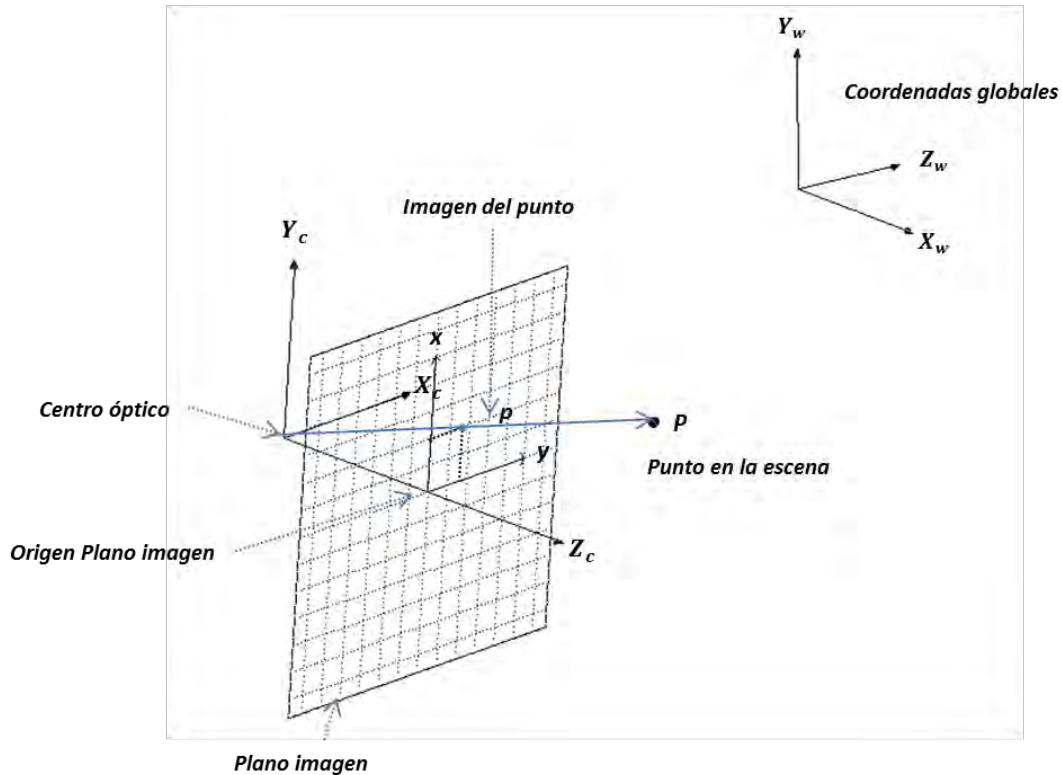


Figura.-1.5 Modelo Pin-hole para una cámara digital.

Este modelo físico ("Pin-hole") corresponde a un modelo matemático de una proyección de perspectiva, la transformación proyectiva traza los puntos a lo largo de rayos o líneas pero que no preservan ángulos y colinearidad [3].

El gran problema con el modelo "Pin-hole" y de su única vista disponible es que la transformación proyectiva ve un objeto pequeño y cercano a la cámara de la misma forma que ve un objeto grande y lejano de la cámara. De aquí se desprende la necesidad de obtener una relación geométrica entre el sistema de visión artificial y el mundo 3D.

En estos modelos la implementación de una técnica de luz estructurada también se hace necesaria para poder localizar y analizar las formas geométricas de objetos.

### 1.3.2 Cámaras

La cámara es el dispositivo que permitirá obtener la información del mundo exterior para poder pasar de una señal luminosa proveniente de una escena a una señal de información ya sea analógica o digital, la cual puede ser usada para conocer datos relevantes [4].

*Tipos de cámaras:* las cámaras pueden ser divididas en analógicas y digitales.

Una cámara analógica transmite continuamente una señal electrónica variable y en tiempo real, la frecuencia y amplitud de esta señal es interpretada como información de video por un dispositivo analógico de salida y la calidad de la señal de video analógica así como la forma en que esta es interpretada afectan las imágenes de video resultantes.

En la cámara digital se transmite información binaria en la forma de una señal electrónica, aunque el voltaje correspondiente a la intensidad dada para un cierto pixel de una imagen sea continuo, el proceso de conversión analógica-digital discretiza tal señal y asigna un valor de una escala de grises entre 0 (negro) y  $2^{N-1}$ , donde N es el número de bits de la codificación. Por último un dispositivo de salida convierte los datos binarios a información de video.

*Características importantes de una cámara para aplicaciones de visión por computadora:*

Los sensores modernos son dispositivos electrónicos de estado sólido los cuales contienen millones de áreas discretas que poseen foto-detectores, estas áreas reciben el nombre de pixeles.

En estos días casi todos los sensores usados en aplicaciones de visión por computadora pueden ser CCD (Charge-Coupled Device) o CMOS (Complementary Metal Oxide Semiconductor).

*Propiedades espectrales de una cámara:* Los sensores CCD y los CMOS son sensitivos a longitudes de onda que van desde los 350 nm – 1050 nm, las cámaras de más alta calidad proveen un filtro de corte infrarrojo para poder trabajar específicamente en el espectro visible.

Dado que las cámaras poseen un sensor de estado sólido basado en el efecto fotoeléctrico y como resultado no se puede distinguir entre colores, es necesario la implementación de sistemas para reconocer colores, hay dos tipos de cámaras CCD a color: las de tipo single chip y las de tipo three-chip (figura 1.6). Las cámaras de tipo single chip usan un filtro óptico para separar la luz incidente en una serie de colores y así dirigir este haz a un conjunto de pixeles que reconozcan cada color, las cámaras de tipo three-chip usan un prisma para dirigir cada sección del espectro incidente a un chip específico, con esto se da una reproducción de color más precisa, además que se tiene una separación de las intensidades RGB del objeto en cuestión.

Otro aspecto importante del sensor es la medida de este, el tamaño del área activa es importante para determinar el campo de visión del sistema, hay varios tamaños de sensores y son: 1/4", 1/3", 1/2", 1/1.8", 2/3", 1" y 1.2".



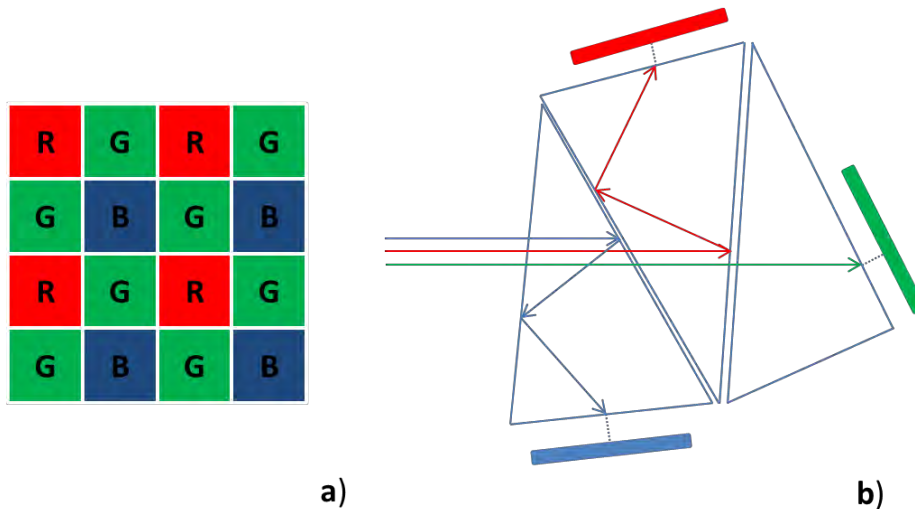


Figura.-1.6 – Sistemas (a) single chip y (b) three chip para reconocimiento de colores

### Resolución e interfaces para cámaras digitales

Una de las características más importantes de una cámara es la resolución, esta es una medida de la habilidad del sistema de visión para reproducir detalles del objeto en cuestión, esta característica está limitada por la óptica del dispositivo y el sensor de visión, a mayor resolución podemos extraer más características del objeto visualizado.

En las cámaras digitales podemos distinguir varias interfaces, entre las más comunes podemos mencionar tarjetas de captura, Firewire, Camera Link, GigE, Ethernet y USB.

### 1.3.3 Procesamiento Digital de Imágenes

La interpretación de las imágenes constituye la principal herramienta de visión por computadora para aproximar problemas que los humanos logran involuntariamente [5]. Una imagen digital puede ser considerada como una representación discreta de información que posee datos tanto espaciales como de intensidad (figura 1.7).

Entonces una imagen puede ser tratada como una señal bidimensional [6], la señal digital discreta representada como  $I(m, n)$  representa la respuesta del sensor para una posición fija  $(m, n)$ , este es el resultado de la discretización que ocurre naturalmente con ciertos tipos de sensores para visión, estos proporcionan un promedio de una señal continua sobre cada píxel, que generalmente son elementos cuadrados.

De manera convencional una imagen se compone de varios canales, cada canal representa un arreglo bidimensional de  $m$  filas por  $n$  columnas, donde una localidad  $I(m, n)$  posee un valor numérico para representar su color de acuerdo a un mapa de colores que el usuario asigna, en el caso de una imagen RGB tenemos tres canales para representar los colores rojo, verde y azul, con opción de un cuarto canal que posee información de la transparencia (canal **alpha**), para una ima-

gen a escala de grises solo tenemos un canal y donde cada localidad  $I(m, n)$  posee un valor entre 0-255, donde 0 equivale a un negro y 255 corresponde a un color blanco.

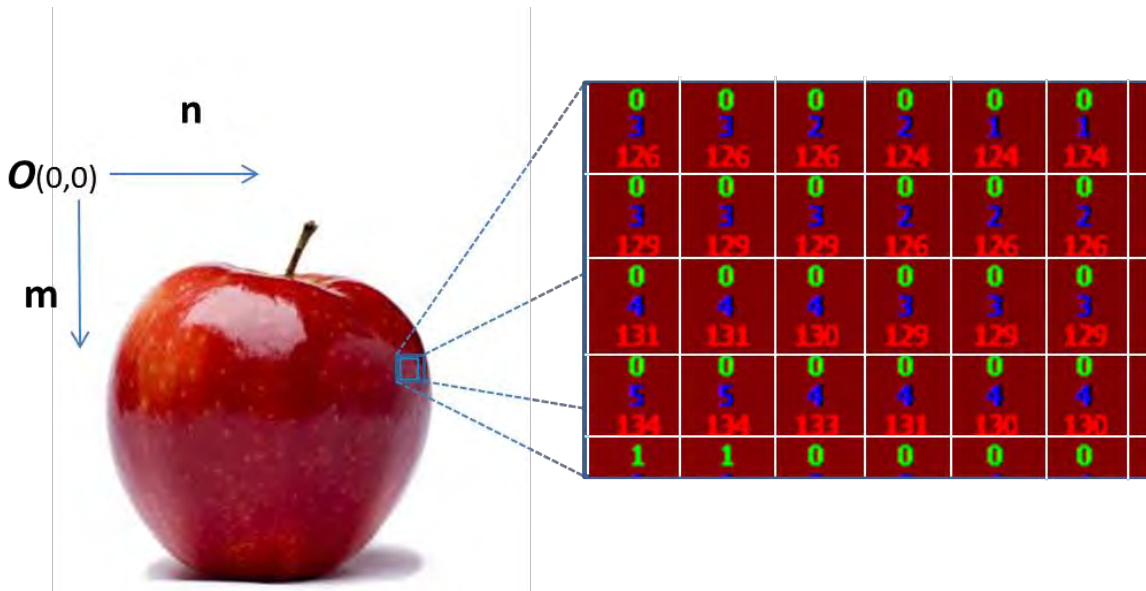


Figura.-1.7 – Representación de una imagen digital RGB

El objetivo del procesamiento digital de imágenes es mejorar la calidad de las imágenes para su posterior utilización o interpretación, a pesar de que esta es un área muy ligada a la visión computacional, el objetivo final de ambas es diferente.

Algunos ejemplos de procesos llevados en esta etapa son:

- Remover ruido.
- Remover problemas por movimiento o desenfoco.
- Mejorar ciertas propiedades como color, contraste, estructura, etc.
- Agregar colores a imágenes monocromáticas etc.

### 1.3.4 Interpretación de las imágenes para su análisis

El entendimiento de una imagen por medio de una computadora puede ser visto como un intento para encontrar una relación entre las imágenes de entrada y el modelo previamente establecido del espacio observado, la transición de las imágenes de entrada hacia el modelo reduce la información contenida en las imágenes a información relevante para la aplicación.

La jerarquía en la interpretación de imágenes y sus respectivos códigos son frecuentemente categorizados en una forma simple, donde solo se distinguen dos niveles: procesamiento de imágenes de bajo nivel y comprensión de imágenes de alto nivel [3].

Métodos de procesamiento de bajo nivel: este tipo de métodos usualmente usan poco conocimiento acerca del contenido de la imagen. Los métodos de bajo nivel con frecuencia incluyen compresión de imágenes, métodos de pre-procesamiento para filtrado de ruido, extracción de bordes y contrastado de imágenes.

Métodos de procesamiento de alto nivel: este tipo de métodos son basados en conocimiento previo, objetivos y planes sobre cómo alcanzar los objetivos, los métodos de inteligencia artificial son los más ampliamente aplicados. Este tipo de métodos tratan de imitar la forma en que un humano adquiere conocimiento y lo usa para tomar decisiones.

## 1.4 TECNICAS DE LUZ ESTRUCTURADA

En esta sección hablaremos solo de algunas técnicas de luz estructurada usadas para la localización y recuperación de la forma de objetos, una de las características principales de este tipo de técnicas es que se pueden recuperar características cuantitativas de los datos 3D  $(x, y, z)$  que pertenecen a la superficie, esto da la posibilidad de conocer la geometría de las formas.

Las técnicas de luz estructurada en general son proyecciones de patrones luminosos que inciden sobre la superficie del objeto a escanear, una cámara que por lo general es electrónica capta el patrón distorsionado por la topografía del objeto y almacena la imagen, con procesos de análisis de imágenes se extrae la información del patrón, por ultimo una relación geométrica del sistema de visión, el proyector y el patrón de luz estructurada nos dará los parámetros para poder cuantificar los datos 3D extraídos del análisis de imágenes, esta relación recibe el nombre de principio de triangulación, usado comúnmente en trabajos de metrología óptica.

Dentro de las técnicas de luz estructurada que usan la triangulación para cuantificar los datos 3D podemos mencionar: Proyección de un punto, Proyección de una línea y la proyección de patrones estáticos.

Existen otras técnicas de luz estructurada que hacen uso de la proyección de patrones codificados para recuperar superficies complejas, este tipo de técnicas requieren de solo unas cuantas imágenes y es la principal ventaja sobre la proyección de patrones simples como la proyección de punto y de línea.

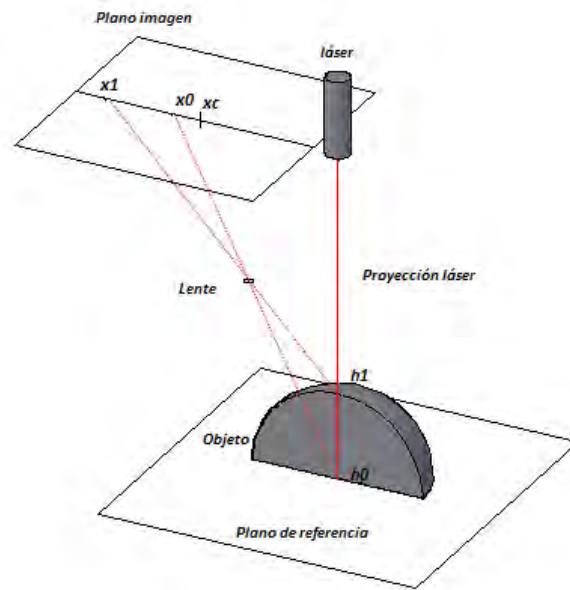
A continuación se dará una pequeña reseña de las técnicas de luz estructurada utilizadas para la recuperación de superficies de objetos.

### 1.4.1 Proyección de un punto

En esta técnica un haz de luz es proyectada sobre la superficie del objeto que está en la escena, la reflexión de este haz en la superficie del objeto es captada por el sistema de visión, con una geometría conocida del arreglo óptico y la información captada por el sistema de visión se realiza una triangulación para obtener la posición geométrica del punto sobre la superficie del objeto.

En la figura 1.8 se muestra un esquema simplificado del arreglo óptico para aplicar la técnica de proyección de punto, en este arreglo tanto la cámara, el proyector del haz y la superficie del objeto están sobre un mismo plano, todos los ángulos del sistema son fijados y calibrados para obtener una referencia a la hora de recuperar los puntos de la superficie del objeto en escena.

Para recuperar la topografía del objeto, es necesario hacer un barrido, donde el objeto es movido a través de la escena o campo de visión de la cámara por medio de un sistema de posicionamiento y donde cada imagen captada contiene solo un punto de la superficie.



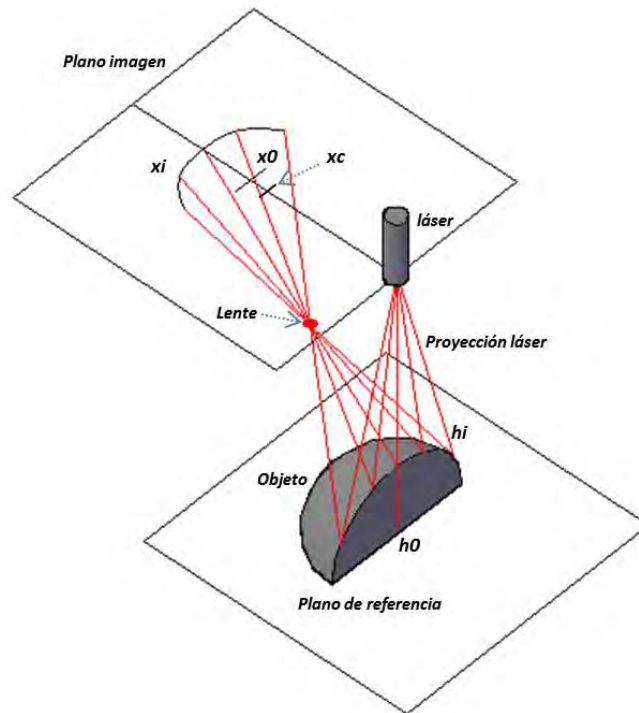
**Figura.-1.8- Geometría del arreglo óptico para la recuperación de forma con proyección de punto.**

#### 1.4.2 Proyección de una línea

En esta técnica se proyecta una línea que describe un plano en el sistema de visión, este plano al incidir sobre la superficie del objeto en la escena, describe una línea que es deformada de acuerdo a la topografía del objeto, esta es utilizada para realizar la triangulación y recuperación de las posiciones de los puntos interceptados.

En este arreglo una cámara es posicionada a un ángulo relativo al proyector (figura 1.9), la posición de la cámara permite obtener una perspectiva de la línea, esta contendrá información de la topografía del objeto, las deformaciones de la línea se verán como desviaciones respecto a una línea de referencia, la cantidad total de esta desviación será calculada y usada para la cuantificación de los puntos 3D  $(x, y, z)$  pertenecientes a la superficie del objeto.

Para obtener la línea de referencia es necesario proyectar el patrón sobre el plano de referencia y así procesar la escena para conocer la desviación concerniente a este plano.



**Figura.-1.9- Geometría del arreglo óptico para la recuperación de forma con proyección de línea.**

Al igual que la técnica de proyección de punto, se debe mover el objeto a través de la escena para poder escanear la totalidad de la superficie, en este caso cada imagen contiene una mayor cantidad de información referente a la superficie, reduciendo la cantidad de imágenes necesarias para la recuperación de la topografía del objeto.

### 1.4.3 Proyección de un patrón de franjas

Cuando un patrón de franjas se proyecta sobre un objeto, estas sufren deformaciones acorde a la topografía del objeto (figuras 1.10 y 1.11), las intensidades de cada pixel  $I(x, y)$  para cada uno de los patrones son descritos por:

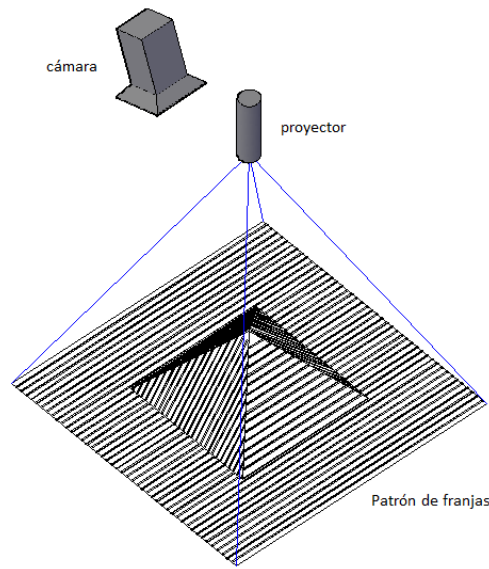
$$I_1(x, y) = I_0(x, y) + I_{mod}(x, y) * \cos[2\pi f_0 x + \phi(x, y)]$$

Donde  $I_0(x, y)$  y  $I_{mod}(x, y)$  son el nivel de DC y el contraste de la iluminación respectivamente,  $f_0$  es la frecuencia espacial del patrón de franjas,  $\phi(x, y)$  es la fase que representa la deformación del patrón de franjas proyectadas sobre el objeto. En este caso la fase contiene la información de la topografía del objeto.

Existen diversas técnicas que son utilizadas para obtener la fase de un patrón de franjas, como: Método de Fourier [7], Método directo de detección de fase [8], método de detección de fase en

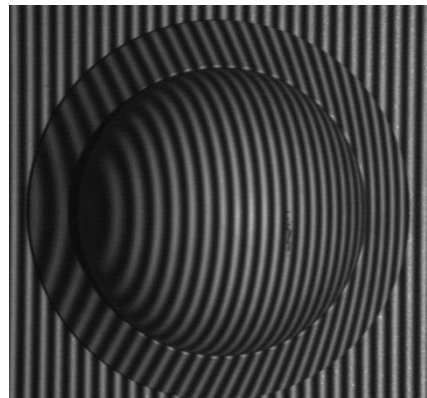
pasos [9] y contorneo por Moire [10]. Sin embargo, la imagen adquirida está envuelta en una fase con discontinuidades de  $2\pi$ , la imagen nos será útil hasta que estas discontinuidades sean eliminadas y esto se llevara a cabo por un desenvolvimiento de fase.

La exactitud de la información medida es determinada por la exactitud en el cambio de fase y por la modulación del patrón de franjas, el número de patrones y el valor de los pasos en la fase depende de la formula usada.



**Figura.-1.10- Arreglo óptico para la recuperación de forma con proyección de franjas.**

Un parámetro importante en la reconstrucción 3D por este método es la determinación de la frecuencia del patrón de franjas, esto para evitar la desintonía, esta se presenta cuando la frecuencia que se utiliza para el procesamiento de imágenes no corresponde al valor de la frecuencia del patrón de franjas proyectado en el objeto, debido principalmente al contenido de alta frecuencia en los cambios de fase.



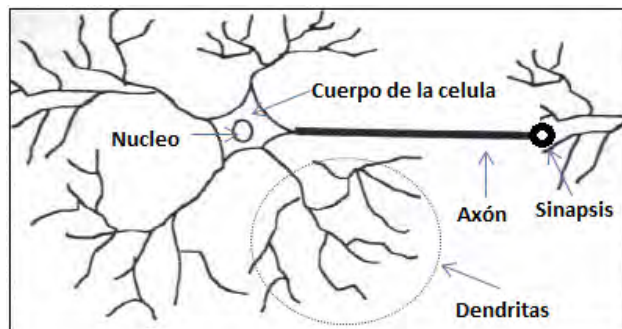
**Figura.- 1.11.- Proyección de franjas sobre un objeto esférico.**

## 1.5 REDES NEURONALES

Una red neuronal artificial es un sistema para procesamiento de información, que es inspirado por la forma en que trabaja el sistema biológico nervioso, tal como el cerebro.

Los elementos claves de este proceso es la estructura llamada “**neurona**”, la interconexión de grandes cantidades de estos elementos permiten solucionar problemas específicos. En la figura 1.12 se tiene la estructura de una neurona, una neurona típica recolecta señales a través de un conjunto de estructuras finas llamadas dendritas y manda señales de actividad eléctrica a través de un cuerpo alargado y fino llamado axón, el cual se divide en cientos de secciones.

Al final de cada axón se encuentra la estructura llamada sinapsis, esta convierte la actividad del axón en efectos eléctricos, que sirven ya sea para inhibir o excitar las conexiones con las otras neuronas.

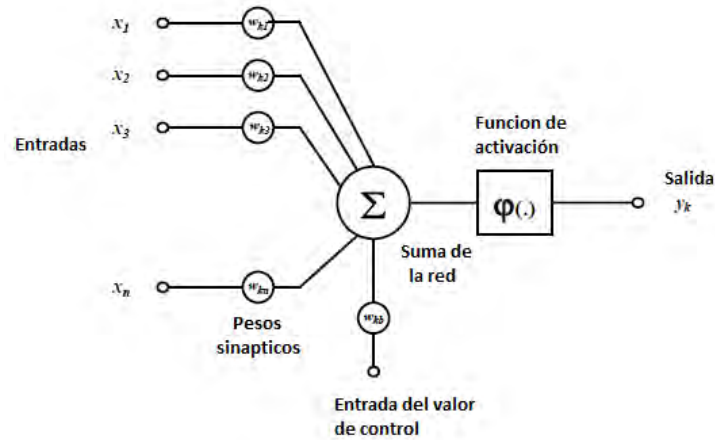


*Figura.-1.12.- Estructura de una neurona biológica*

Las redes neuronales son usadas principalmente para dos tareas, la primera es el reconocimiento de patrones y la segunda es para aproximaciones de datos, llamado comúnmente como modelado.

En la figura 1.13 se tiene la construcción del elemento más básico de una red neuronal artificial, la neurona recibe los datos de entrada  $x_i$ , realizando una suma pesada de estos datos junto con el valor de control, el resultado de esta suma se manda a una función de salida, que deberá activarse solo para ciertos valores.

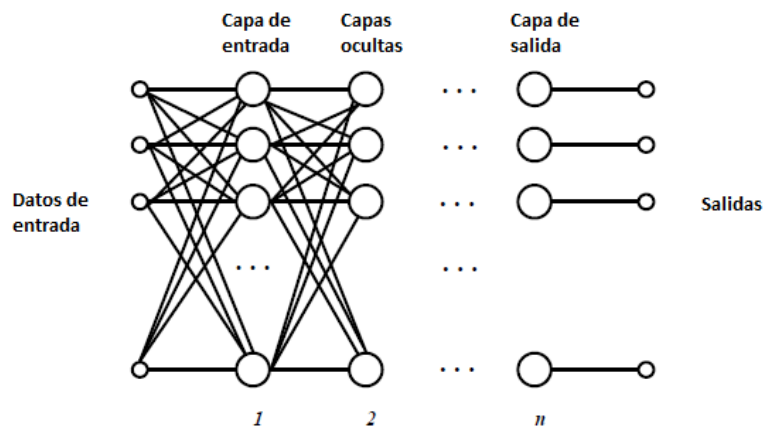
Las redes neuronales pueden ser divididas en estáticas y dinámicas, donde las redes dinámicas hacen referencia a una adaptación continua de sus parámetros y en el caso de las redes estáticas son entrenadas solo una vez y por tanto sus valores permanecen sin cambio alguno, ambas redes poseen una retroalimentación hacia adelante [11].



**Figura.-1.13.- Configuración básica de una neurona artificial**

En aproximaciones con la técnica clásica, es necesario conocer el modelo matemático básico que aproxima el problema, la propiedad más importante de las redes neuronales es su habilidad para aprender el modelo a partir de los datos presentados, estas pueden ser entrenadas ya sea por datos conocidos, lo cual se conoce como un entrenamiento supervisado y la otra forma es donde la red es entrenada sin ningún conocimiento de los datos, en este caso se habla del entrenamiento no supervisado.

La figura 1.14 muestra una red neuronal compuesta de  $n$  capas, donde cada capa contiene varias neuronas, este es un arreglo complejo que permitirá reconocer o modelar algún conjunto de datos de entrada, que ingresan a través de la capa de entrada y donde la capa de salida arroja la respuesta de la red neuronal artificial.



**Figura.-1.14.- Estructura de una red neuronal artificial.**



## CAPITULO II.- TEORIA BASICA

---

En este capítulo se introducirá a la teoría del modelo matemático aplicado al sistema de visión, se dará la descripción de la técnica de proyección de línea y los fundamentos matemáticos para el modelado de superficies.

El modelado de superficies es una forma viable de construir formas geométricas complicadas, la técnica aplicada permite un análisis matemático de la superficie recuperada, así se puede someter la forma generada a procesos de diseño que sean más versátiles y viables en procesos de ingeniería inversa.

La proyección de línea es un método que permite la recuperación de la topografía de un objeto de acuerdo al desplazamiento relativo de la línea, el desplazamiento es calculado a partir del procesamiento digital de las imágenes extraídas de la escena, el método permite una reconstrucción rápida y precisa de superficies.

La aplicación de técnicas de luz estructurada requiere de la caracterización del sistema de visión, esto permite una correspondencia entre la información contenida en el plano imagen del sistema de visión y los puntos de la superficie del objeto en análisis. La técnica de proyección de una línea laser es un método que requiere una calibración de los parámetros tanto intrínsecos como extrínsecos del sistema de visión, estos se calculan en base al modelo adaptado al sistema.

### 2.1 MODELADO DE LA SUPERFICIE CON FUNCIONES NURBS

Los NURBS son una técnica de modelado basada en el producto tensorial de funciones base B-Splines, la característica principal de este proceso es que se lleva a cabo en un plano dimensional más alto, es decir que las coordenadas son tomadas en su forma homogénea, esto ofrece una variable extra que permite una manipulación de las formas de las superficies, para poder representar un objeto real por medio de esta técnica se debe realizar un proceso de varias etapas que incluye: Un proceso de interpolación por medio de las funciones base, una parametrización de los puntos recuperados, un ajuste de pesos basado en un error por mínimos cuadrados y por último la evaluación de la función NURBS.

En las siguientes secciones se dará una descripción de las funciones NURBS y el proceso de la formación de las superficies, se describe el modelo matemático implementado para la cámara y en última instancia se presenta las bases para el procesamiento digital y recuperación de las formas geométricas.

### 2.1.1 funciones base de curvas y superficies B-Spline no uniformes y racionales (NURBS)

La forma más general de una curva B-Spline racional no uniforme se expresa como ecuación 2.1.

$$C(u) = \frac{\sum_{i=0}^n w_i * N_{i,k}(u) * P_i}{\sum_{i=0}^n w_i * N_{i,k}(u)} \quad a \leq u \leq b \quad \text{Ec.-2.1}$$

Para la curva NURBS, el parámetro  $k$  es el grado de la función base,  $P_i$  son los puntos del polígono de control,  $w_i$  son los pesos asociados a cada punto control  $i$  y  $N_{i,k}$  son las funciones base del modelo, la variable paramétrica queda limitada de acuerdo a la siguiente expresión:

$$0 \leq u \leq n - K + 2$$

La función B-Spline base no uniforme es definida recursivamente por la expresión 2.2, donde las funciones de grado 0 se calculan bajo los siguientes criterios:

$$N_{i,1}(u) = 1, \quad \text{si } t_i \leq u \leq t_{i+1}$$

$$N_{i,1}(u) = 0, \quad \text{con cualquier otro valor}$$

y

$$N_{i,k}(u) = \frac{(u-t_i)*N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)*N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}} \quad \text{Ec.-2.2}$$

Para valores enteros de  $k: k = 2, \dots, k$ . Los  $t_i$  son llamados "valores de nodo", y un conjunto de nodos componen un "vector de nodos", estos relacionan la variable paramétrica  $u$  a  $p_i$  que son los puntos control donde  $i = 0, \dots, n$ . Para una curva no uniforme abierta que interpola sus puntos finales, la  $t_i$  es calculada con los siguientes criterios:

$$t_j = 0 \quad \text{si } j < K$$

$$t_j = j - K + 1 \quad \text{si } K \leq j \leq n$$

$$t_j = 0 \quad \text{si } j > n$$

Para valores enteros de  $j: j = 0, \dots, n + K$ .

Los parámetros que determinan el número de puntos control, nodos y el grado del polinomio están relacionados por la ecuación 2.3.

$$n + K + 1 = T \quad \text{Ec.-2.3}$$

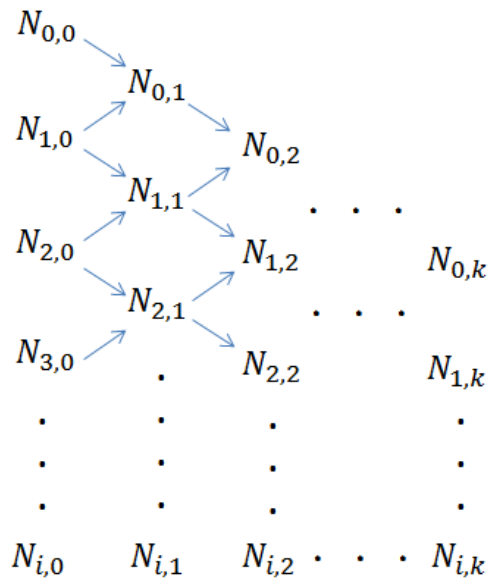
Donde  $T$  es el número de nodos,  $n$  es el número de puntos control y  $K$  el grado de la superficie.

La siguiente expresión muestra el vector de nodos  $\mathbf{t}$ , donde un espacio nodal es un rango de valores para  $u \in [t_i, t_{i+1})$ , en cada espacio nodal queda definido un trozo de curva y cada trozo es formado por diferentes funciones base  $N_{i,k}(u)$ . De aquí el nombre de B-SPLINES NO UNIFORMES.

$$\mathbf{t} = \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\} = \{t_0, t_1, \dots, t_i, t_{i+1}, \dots, t_T\}$$

La configuración de las funciones base no uniformes requieren un método de cálculo progresivo, la figura 2.1 muestra el gráfico donde se aprecia el orden para obtener las funciones base de grado superior, donde es necesario calcular primero las funciones de grado  $k = 0$ , hasta llegar al grado deseado para la función base del modelado.

Como se puede apreciar a medida que el grado  $k$  de una función se incrementa, el número de funciones base de grado menor involucradas es mayor y por tanto se necesitan más puntos control para definir la función en ese espacio nodal.



**Figura 2.1.- Orden del cálculo de las funciones base NURBS de grado K.**

La figura 2.2 muestra el cálculo de las funciones base de grado  $k = 0$  y grado  $k = 1$ , en este caso se puede ver que para  $k = 0$  solo una función base es activada a la vez y esto es controlado por la distribución del vector de nodos, esto se puede ver como una función de activación donde vale 1 para valores de la variable paramétrica  $u$  dentro del espacio nodal  $[t_i, t_{i+1})$  y donde vale cero para cualquier otro valor, para  $k = 1$  se tiene una combinación de las funciones de grado  $k = 0$  y por tanto se esperaría una función lineal, en la figura 2.9-b se tiene las gráficas para las funciones base de grado  $k = 1$ .

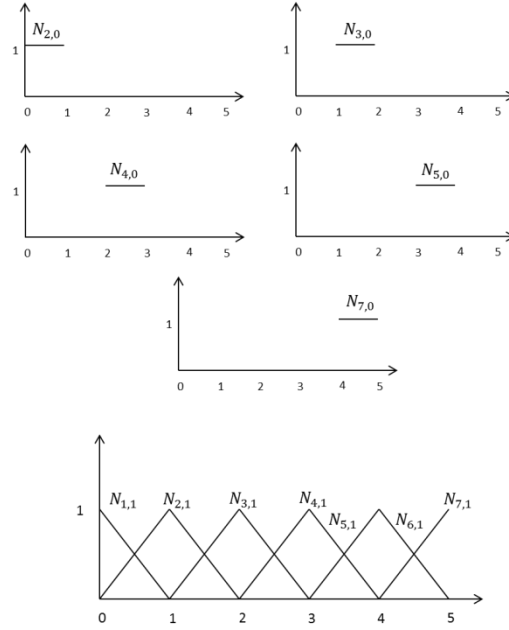


Figura 2.2.- Gráficas de las funciones NURBS (a) de grado  $k=0$ , (b) de grado  $k=1$ .

### 2.1.2 Superficies NURBS

Una superficie NURBS de grado  $n$  en la dirección  $u$  y grado  $m$  en la dirección  $v$ , es una función bivaluada y se muestra como ecuación 2.4.

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n w_{ij} P_{ij} N_{i,m}(u) N_{j,n}(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{ij} N_{i,m}(u) N_{j,n}(v)} \quad \text{Ec.-2.4}$$

Donde  $P_{ij}$  forma un polígono de control bidireccional que representa la superficie del objeto,  $w_{ij}$  son los pesos asociados a cada punto control,  $N_{i,m}(u)$  y  $N_{j,n}(v)$  son las funciones base B-Splines no racionales definidas por los vectores nodo  $U$  y  $V$  expresados como;

$$U = \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\}$$

$$V = \{0, \dots, 0, v_{p+1}, \dots, v_{r-p-1}, 1, \dots, 1\}$$

Asumiendo que:

$$R_{i,j}(u, w) = \frac{w_{ij} N_{i,m}(u) N_{j,n}(v)}{\sum_{d=0}^m \sum_{g=0}^n w_{dg} N_{d,m}(u) N_{g,n}(v)}$$

Introduciendo este término en la ecuación 2.20 que define una superficie NURBS, permite obtener una nueva expresión para la superficie (Ec. 2.5).

$$S(u, w) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j} = 1 \quad \text{Ec.-2.5}$$

### 2.1.3 Selección de los pesos $W$ en la expresión para una superficie NURBS

Una de las características importantes de los NURBS son los pesos  $w_{ij}$  definidos en cada punto control  $P_{ij}$ , tales pesos son debido a que se tiene un sistema dimensional más alto para definir los puntos control, es decir que las coordenadas están expresadas en forma homogénea y donde esto permite modificar la forma de la proyección de la línea a través de la superficie.

La expresión de una superficie NURBS en coordenadas homogéneas se puede ver como ecuación 2.6.

$$\mathbf{S}^w(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,m}(u) N_{j,n}(v) \mathbf{P}_{ij}^w \quad \text{Ec.-2.6}$$

Dónde  $\mathbf{P}_{ij}^w = (\mathbf{h}_{ij} \mathbf{x}_{ij}, \mathbf{h}_{ij} \mathbf{y}_{ij}, \mathbf{h}_{ij} \mathbf{z}_{ij}, \mathbf{h}_{ij})$ , este vector representa los puntos control en la forma homogénea, donde la superficie o puntos control serian una proyección de la superficie homogénea expresada como  $\mathbf{P}(u, v) = H\{\mathbf{P}^w(u, v)\}$ .

Para poder establecer los pesos  $w_{ij}$  es necesario un procedimiento de ajuste conocido como el "Surface fitting", donde se puede hacer un ajuste ya sea por un método de interpolación u los métodos de aproximación [12].

Para ajustar los pesos se establecen inicialmente todos los pesos  $w_{ij} = 1$ , esto para poder realizar un ajuste por métodos de interpolación, donde se da solución a un sistema lineal de la forma:

$$\mathbf{Q}_{k,l} = \sum_{i=0}^n N_{i,p}(\bar{u}_k) \mathbf{R}_{i,l} \quad \text{Ec.-2.7}$$

Dónde:

$$\mathbf{R}_{i,l} = \sum_{j=0}^m N_{j,q}(\bar{u}_l) \mathbf{P}_{i,j}$$

Una vez interpolada la superficie es necesario parametrizar los puntos  $Q_k$ , es decir que asociaremos la posición de este punto con sus coordenadas paramétricas  $(u, v)$ , donde el fin es localizar y poder ajustar los pesos referentes a estos puntos por medio del método de mínimos cuadrados[43], el resultado será una superficie aproximada a la forma real del objeto, en la figura 2.3 se muestra una superficie NURBS con sus respectivos puntos control.

Algunas propiedades de las superficies generadas con NURBS son:

- La no negatividad:  $R_{i,j}(u, v) \geq 0$  para todo  $i, j, u$  y  $v$ .
- Partición de la unidad:  $\sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j} = 1$  para todo  $(u, v) \in [0,1] \times [0,1]$
- Soporte local  $R_{i,j}(u, v) = 0$  si  $(u, v)$  esta afuera del rectángulo dado por  $[u_i, u_{i+L+1}] \times [v_j, v_{j+k+1}]$ ;
- $R_{0,0}(0,0) = R_{n,0}(1,0) = R_{0,m}(0,1) = R_{n,m}(1,1) = 1$
- Modificación local: si  $\mathbf{P}_{i,j}$  es modificado, o  $w_{i,j}$  es variado, esto solo afectara la forma de la superficie en el rectángulo comprendido por:  $[u_i, u_{i+L+1}] \times [v_j, v_{j+k+1}]$ ;

- B-Splines no racionales, Bézier y superficies de Bézier racionales son casos especiales de las superficies NURBS.
- Diferenciabilidad:  $S(u, v)$  es  $m - k(n - k)$  veces diferenciable con respecto a  $u$  ( $v$ ) en el nodo  $u$  (nodo  $v$ ) de multiplicidad  $k$ .

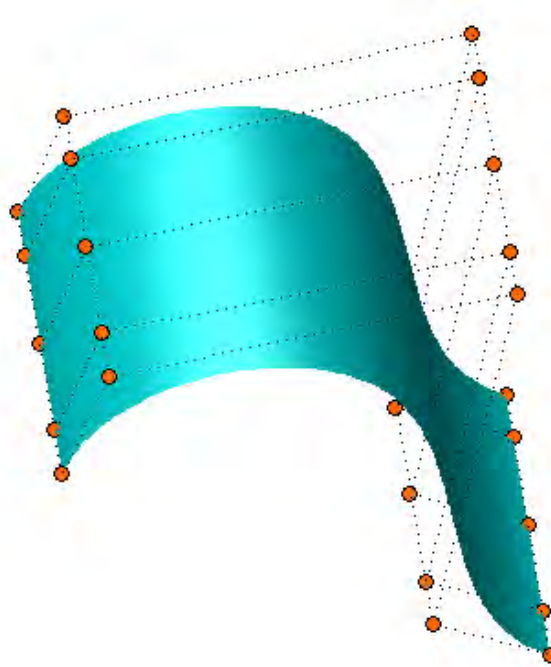


Figura 2.3.- Imagen de una superficie construida mediante NURBS.

#### 2.1.4 Diferenciabilidad en los NURBS

En particular tenemos que la derivada de la función base  $N_{i,m}(u)$  (Formula cox, De Boor) respecto de  $u$  queda expresada como:

$$N'_{i,k}(u) = \frac{k \cdot N_{i,k-1}(u)}{t_{i+k} - t_i} - \frac{k \cdot N_{i+1,k-1}(u)}{t_{i+k+1} - t_{i+1}} \quad \text{Ec.-2.8}$$

Donde se asume que la ecuación 2.8 es verdadera para  $k - 1, k > 1$ . Usando la regla del producto para derivar la función base, se puede obtener una expresión general para la derivada de la función (Ec.2.9) [12].

$$N_{i,k}^d(u) = k \left( \frac{N_{i,k-1}^{d-1}}{u_{i+k} - u_i} - \frac{N_{i+1,k-1}^{d-1}}{u_{i+k+1} - u_{i+1}} \right) \quad \text{Ec.-2.9}$$

La derivada de una curva en su forma más general queda expresada como ecuación 2.10.

$$C^{(d)}(u) = \sum_{i=0}^n N^{(d)}_{i,k}(u) * P_i \quad \text{Ec.-2.10}$$

Dada una curva NURBS como:

$$C(u) = \frac{\sum_{i=0}^n w_i * N_{i,k}(u) * P_i}{\sum_{i=0}^n w_i * N_{i,k}(u)} = \frac{A(u)}{w(u)} \quad \text{Ec.-2.11}$$

La derivada de la ecuación 2.11 se expresa como:

$$C^{(k)}(u) = \frac{A^{(k)}(u) - \sum_{i=1}^k \binom{k}{i} * w^{(i)}(u) * C^{(k-i)}(u)}{w(u)} \quad \text{Ec.-2.12}$$

### Derivada de una superficie NURBS

La derivada de una superficie en su forma más general queda expresada como:

$$\frac{\partial^{k+l}}{\partial^k u \partial^l w} S(u, w) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}^{(k)}(u) * N_{j,q}^{(l)}(w) * P_{i,j} \quad \text{Ec.-2.13}$$

De una superficie NURBS tenemos:

$$S(u, w) = \frac{\sum_{i=0}^m \sum_{j=0}^n w_{ij} P_{ij} N_{i,m}(u) N_{j,n}(w)}{\sum_{i=0}^m \sum_{j=0}^n w_{ij} N_{i,m}(u) N_{j,n}(w)} = \frac{A(u, w)}{w(u, w)}$$

La derivada de la superficie NURBS queda definida como:

$$S^{(k,l)} = \frac{1}{w} \left( A^{k,l} - \sum_{i=1}^k \binom{k}{i} w^{(i,0)} S^{(k-i,l)} - \sum_{j=1}^l \binom{l}{j} w^{(0,j)} S^{(k,l-j)} - \sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} w^{(i,j)} S^{(k-i,l-j)} \right)$$

**Ec.-2.14**

#### 2.1.5 Continuidad en los NURBS

La continuidad geométrica de una superficie se refiere principalmente a la forma de la superficie resultante, ya que las superficies NURBS son funciones, es posible realizar las derivadas de la superficie respecto a las variables paramétricas, por tanto se tendría una continuidad paramétrica.

Una continuidad paramétrica de un cierto grado, implica una continuidad geométrica de ese grado, la definición de una continuidad  $C_n$  necesita que la  $n^{th}$  derivada de la curva o superficie sea igual en el punto de unión, estas derivadas pueden ser vistas como vectores que poseen dirección y magnitud.

Los NURBS poseen la capacidad de ofrecer continuidades  $G0$ ,  $G1$  y  $G2$  (figura 2.4), estas son una medida de que tan suave dos curvas se unen en un punto específico, mientras mayor sea el grado de la continuidad, la superficie tendrá una apariencia más lisa y sin cambios bruscos.

La continuidad posicional  $G0$  establece que cualesquiera que sean los dos puntos finales de dos curvas, estos deben coincidir. Sin embargo las curvas o superficies aun tendrán un ángulo relativo en esta unión, dando lugar a cambios bruscos en la continuidad de la curva.

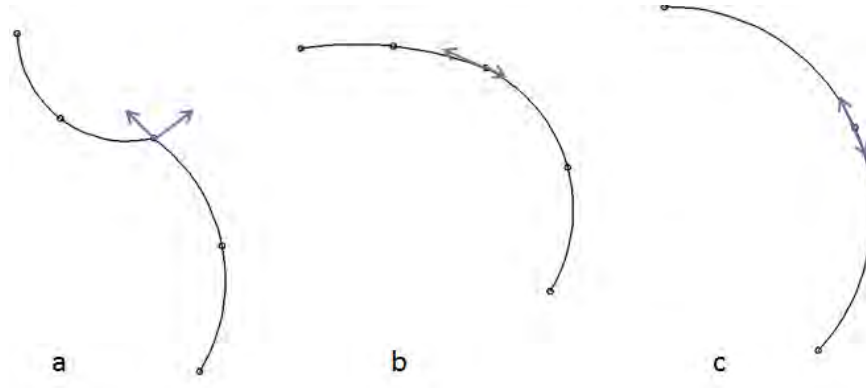


Figura 2.4.- Continuidades en curvas: (a) continuidad  $G0$ , (a) continuidad  $G1$ , (a) continuidad  $G2$ .

Para obtener una continuidad tangencial  $G1$ , se requiere que los vectores tangenciales en los puntos finales de las curvas sean paralelos, evitando así uniones agudas o bruscas. Debido a que las características obtenidas con la continuidad tangencial son suficientes para mantener una buena continuidad o una continuidad natural, este grado de continuidad es el más común en el modelado.

Por ultimo para la continuidad de curvatura  $G2$ , se requiere que los vectores en los puntos finales de dos curvas sean paralelos y además tales vectores deben tener proporciones longitudinales y grados de cambio equivalentes. Las curvas con esta continuidad no presentan grandes cambios, causando que las diferentes curvas parezcan una misma curva, esto se aprecia en una superficie como un perfecto difuminado o "smoothing".

Para verificar las continuidades en los NURBS vamos a desarrollar la ecuación 2.14 y así también calcular sus derivadas, demostrando así las continuidades presentes en esta función.

$$C(u, v_1) = \sum_{i=0}^n N_{i,k}(u) * P_i^w \quad \text{Ec.-2.14}$$

En este caso la ecuación 2.30 representa una curva evaluada en una superficie NURBS, esta servirá como ejemplo para demostrar la continuidad en una superficie NURBS,  $P_i^w$  son las coordenadas homogéneas del polígono de control, en estas está inmerso los pesos asociados a los puntos control.

Teniendo una vector de nodos como:

$$t = \{0,0,0,1,2,3,4,5,5,5\}$$

Donde el vector de puntos control homogéneo se define como:

$$P^w = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$$

Desarrollamos las funciones base NURBS de grado 3 para obtener la expresión matemática que define la curva descrita por la ecuación 2.14.



$$N_{0,3} = (1 - u)^3 \quad u \in [0,1) \quad \text{y} \quad N_{0,3} = 0 \quad u \in [1,2)$$

$$N_{1,3} = 3u - \frac{9}{2}u^2 + \frac{7}{4}u^3 \quad u \in [0,1) \quad \text{y} \quad N_{1,3} = 2 - 3u + \frac{3}{2}u^2 - \frac{1}{4}u^3 \quad u \in [1,2)$$

$$N_{2,3} = \frac{3}{2}u^2 - \frac{11}{12}u^3 \quad u \in [0,1) \quad \text{y} \quad N_{2,3} = -\frac{3}{2} + \frac{9}{2}u - 3u^2 + \frac{7}{12}u^3 \quad u \in [1,2)$$

$$N_{3,3} = \frac{1}{6}u^3 \quad u \in [0,1) \quad \text{y} \quad N_{3,3} = \frac{2}{3} - 2u + 2u^2 - \frac{3}{6}u^3 \quad u \in [1,2)$$

$$N_{4,3} = 0 \quad u \in [0,1) \quad \text{y} \quad N_{4,3} = \frac{1}{2}(u - 1)^3 \quad u \in [1,2)$$

Evaluando para  $u_1 \in [0,1)$  y  $u_2 \in [1,2)$  la ecuación 2.14, obtenemos las expresiones de ambos segmentos de curva.

$$\mathbf{S}(u_1) = (1 - u)^3 * \mathbf{P}_0 + \left(3u - \frac{9}{2}u^2 + \frac{7}{4}u^3\right) * \mathbf{P}_1 + \left(\frac{3}{2}u^2 - \frac{11}{12}u^3\right) * \mathbf{P}_2 + \left(\frac{1}{6}u^3\right) * \mathbf{P}_3 \quad \text{Ec.-2.15}$$

$$\mathbf{C}(u_2) = \left(2 - 3u + \frac{3}{2}u^2 - \frac{1}{4}u^3\right) * \mathbf{P}_1 + \left(-\frac{3}{2} + \frac{9}{2}u - 3u^2 + \frac{7}{12}u^3\right) * \mathbf{P}_2 + \left(\frac{2}{3} - 2u + 2u^2 - \frac{3}{6}u^3\right) * \mathbf{P}_3 + \frac{1}{2}(u - 1)^3 * \mathbf{P}_4 \quad \text{Ec.-2.16}$$

Estas curvas convergerán cuando las variables paramétricas  $u_1$  y  $u_2$  tengan un valor de 1.

Ahora se tiene que para obtener una continuidad  $C_0$ , las dos curvas deben converger en un punto y por tanto se debe satisfacer que  $\mathbf{S}(u_1) = \mathbf{C}(u_2)$  para  $u_1 = u_2 = 1$ , evaluando las ecuaciones 2.31 y 2.32 se tiene:

$$\mathbf{S}(1) = \left(\frac{1}{4}\right) * \mathbf{P}_1 + \left(\frac{7}{12}\right) * \mathbf{P}_2 + \left(\frac{1}{6}\right) * \mathbf{P}_3$$

$$\mathbf{C}(1) = \left(\frac{1}{4}\right) * \mathbf{P}_1 + \left(\frac{7}{12}\right) * \mathbf{P}_2 + \left(\frac{1}{6}\right) * \mathbf{P}_3$$

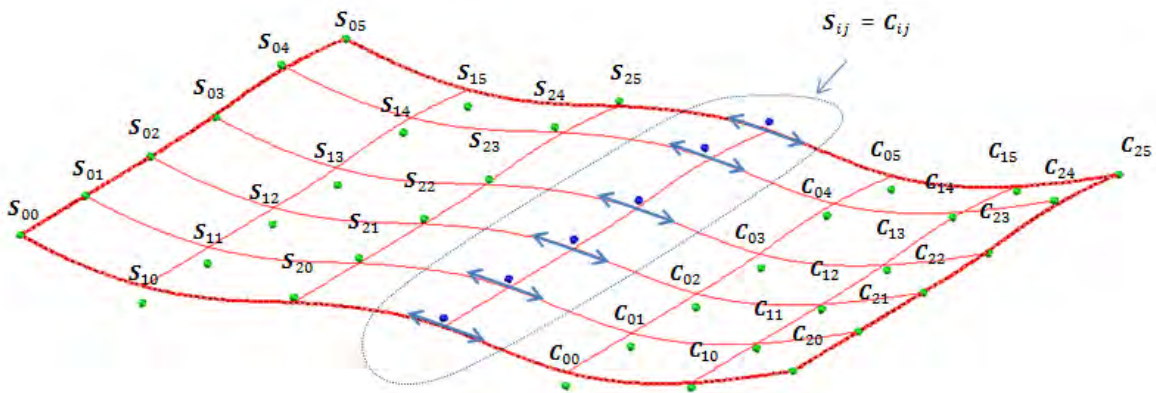
Por tanto se cumple el criterio de convergencia y se tiene una continuidad  $C_0$ , que a su vez lleva asociada la continuidad geométrica  $G_0$ , que está relacionada a la forma del objeto.

La continuidad paramétrica  $C_1$  requiere que además de la convergencia de ambas curvas, los vectores tangentes en el extremo de unión deben ser paralelos (figura 2.5), ya que una derivada parcial representa tal vector se debe cumplir que  $\frac{\partial \mathbf{C}}{\partial u_2} = \frac{\partial \mathbf{S}}{\partial u_1}$ , desarrollando las derivadas para ambas curvas y evaluando para  $u_1 = u_2 = 1$ .

$$S'(1) = \left(-\frac{3}{4}\right) * P_1 + \left(\frac{1}{4}\right) * P_2 + \left(\frac{3}{6}\right) * P_3$$

$$C'(1) = \left(-\frac{3}{4}\right) * P_1 + \left(\frac{1}{4}\right) * P_2 + \left(\frac{1}{2}\right) * P_3$$

Como se puede observar, las derivadas de ambos segmentos de curva son iguales y por tanto se tiene una continuidad  $C_1$ , que se asocia con la continuidad geométrica  $G1$ .



**Figura 2.5.- Imagen de una superficie NURBS compuesta por las superficies S y C, donde se aprecia los vectores tangentes en la unión.**

Por último para demostrar que los NURBS poseen una continuidad  $G2$ , se debe comprobar que la razón de cambio en la curvatura de ambos segmentos  $S$  y  $C$  debe ser igual, por tanto se debe cumplir  $\frac{\partial^2 C}{\partial u_2^2} = \frac{\partial^2 S}{\partial u_1^2}$  para  $u_1 = u_2 = 1$ , desarrollando la segunda derivada en ambos segmentos de curva y evaluando con  $u_1 = u_2 = 1$ .

$$S''(1) = \left(\frac{3}{2}\right) * P_1 + \left(-\frac{5}{2}\right) * P_2 + P_3$$

$$C''(1) = \left(\frac{3}{2}\right) * P_1 + \left(-\frac{5}{2}\right) * P_2 + P_3$$

Como se puede ver se cumple la igualdad y queda comprobado que los NURBS dan una continuidad  $G2$ , que se relaciona a una superficie geométrica de trazos alisados y que pareciera ser una sola superficie.

Ahora que se ha comprobado que una superficie NURBS de grado  $p$  posee una continuidad que se relaciona con la derivada de la función de orden  $p + 1$ , se tiene de las características elementales de las funciones base que conforme se incrementa la continuidad de la superficie, esta tiende a interpolar los puntos control de una forma menos aproximada, debido a las restricciones presen-

tes en las funciones de grado más elevado, los trazos bruscos en la superficie tienden a alisarse, creando de esta forma una superficie continua y con cambios suaves.

Ya que los NURBS pueden aproximarse o interpolar un conjunto de puntos a través de los pesos, es de gran relevancia la correcta parametrización de los vectores de nodos, ya que estos definirán que tan exacta será la aproximación de la superficie modelada a la forma real del objeto.

En general se toma un modelado con funciones NURBS de grado 3, ya que muestran buen comportamiento en la interpolación de los datos y además posee una continuidad  $G2$  que ofrece un buen modelado de la superficie, además que se logra filtrar el ruido presente en los datos recuperados.

Con estas consideraciones se pretende realizar un modelado que sea exacto y además posea las características de continuidad geométrica y paramétrica que se buscan para la virtualización del objeto.

## 2.2 MODELADO DEL SISTEMA DE VISIÓN

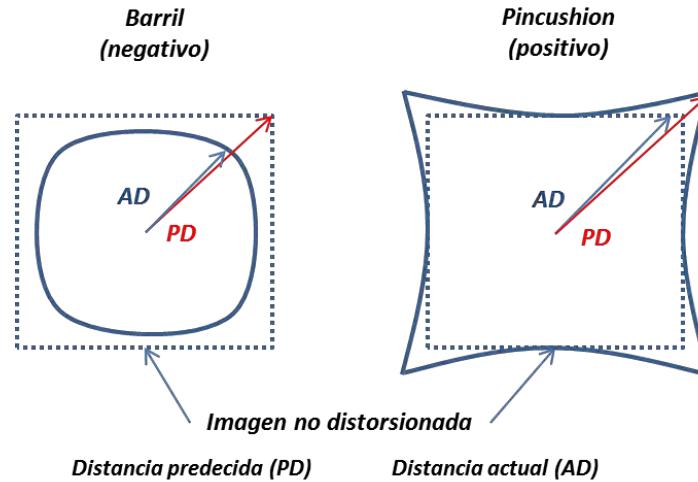
La mejor aproximación para describir el comportamiento real en un sistema óptico puede ser lo- grado por medio del uso de la perspectiva de transformación proyectiva la cual puede ser descrita por una ecuación lineal, esto en un espacio dimensional más alto el cual es llamado coordenadas homogéneas. Adicionalmente cuando se describen elementos ópticos reales la transformación proyectiva simple ha de ser complementada con términos no lineales, esto con el fin de tomar los factores físicos que influyen en la obtención de una imagen.

### 2.2.1 Distorsión de la lente

Ya que el modelo matemático aplicado a la cámara sugiere una transformación donde un punto 3D es proyectado idealmente al plano imagen a través de un centro óptico, es necesario corregir las distorsiones presentes en el sistema. Estos errores ópticos causados por la lente generan diferencias en la magnificación de los puntos pertenecientes a la imagen, donde el resultado es una especie de desplazamiento radial de los puntos relativo al centro de la imagen.

Aunque la distorsión existe en todas las lentes estas pueden ser corregidas con gran precisión, la distorsión entonces son aberraciones geométricas, esto significa que la información del objeto no se pierde sino que solo está fuera de lugar en el plano imagen.

En la figura 2.6 se muestran los tipos de distorsiones presentes en una imagen y la forma de presentar la cantidad de distorsion de una camara.



**Figura 2.6.- Representación esquemática de las distorsiones presentes en las lentes, imagen tomada de [35].**

Para lograr corregir este problema es necesario medir la distorsión presente en la imagen, esto se logra comparando la imagen distorsionada e ideal de un objeto calibrador cuya forma es conocida, con la información recabada se corrige la distorsión por medio de software, logrando con esto una imagen más acorde con el modelo de proyección de perspectiva.

Otra forma de corregir la distorsión radial es ajustando la distorsión por medio de una función matemática no lineal [13], la cual modela el desplazamiento radial sufrido por cada punto de la imagen, en la sección 2.1.4 se presenta la función matemática aplicada al modelo de proyección de perspectiva.

Contemplado esto ahora se puede desarrollar el modelo de proyección de perspectiva donde los parámetros intrínsecos y extrínsecos corresponden a factores propios al sistema de visión.

### 2.2.2 Modelado de la Cámara de proyección de perspectiva

En este modelo existe una correspondencia entre cada área 2D en el plano imagen y el área en la escena 3D, por medio de este mecanismo es construida o proyectada una imagen desde el espacio real a un plano imagen. Un modelo matemático simple para la cámara de proyección de perspectiva se muestra en la figura 2.7.

Ahora se explicara el modelo matemático para poder conocer los parámetros físicos involucrados en la formación de la imagen.

Teniendo los puntos:  $P = [X, Y, Z]^T$  y  $p = [x, y, z]^T$

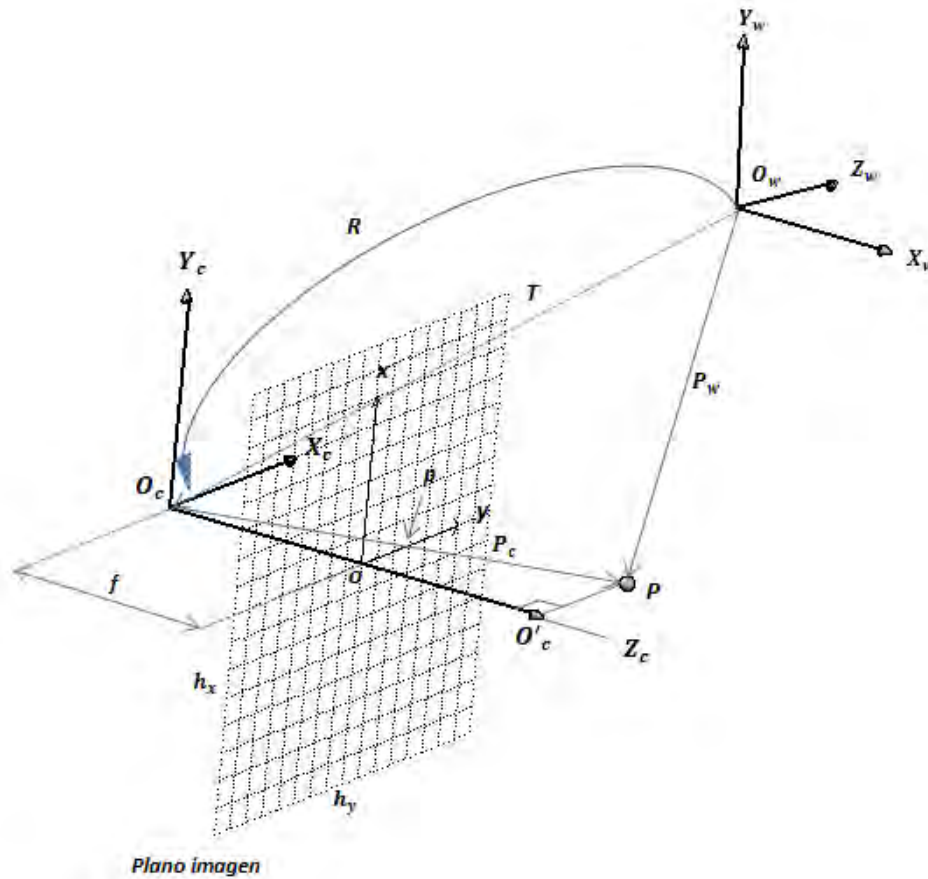


Fig. 2.7.-Esquema del modelo de la cámara Pin-hole.

Donde  $p$  es la imagen del punto  $P$  que se ubica en una escena 3D del mundo real, tomando estos puntos se deduce por triángulos semejantes que:

$$(x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}, \quad z = f)$$

Estas expresiones constituyen la base para la fundamentación del modelo "Pin-hole" de una cámara. Con esto podemos mencionar que el modelo para una cámara real queda definido también por los parámetros extrínsecos e intrínsecos del sistema de visión.

### 2.2.3 Los parámetros extrínsecos

La descripción de una escena depende del sistema coordenado elegido, con el sistema coordenado establecido y la ubicación del plano imagen se puede conocer exactamente la ubicación del sistema coordenado de la cámara, por tanto se tendrán tres sistemas coordenados: el sistema coordenado del plano imagen, el sistema coordenado de la cámara y el sistema coordenado del mundo real o "world". Un cambio del sistema coordenado de la cámara al sistema world se llevara a cabo por medio de una traslación  $T$  y una rotación  $R$ , la expresión matemática queda como:

$$P_c = R(P_w - T)$$

La matriz de rotación y la de traslación pueden ser representada como:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}_{3 \times 3} \quad y \quad T = O_w - O_c = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

Con esto podemos conocer que la totalidad de los parámetros extrínsecos que están relacionados con las transformaciones geométricas presentes en la transformación proyectiva del modelo Pin-hole.

#### 2.2.4 Parámetros intrínsecos

Los parámetros de la transformación proyectiva, en el caso del modelo de cámara Pin-hole son calculados a partir de la distancia focal  $f$ .

Los parámetros que mapean el sistema coordenado de la cámara en el sistema coordenado del plano imagen, en este caso sería el origen  $o(o_x, o_y)$ , que es el punto central del plano imagen, las dimensiones físicas de los pixeles presentes en la cámara  $(h_x, h_y)$ , la relación entre coordenadas imagen  $(x, y)$  y las coordenadas de la cámara  $(x_c, y_c)$  están relacionadas por medio de la siguiente relación matemática:

$$x_c = (x - o_x) * h_x \quad , \quad y_c = (y - o_y) * h_y$$

Donde las coordenadas  $(x_c, y_c)$  son coordenadas relacionadas con el sistema coordenado de la cámara y los demás parámetros están relacionados al sistema coordenado del plano imagen.

Las distorsiones geométricas son otro parámetro importante a tener en cuenta, este tipo de parámetro se da debido a los factores físicos que se involucran en la formación de imágenes por medio de sistemas ópticos, el modelo matemático más generalizado es el siguiente:

$$x_v = \frac{x}{1 + k_1 r^2 + k_2 r^4} \quad , \quad y_v = \frac{y}{1 + k_1 r^2 + k_2 r^4}$$

Donde  $r^2 = x_v^2 + y_v^2$  es llamada distorsión radial,  $K_1$  y  $K_2$  son parámetros que se han de calcular.

#### 2.2.5 Transformación proyectiva

Para poder proyectar un punto  $P$  sobre el plano imagen y así obtener un punto  $p$  es necesaria una matriz de proyección, la cual es una expresión matemática lineal que representa la cámara Pin-hole, la ecuación es:

$$p = MP$$

Donde  $M$  es la matriz de proyección y está compuesta por los parámetros intrínsecos y extrínsecos, por tanto la matriz de proyección puede ser descrita por:

$$M = M_i M_e$$

Dónde:

$$M_i = \begin{bmatrix} \frac{f}{h_x} & S & o_x \\ 0 & \frac{f}{h_y} & o_y \\ 0 & 0 & 1 \end{bmatrix}, \quad M_e = \begin{bmatrix} R_1 & -R_1 T \\ R_2 & -R_2 T \\ R_3 & -R_3 T \end{bmatrix}$$

Por tanto para proyectar un punto de una escena 3D en un plano imagen es necesario aplicar estas matrices y el proceso queda descrito por la siguiente relación matemática:

$$p = \begin{bmatrix} x_{uh} \\ y_{uh} \\ z_{uh} \end{bmatrix} = \begin{bmatrix} \frac{f}{h_x} & S & o_x \\ 0 & \frac{f}{h_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} R_1 & -R_1 T \\ R_2 & -R_2 T \\ R_3 & -R_3 T \end{bmatrix} * P$$

Donde  $P = [P_w \ 1]^T$  y este es un punto expresado en coordenadas homogéneas, también el punto imagen creado  $p$  es expresado en coordenadas homogéneas y la relación entre las coordenadas imagen y sus coordenadas homogéneas se expresan como:

$$x_u = \frac{x_{uh}}{z_{uh}}, \quad y_u = \frac{y_{uh}}{z_{uh}}$$

### 2.2.6 Calibración clásica del sistema de visión

Los métodos clásicos para la calibración de sistemas ópticos son aquellos en los que se analiza un patrón de calibración, estos deben contar con un entramado que posea alto contraste, esto para poder ubicar puntos y así poder localizarlos en la formación de la imagen, otra característica de estos patrones de calibración es que deben ser conocidas las dimensiones de todo su entramado para poder generar las ubicaciones de los puntos en el sistema coordenado de la escena 3D [13].

Las expresiones matemáticas para poder calcular los parámetros extrínsecos e intrínsecos son las siguientes:

$$x_u = \frac{x_{uh}}{z_{uh}} = \frac{p_1}{p_3} = \frac{m_{11}X_1 + m_{12}X_2 + m_{13}X_3 + m_{14}}{m_{31}X_1 + m_{32}X_2 + m_{33}X_3 + m_{34}} \quad \text{Ec.-2.17}$$

$$y_u = \frac{y_{uh}}{z_{uh}} = \frac{p_2}{p_3} = \frac{m_{21}X_1 + m_{22}X_2 + m_{23}X_3 + m_{24}}{m_{31}X_1 + m_{32}X_2 + m_{33}X_3 + m_{34}} \quad \text{Ec.-2.18}$$

Estas son las expresiones de las coordenadas para los puntos imagen. Despejando las ecuaciones 2.17 y 2.18 podemos llegar a la siguiente expresión:

$$Qm = 0$$

Donde la matriz Q se expresa como:

$$Q = \begin{bmatrix} X_{11} & X_{12} & X_{13} & 1 & 0_{1 \times 4} & -x_{u1}X_{11} - x_{u1}X_{12} - x_{u1}X_{13} - x_{u1} \\ & \dots & \dots & \dots & X_{11} & X_{12} & X_{13} & 1 & -y_{u1}X_{11} - y_{u1}X_{12} - y_{u1}X_{13} - y_{u1} \\ & & 0_{1 \times 4} & & \dots & \dots & \dots & \dots & \dots \\ & & & & X_{N1} & X_{N2} & X_{N3} & 1 & -y_{uN}X_{N1} - y_{uN}X_{N2} - y_{uN}X_{N3} - y_{uN} \end{bmatrix}$$

El vector **m** se expresa como:

$$m = [m_{11}, \dots, m_{14}, \dots, m_{21}, \dots, m_{24}, \dots, m_{31}, \dots, m_{34}]^T$$

Por último la solución para este sistema viene dada por la descomposición *SVD* [15] de la forma:

$$Q = SVD^T$$

## 2.3 CALIBRACION DEL ARREGLO OPTICO

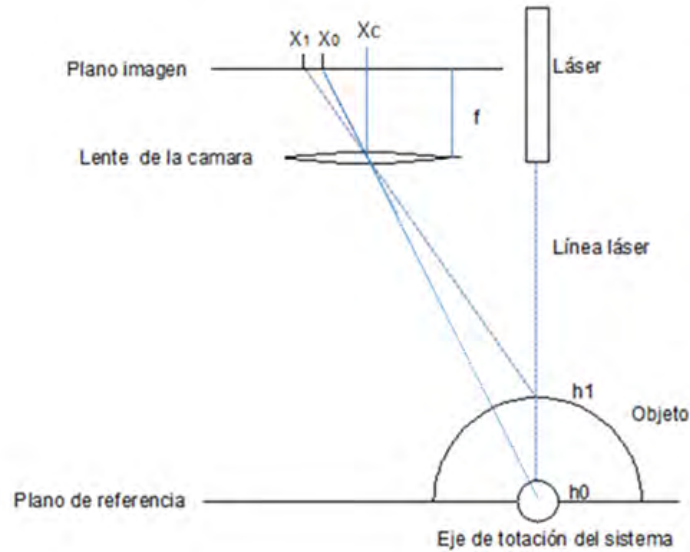
En este trabajo se utilizó la proyección de línea para la recuperación de las dimensiones de una superficie, para aplicar esta técnica se integró una cámara CCD, un proyector láser y el control correspondiente para la captura y desplazamiento del sistema.

El sistema de visión puede tener varias configuraciones geométricas posibles: una primera alineación es cuando el sistema puede tener tanto la cámara como el diodo láser alineados a un ángulo relativo, otra configuración posible es que el diodo láser es posicionado de tal forma que la proyección incide perpendicularmente sobre el plano de referencia y la cámara por su parte se posiciona oblicuamente respecto al proyector, en una tercera configuración la cámara es posicionada con su eje óptico perpendicular al plano de referencia, mientras el proyector es posicionado en un ángulo oblicuo respecto a la cámara.

### 2.3.1 Arreglo óptico implementado

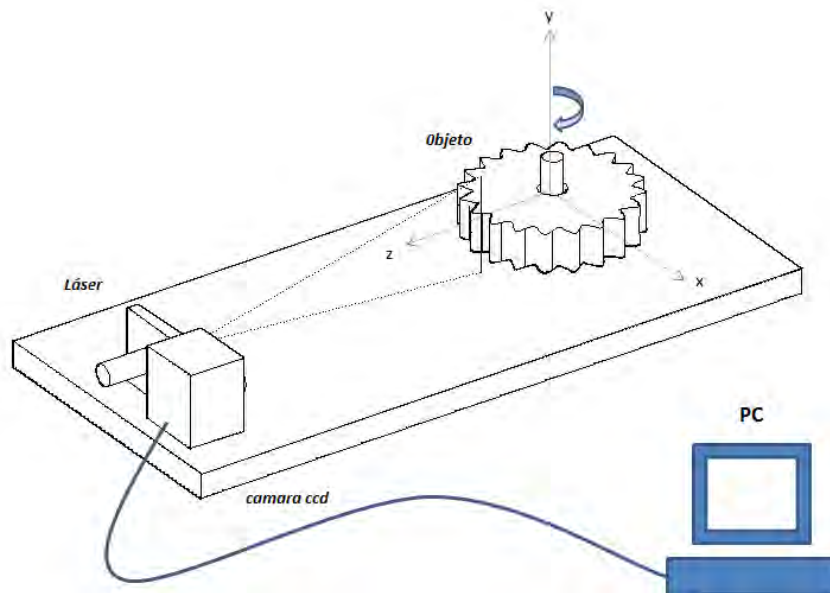
Para el escaneo de los objetos se implementó el arreglo óptico con una configuración donde el diodo laser es posicionado sobre el objeto y la proyección de este incide perpendicularmente sobre un plano referencia, la cámara por su parte es posicionada a un ángulo relativo al proyector, la figura 2.8 esquematiza la configuración geométrica usada.





**Figura. 2.8.- Geometría del arreglo óptico implementado**

En este sistema el objeto es fijado sobre una plataforma que gira sobre un eje de rotación (figura 2.9), para recuperar la topografía una línea vertical es proyectada sobre la superficie del objeto, al ser capturada la imagen de la proyección por la cámara CCD, la línea sufre un desplazamiento relativo que será proporcional a la topografía del objeto escaneado. El desplazamiento de la línea láser posee una relación geométrica con la topografía del objeto (alturas), por medio de procesos computacionales y algoritmos matemáticos las posiciones de la línea nos ayudan a calcular las coordenadas  $x$ ,  $y$  e  $z$  de la superficie del objeto.



**Figura2.9.- Esquema simplificado del sistema óptico implementado.**

La información obtenida de cada una de las imágenes de la línea láser corresponden a una sección transversal de la superficie del objeto escaneado, estos datos son almacenados en arreglos de memoria para posteriormente obtener la topografía del objeto.

Para describir la relación geométrica del desplazamiento de la línea y la altura del objeto tomaremos de referencia el esquema mostrado en la figura 2.10, el plano de referencia se encuentra ubicado a una distancia  $D$  respecto al sistema de visión, el plano de la imagen corresponde al sensor de la cámara CCD, las alturas del objeto se muestran como  $h_i$ ,  $l_a$  es distancia entre el eje óptico de la cámara y la proyección láser,  $f$  es la distancia focal de la lente y  $X_c$  corresponde al centro de la imagen.

Cuando la línea láser es proyectada sobre la superficie del objeto esta se ve reflejada hacia la cámara donde el sistema óptico enfoca la línea sobre el sensor CCD (plano imagen), por tanto las reflexiones de las alturas  $h_i$  generan las proyecciones  $x_i$  mostradas en el plano imagen.

La referencia para calcular el desplazamiento de la línea será  $x_0$ , que corresponde al plano de referencia, por tanto cualquier desplazamiento de la línea láser queda definido por:

$$S(x, y) = x_i - x_0 \quad \text{Ec.-2.19}$$

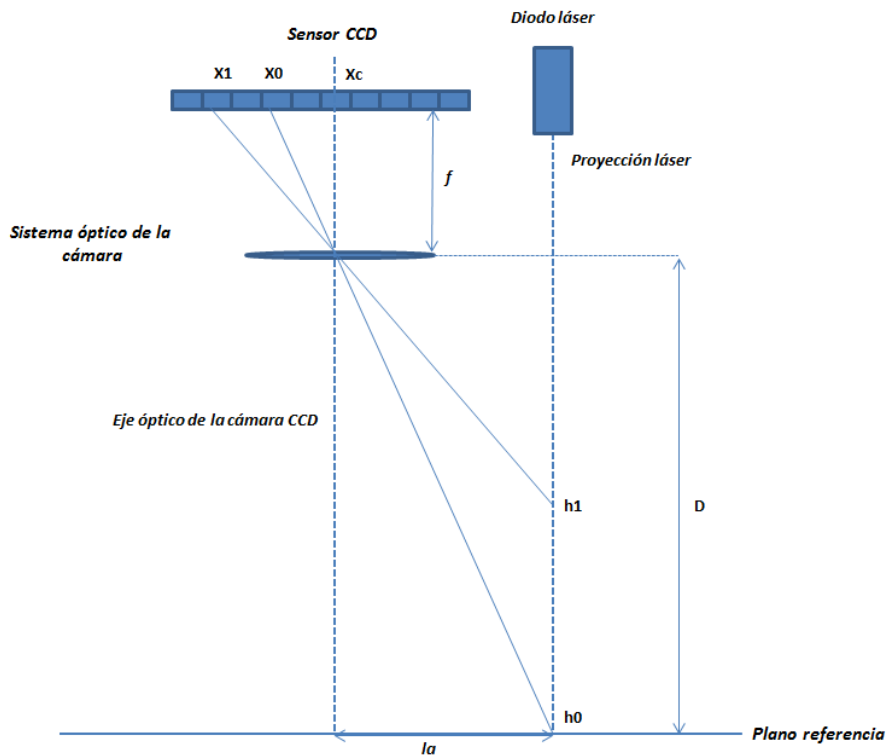


Figura 2.10.- Esquema de la geometría para una proyección de línea láser.

Las alturas  $h_i$  del objeto son calculadas a partir de las variables extrínsecas e intrínsecas del modelo Pin-hole de una cámara, la figura 2.11 muestra la relación geométrica para  $h_i$  y las variables del modelo.

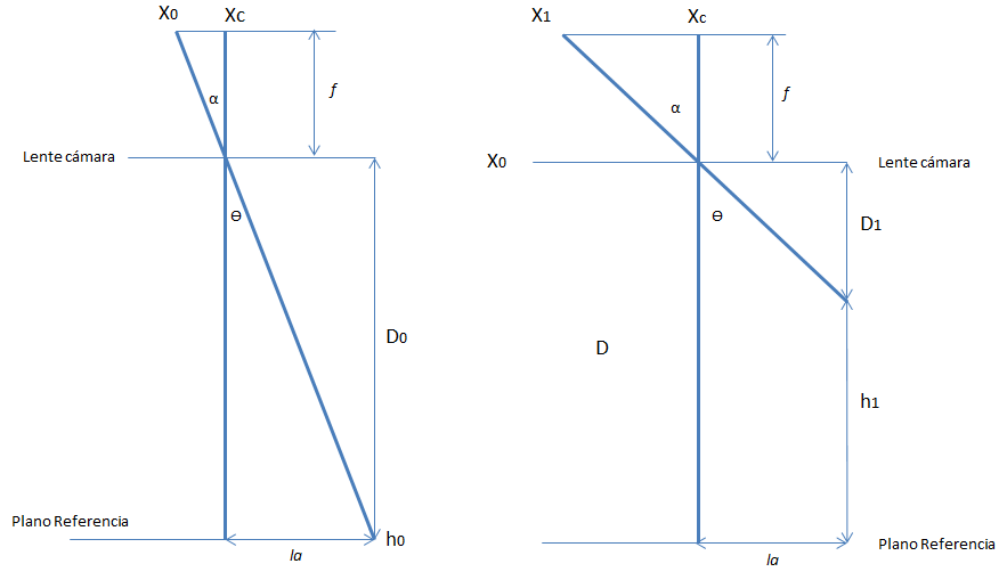


Figura 2.11.- Esquema que muestra las relaciones geométricas entre las alturas  $h_i$  y las variables extrínsecas e intrínsecas.

Por triángulos semejantes se tiene la ecuación 2.20.

$$\tan \alpha = \frac{\eta(x_c - x_0)}{f} = \frac{l_a}{D_o} = \tan \theta \quad \text{Ec.-2.20}$$

Ahora que tenemos la ecuación 2.4 la desarrollamos para  $f$  y  $l_a$  obteniendo la ecuación 2.21, realizando el mismo proceso para  $h_1$  se puede llegar a la ecuación 2.22.

$$f * l_a = \eta(x_c - x_0) * D_o \quad \text{Ec.-2.21}$$

$$f * l_a = \eta(x_1 - x_c) * (D_o - h_1) \quad \text{Ec.-2.22}$$

Igualando las ecuaciones 2.21 y 2.22 se llega a la ecuación 2.23, la cual se desarrolla para  $h_i$  y así se obtiene la expresión 2.24 que permite calcular la altura  $h_1$  del objeto.

$$(x_c - x_0) * D_o = (x_1 - x_c) * (D_o - h_1) \quad \text{Ec.-2.23}$$

$$h_1 = D_o - \frac{(x_c - x_0) * D_o}{x_1 - x_c} \quad \text{Ec.-2.24}$$

La altura  $h_i$  de la superficie queda expresada de forma general como:

$$h_i = D_o - \frac{(x_c - x_0) * D_o}{x_i - x_c} \quad \text{Ec.-2.25}$$

## 2.4 PROCESAMIENTO DE LAS IMÁGENES PARA OBTENER LA POSICION DE LA LINEA LASER

Como se mencionó anteriormente, para recuperar las alturas  $h_i$  de una superficie es necesaria una triangulación geométrica que requiere de la posición de la proyección, para la detección de esta se debe tomar en cuenta un modelo matemático que nos permita calcular tal posición con una exactitud de fracciones de pixel. De acuerdo a las propiedades de una proyección laser, el perfil de intensidades que esta posee puede ser modelado por una función gaussiana.

### 2.4.1 Aproximación gaussiana

Como se había mencionado, una de las formas de modelar el perfil de intensidades de una línea laser, es ajustando este perfil a una función gaussiana [16]. Analizando la sección transversal de la proyección, podemos obtener a partir de la imagen, la posición y la intensidad de cada uno de los pixeles, el valor de cada uno de estos pixeles es representado por pares de variables que representan tanto la intensidad como la posición del pixel y se expresa como:

$$\{(x_0, z_0), (x_1, z_1), \dots, (x_i, z_i)\}.$$

Donde  $x_i$  representa la posición y  $z_i$  la intensidad del pixel.

Se puede expresar al conjunto de pixeles como una función continua de intensidad  $f(x)$ , empleando la ecuación 2.26.

$$f(x) = \frac{Ni}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{Ec.-2.26}$$

Donde  $Ni$  es el área bajo la curva,  $\mu$  es la esperanza matemática de la función y  $\sigma$  es la desviación estándar. El parámetro que nos dará la posición central de la función gaussiana es el valor de la esperanza matemática, esta es calculada por la expresión 2.27.

$$\mu = \frac{\sum_{i=1}^n z_i x_i}{\sum_{i=1}^n z_i} \quad \text{Ec.-2.27}$$

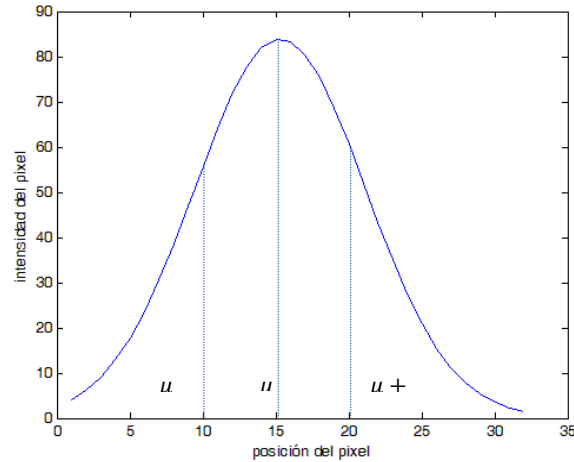
La variable  $\sigma$  es la desviación estándar de la función Gaussiana, que se calcula como:

$$\sigma = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n-1} \quad \text{Ec.-2.28}$$

Donde  $Ni$  es el área bajo la curva y se puede calcular como:

$$Ni = \sum_{i=0}^n x_i l \quad \text{Ec.-2.29}$$

En la figura 2.12 se muestra una aproximación del perfil de una línea laser digitalizada. Para determinar la posición de la línea láser usando la ec.2.27, la posición de la intensidad máxima debe ser calculada para cada fila de la imagen. Ubicado el máximo se realiza un barrido de los pixeles adyacentes del perfil, para discriminar los pixeles que pertenecen al perfil se establece un umbral de intensidad, obtenidos los pixeles que pertenecen a la imagen de la proyección, se calcula el valor de la esperanza matemática  $\mu$ .



**Figura 2.12.- Posición de la línea laser a partir del ajuste de puntos a una función gaussiana.**

$\mu$  es un cálculo del promedio ponderado, este nos permite conocer la posición exacta de la línea, no importando si la imagen de la proyección está saturada.

Con el cálculo de la posición para cada fila de la imagen, el desplazamiento relativo puede ser calculado para cada fila y las  $h_i$  pertenecientes a la sección transversal de la superficie pueden ser recuperadas.

#### 2.4.2 Curvas Bézier

El principio de esta técnica de modelado establece que cualquier punto sobre un segmento de curva es dado por una función paramétrica (Ec.-2.30) [17].

$$\mathbf{P}(u) = \sum_{i=0}^n \mathbf{P}_i \mathbf{B}_{i,n}(u) \quad u \in [0,1] \quad \text{Ec.-2.30}$$

Donde  $\mathbf{P}_i$  son los vértices de un polígono característico, estos vértices son los puntos de control y  $\mathbf{B}_{i,n}(u)$  es la función base. Por tanto se puede decir que una serie de puntos arbitrarios puede aproximarse a una curva continua por medio de este polinomio de grado  $n$ .

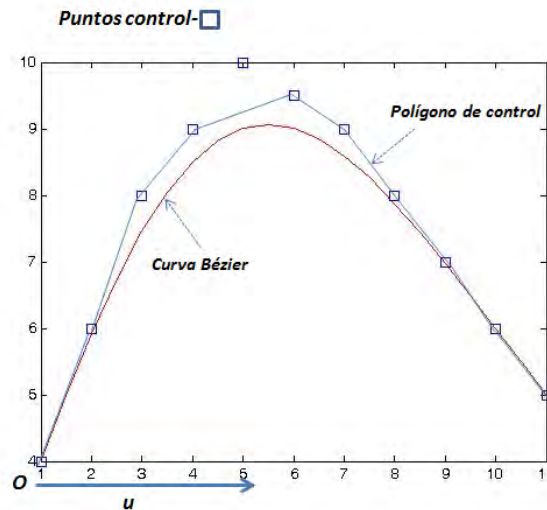
La función base queda descrita por la ecuación 2.31.

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad \text{Ec.-2.31}$$

Y donde  $\binom{n}{i}$  denota la función del coeficiente binomial, o distribución binomial de la probabilidad y estadística (Ec. 2.32).

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad \text{Ec.-2.32}$$

Dadas las características de la función base, una curva de Bézier recae en la parte interna del polígono de control (figura 2.13), la curva modelada es tangente al segmento inicial y final del polígono de control, esta permiten añadir puntos control para un mejor ajuste al modelo y donde el modelo es afectado por cada uno de los puntos control que componen el segmento de curva.



**Figura 2.13.- Modelado de un conjunto de puntos por medio de una curva Bézier.**

Esta es una técnica de ajuste donde los cambios bruscos son suavizados por la curva, tal característica permite reducir el ruido en los datos recuperados, donde este ruido es comúnmente asociado a los errores de medición.

Para lograr un mejor ajuste, es necesario implementar esta técnica bajo otro concepto, en este caso se realiza un ajuste por medio de pesos asociados a cada punto control y donde se realiza bajo la idea de una red neuronal.

## CAPITULO III.- SISTEMA DE VISION PARA RECONSTRUCCION Y MODELADO DE OBJETOS.

---

Para implementar la técnica de modelado mediante NURBS es necesaria una medición de la superficie a través de la proyección de línea laser descrita en el capítulo 2, donde un sistema mecánico posiciona el objeto dentro de la escena 3D, el arreglo óptico que permite recuperar la topografía del objeto y un programa que facilita la extracción, el procesamiento y despliegue de la información obtenida por el sistema de visión. Con este sistema se pretende recuperar los puntos de la superficie analizada, los cuales servirán como una referencia para realizar el modelado del objeto mediante la técnica NURBS.

En este capítulo se describe el desarrollo e implementación de un sistema para llevar a cabo la reconstrucción y modelado de un objeto, en la primera parte se dará una descripción del programa para el control de la cámara y se hablara de la calibración del sistema de visión.

La última sección hablara del funcionamiento del sistema mecánico y el proceso de reconstrucción, así también se describirá la interfaz gráfica para el control del sistema y los algoritmos computacionales para el proceso de reconstrucción y modelado de la superficie.

Con esto se pretende integrar un sistema que facilite la obtención y procesamiento de superficies de objetos 3D a través de un sistema de visión artificial, dando como resultado un modelo de ingeniería que puede ser aplicado a un proceso de desarrollo y mejoramiento en sistemas de manufactura asistidos por computadora.

El desarrollo de la interfaz gráfica y los algoritmos computacionales para el control de la cámara y el sistema mecánico de posicionamiento, se utilizó el ambiente de desarrollo Qt Creator 2.4.1, este es un ambiente integrado de desarrollo que facilita el desarrollo de software para aplicaciones de diversas áreas, este consiste de forma general de un editor de código fuente, herramientas para una construcción automática y un depurador de programas.

Este ambiente de desarrollo (IDE) tiene una programación basada en C++, que es un lenguaje de programación orientado a objetos, cuenta con librerías que ofrecen funciones ya desarrolladas y clases que permiten un manejo de datos de manera fácil y eficiente, además de las clases desarrolladas en Qt 4.7.4 que facilitan la construcción de interfaces gráficas para el usuario (GUI).

La comunicación con la cámara y el procesamiento de las imágenes se desarrolló con OpenCV 2.4.1, este es una biblioteca libre dedicada a aplicaciones de visión artificial, este proyecto es una forma fácil y eficiente de manipular cámaras así como el procesamiento de imágenes, al igual que Qt este proyecto se basa en el lenguaje de programación de C y C++.

### 3.1 COMPONENTES DEL SISTEMA DE VISION

El sistema de visión está compuesto por una cámara y un proyector laser, donde la principal función de la cámara es capturar información de la topografía del objeto, mientras el otro elemento tiene la función de proyectar un patrón sobre la superficie del objeto, esto para facilitar la recuperación de la topografía en el post-procesamiento.

De las ventajas que se tiene con una cámara digital podemos mencionar que la señal digital de video es exactamente igual cuando sale de la cámara que cuando llega al dispositivo de salida y que puede ser interpretada de solo una forma [4], con esto nos cercioramos de que no va a ver perdidas de información y que la información obtenida es correcta.

Para la implementación del dispositivo de visión se eligió la cámara Logitech C525 (figura 3.1), esta es una cámara digital con las siguientes características:

- Tipo de conexión: USB
- Material del lente y el sensor: Plástica
- Resolución Óptica máxima: 8MP
- FOV: 69°
- Formatos de la imagen: 320x240, 640x480, 1280x720
- Cuadros por segundo: 30fps@640x480
- Sensor: CMOS
- Tamaño del sensor: 1/6 in



*Figura.-3.1.- Cámara digital implementada en el arreglo óptico para reconstrucción 3D.*

El dispositivo implementado para la proyección del patrón estructurado es un diodo láser (figura 3.2), este emite un haz luminoso con un perfil de intensidad de tipo gaussiano, este dispositivo tiene integrada una lente cilíndrica para lograr la proyección de línea, así también posee un sistema que permite enfocar el haz manualmente, con lo cual se logra un adelgazamiento de la línea.

Las características del proyector láser son las siguientes:

- Longitud de onda: 635nm
- Potencia de salida: 10 mW
- Marca: Coherent





**Figura.-3.2.- Diodo laser implementado en el arreglo óptico para reconstrucción.**

Básicamente con estos dos dispositivos y una configuración geométrica correcta, el sistema es capaz de recuperar los puntos para la reconstrucción de la superficie.

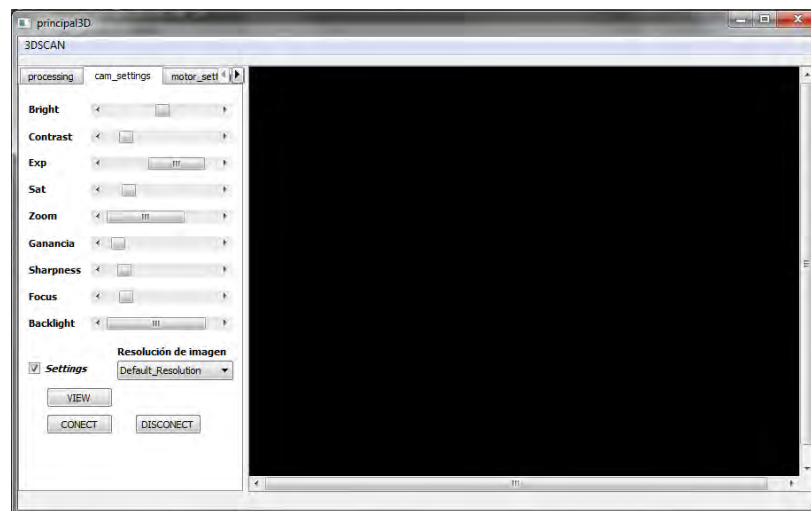
## 3.2 CONTROL DEL DISPOSITIVO DE VISION

La cámara es una parte importante del sistema de visión, esta proporciona información del mundo que se observa y la traduce a una señal digital, que puede ser leída por la computadora, una parte esencial en este elemento es el control para lograr que nos entregue información de calidad, que sea interpretable por el sistema.

El principal objetivo ahora es controlar este dispositivo, por tanto es necesario el desarrollo de una interfaz gráfica y los algoritmos computacionales para la manipulación de la cámara, lo cual permitirá obtener y mejorar la información proveniente del dispositivo, con esto se pretende obtener información precisa de la escena 3D.

### 3.2.1 Interfaz gráfica para la cámara

La interfaz para el control de la cámara (figura 3.3), integra una ventana para visualizar los cambios realizados a los parámetros de la imagen y una barra lateral que contiene los controles de los parámetros.



**Figura.-3.3.- Panel para ajuste de los parámetros de la cámara y área de desplegado.**

Ahora que se ha integrado una interfaz para manipular los parámetros de la cámara, pasamos a la explicación de los algoritmos. En primera instancia tenemos las funciones básicas para poder leer la información de la cámara Logitech, estas son la conexión y leer video.

Para la conexión de la cámara se utilizó funciones provenientes de la librería OpenCV. En la figura 3.4 tenemos el código para la conexión de la cámara, donde **capture** es una variable que tendrá la información de la cámara y nos proveerá una conexión con la misma.

En la línea 1 se muestra la función que crea la variable para la conexión con la cámara, en la línea 2 se tiene la función para adquirir imágenes de la cámara y a partir de la línea 4 se tiene el código para mostrar la imagen en la interfaz gráfica. Con este código se ha establecido la comunicación con la cámara y donde el tipo de captura se especifica cómo tipo **DirectShow** en la línea 1, esta es una arquitectura abierta que soporta varios formatos para interfaces en Microsoft Windows.

```
Capture = cvCreateCameraCapture(CV_CAP_DSHOW+0);      -1  
imview = cvQueryFrame(capture);                      -2  
imQview = IplImage2Qimage(imview);  
scene->addPixmap(QPixmap::fromImage(*imQview));      -4  
ui->graphicsView->setScene(scene);  
ui->settings->setDisabled(false);
```

**Figura.-3.4.- Algoritmo para capturar, leer y mostrar imágenes provenientes de la cámara.**

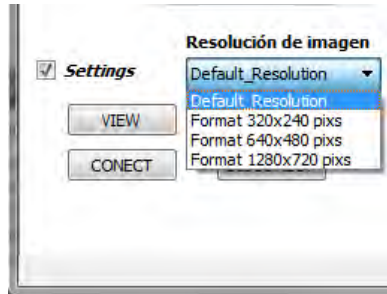
Para editar los parámetros de la cámara contamos con diversas funciones de las librerías OpenCv, en la figura 3.5 se muestra el código para editar los parámetros. En la línea 1 se almacena el valor de la variable para mantenerla como referencia, en la línea 2 se tiene la función para modificar cualquier parámetro de la cámara, donde **PROPERTY** identifica la propiedad a modificar y **capture** hace referencia al dispositivo de video. En este caso si se activa la barra es modificado el valor de la propiedad de la cámara, las propiedades pueden ser identificadas como: CV\_CAP\_PROP\_GAIN, CV\_CAP\_PROP\_SHARPNESS, CV\_CAP\_PROP\_ZOOM, CV\_CAP\_PROP\_SATURATION, CV\_CAP\_PROP\_CONTRAST, CV\_CAP\_PROP\_BRIGHTNESS, CV\_CAP\_PROP\_FOCUS, etc.

```
//----- camera properties -----  
If(Si la barra es movida a un valor diferente){  
references[0]=ui->ScrollBar->value();                -1  
cvSetCaptureProperty(capture,PROPERTY,ui->ScrollBar->value()); -2  
}
```

**Figura.-3.5.- Algoritmo para modificar parámetros de la cámara.**

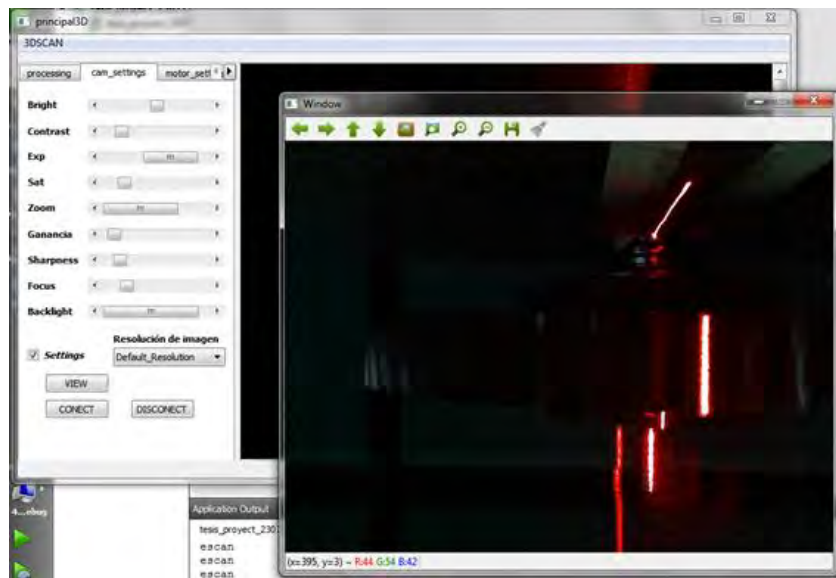
La resolución de salida puede ser seleccionada por la pestaña **Resolución de imagen**, en la figura 3.6 se muestra la pestaña desplegada, en esta se puede visualizar los formatos disponibles para la

cámara Logitech C525, donde se ha establecido por defecto el formato 640x480, el procesamiento de las imágenes se llevó a cabo en los formatos 640x480 pixeles y 1280x720 pixeles, obteniendo una mejora en la precisión del cálculo del desplazamiento con el formato 1280x720 pixeles.



**Figura.-3.6.-** Panel para selección de la resolución de la imagen.

La opción **VIEW** en la barra de control sirve para poder visualizar el video proveniente de la cámara, esta opción facilita el posicionamiento del dispositivo y a su vez sirve como retroalimentación cuando se modifican los parámetros de la imagen, en la figura 3.7 se muestra la ventana de video.



**Figura.-3.7.-** Ventana de video del software de reconstrucción 3D.

Por último se tiene el botón **DISCONNECT**, esta opción permite cerrar la conexión entre la cámara y la PC.

### 3.3 CALIBRACION DE UN SISTEMA DE VISION ARTIFICIAL

Tomando el arreglo del sistema óptico de la figura 3.8, podemos determinar los parámetros para la calibración del sistema, en este se tiene la configuración básica de un proyector láser, una cámara CCD y su respectivo plano de referencia.

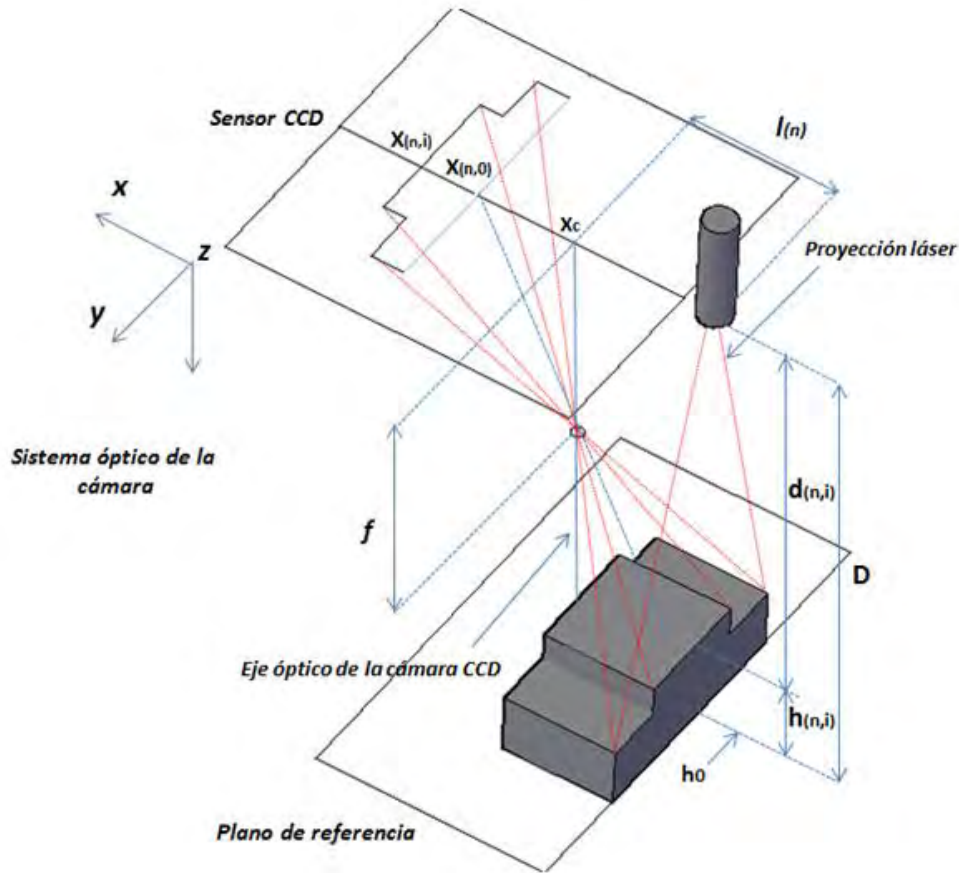


Figura.-3.8.- Esquema de la geometría del arreglo óptico implementado en la reconstrucción 3D.

Realizando una relación de triángulos semejantes, se puede relacionar  $D$ ,  $h_i$ ,  $l_i$ ,  $\eta$ ,  $f$  y  $x_i$  con la ecuación 3.1:

$$\frac{f}{\eta(x_c - x_{n,i})} = \frac{D - h_{n,i}}{l_n} \quad \text{Ec.-3.1}$$

Donde  $\eta$  representa la dimensión del pixel del sensor de la cámara, desarrollando la ecuación para  $h_{11}$  se tiene:

$$h_{11} = D - \frac{f * l_1}{\eta(x_c - x_{11})} \quad \text{Ec.-3.2}$$

De la ecuación 3.2 se puede ver que la altura  $h$  puede ser calculada si se conoce  $D$ ,  $l$  y  $f$ , las posiciones  $x_i$  de la línea laser pueden ser calculadas por aproximación de gauss y la dimensión  $\eta$  es conocida de datos del fabricante, por tanto es necesario conocer la posición central ( $X_c$ ) de la imagen,  $D$  y  $f$  para recuperar la altura de la superficie. Para conocer los parámetros faltantes es necesario calcularlos de manera indirecta, esto por medio de una serie de imágenes de calibración.

Este cálculo consiste en tomar un conjunto de imágenes, donde se tienen posiciones  $x_{i,n}$  para una altura  $h_{i,n}$  conocida, la información podrá ser usada para calcular los parámetros desconocidos por medio de la ecuación 3.1, entonces desarrollando la expresión para  $h_{11}$  obtenemos:

$$f * l_1 = \eta(D - h_{11})(x_c - x_{11}) \quad \text{Ec.-3.3}$$

Haciendo lo mismo para la altura  $h_{12}$ :

$$f * l_1 = \eta(D - h_{12})(x_c - x_{12}) \quad \text{Ec.-3.4}$$

Igualando ambas expresiones (3.3-3.4) y desarrollando para  $D$  se llega a la ecuación 3.5:

$$D = \frac{h_{11}*(x_c-x_{11})-h_{12}*(x_c-x_{12})}{x_{12}-x_{11}} \quad \text{Ec.-3.5}$$

Ahora haciendo el mismo desarrollo para  $h_{11}$  y  $h_{13}$ , se llega a la ecuación 3.6:

$$D = \frac{h_{11}*(x_c-x_{11})-h_{13}*(x_c-x_{13})}{x_{13}-x_{11}} \quad \text{Ec.-3.6}$$

Igualando ambas ecuaciones (3.5-3.6) y desarrollando para  $X_c$  se llega a la ecuación 3.7.

$$x_{c1} = \frac{h_{11}x_{11}(x_{13}-x_{11})-h_{12}x_{12}(x_{13}-x_{11})-h_{11}x_{11}(x_{12}-x_{11})+h_{13}x_{13}(x_{12}-x_{11})}{h_{11}(x_{13}-x_{11})-h_{12}(x_{13}-x_{11})-h_{11}(x_{12}-x_{11})+h_{13}(x_{12}-x_{11})} \quad \text{Ec.-3.7}$$

Ahora se conoce la posición central ( $X_c$ ) de la imagen, los parámetros faltantes serian la distancia focal  $f$  y  $l$ , para calcular  $f$  se usa la ecuación 3.1, donde ahora se ha denotar que se modifica la distancia ( $l_a$ ) entre la cámara y el proyector para poder calcular este parámetro, al desarrollarla se llega a la siguiente expresión para  $h_{11}$  y  $l'_1$ :

$$f * l'_1 = \eta(D - h_{11})(x_{c1} - x_{11}) \quad \text{Ec.-3.8}$$

Despejando  $X_c$  de las ecuaciones 3.3 y 3.8 e igualando ambas expresiones se obtiene:

$$f * (l'_1 - l_1) = \eta(D - h_{11})(x_{11} - x'_{11}) \quad \text{Ec.-3.9}$$

Desarrollando para  $f$  en la ecuación 3.9 se llega a la ecuación 3.10:

$$f = \eta \frac{(D-h_{11})(x_{11}-x'_{11})}{\Delta l} \quad \text{Ec.-3.10}$$

Donde  $\Delta l$  corresponde a la diferencia ( $l'_1 - l_1$ ), estos parámetros son conocidos y por tanto  $f$  puede ser calculada. Análogamente puede ser calculada la distancia  $l_n$  entre la cámara CCD y el proyector con la ecuación 3.11.

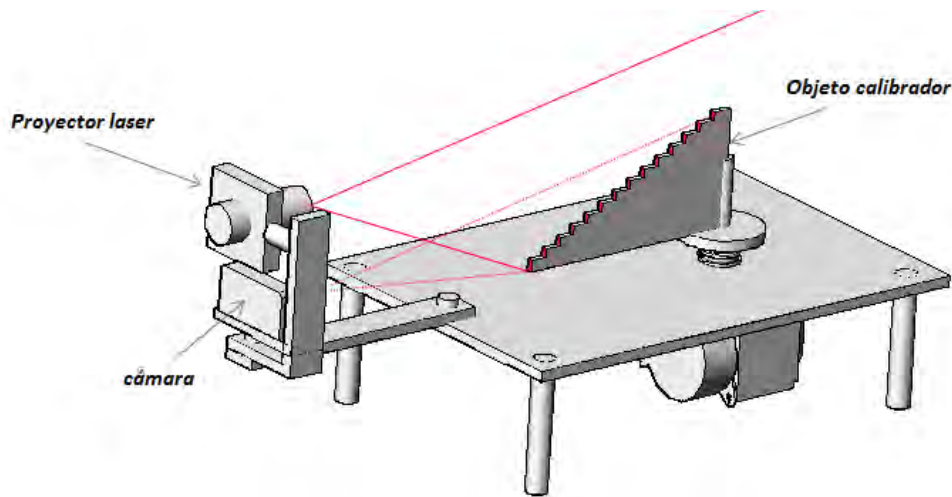
$$l_1 = \eta \frac{(D-h_{11})(x_{c1}-x_{11})}{f} \quad \text{Ec.-3.11}$$

Con esto se tiene la calibración del sistema que nos permitirá recobrar las alturas  $h_i$  de la superficie en análisis.

### 3.4 CALIBRACION IMPLEMENTADA

Ya que no es viable establecer el método de calibración expuesto en la sección 3.3, debido a la dificultad para conocer la posición exacta del plano de referencia y de las demás variables relativas al arreglo, se optó por integrar una calibración basada en redes neuronales con funciones Bézier y un objeto de referencia, este método permite calibrar el sistema sin la medición de las variables externas del arreglo y por tanto se reduce el error introducido en la medición de las alturas  $h_i$ .

En el proceso de calibración del sistema implementado se utiliza el arreglo mostrado en la figura 3.9, en este se tiene el dispositivo de visión, un proyector laser y un objeto de dimensiones conocidas que servirá como referencia para la calibración.



**Figura.-3.9.- Arreglo óptico para calibrar el sistema de visión artificial.**

Para llevar a cabo la calibración es necesario el proyector, como parte fundamental del arreglo óptico, el proyector debe facilitar la ubicación y recuperación de la topografía del objeto en escena, logrando esto por medio de la proyección del patrón estructurado, en la figura 3.10 se tiene la imagen del patrón utilizado para la calibración del sistema.

La calibración tiene por objetivo, obtener una relación de los desplazamientos sufridos por la línea proyectada y las alturas del objeto referencia, para lograrlo se proyecta la línea sobre el perfil del objeto referencia, obteniendo así los desplazamientos referentes a las alturas  $h_i$ .



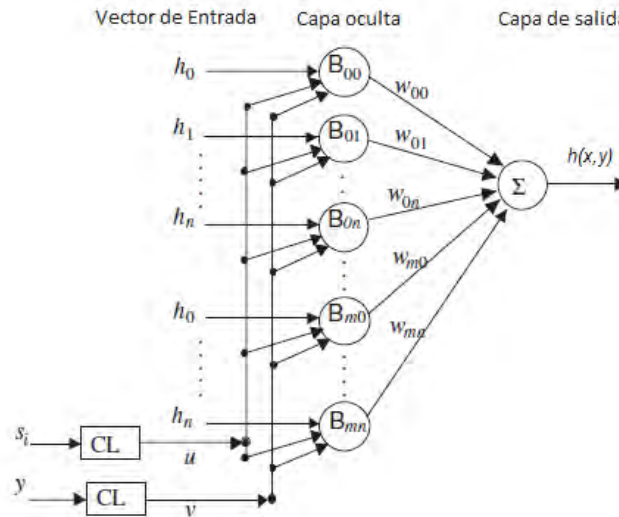
**Figura.-3.10.- Imagen del objeto utilizado para encontrar la curva de calibración**

Con la captura de estos desplazamientos, se realiza un procesamiento para obtener las relaciones discretas de alturas y desplazamientos, estas relaciones son puntos que forman parte de la curva de calibración del sistema, donde para obtener la función que nos represente esta curva, se realiza un ajuste por medio de una red neuronal basada en funciones Bézier.

Esta red neuronal proveerá la curva de calibración, solo que esta vez la curva quedara representada por una función continua, esta curva será la referencia para la reconstrucción de la topografía de los objetos en análisis.

#### 3.4.1 Estructura de la red neuronal

La estructura propuesta para la red neuronal consiste de un vector de entrada  $S$ , dos entradas paramétricas  $(u, v)$ , una capa oculta y una capa de salida, la estructura de la red neuronal se muestra en la figura 3.11.



**Figura.-3.11.- Estructura de la red neuronal de Bézier para encontrar la curva de calibración, figura obtenida de referencia [3].**

Los datos de entrada incluyen las profundidades  $h_i$ , el desplazamiento  $S_{ij}$  de la línea láser y la posición  $y_i$  de cada fila de la imagen, para obtener las profundidades  $h_i$  es necesario un desplazamiento del plano de referencia desde la posición inicial  $h_0$  a las posiciones  $h_1, \dots, h_n$ , en este proceso el sistema toma de referencia un objeto de calibración que permite obtener las profundidades  $h_i$  y sus respectivos desplazamientos  $S_{ij}$ .

El desplazamiento  $S_{ij}$  de la línea es representado en términos de la variable paramétrica  $u$ , por medio de la siguiente combinación lineal (CL):

$$u = a_0 + a_1 S \tag{Ec.-3.12}$$

Donde  $a_0$  y  $a_1$  son constantes, para determinar estas constantes es necesario evaluar la ecuación 3.12 en dos valores conocidos de  $S_{ij}$  e  $u$ , se conoce que la expresión es evaluada en un rango  $0 \leq u \leq 1$  por tanto se tiene que para  $u = 0, S_{ij} = S_0$  y cuando  $u = 1, S_{ij} = S_n$ .

Sustituyendo estos valores en la expresión se obtiene un sistema de dos ecuaciones con dos incógnitas, el cual se puede resolver para conocer los valores de las constantes  $a_0$  y  $a_1$ .

La posición de la línea laser  $y_i$  es representada por la variable paramétrica  $v$ , cuya relación es expresada como:

$$v = b_0 + b_1 y \tag{Ec.-3.13}$$

Donde  $b_0$  y  $b_1$  son constantes, para determinar estas constantes es necesario evaluar la expresión 3.13 en dos valores conocidos de  $y$  e  $v$ , se conoce que la expresión es evaluada en un rango  $0 \leq v \leq 1$  por tanto se tiene que para  $v = 0, y_i = y_0$  y cuando  $v = 1, y_i = y_n$ .

Sustituyendo estos valores en la expresión se obtiene un sistema de dos ecuaciones con dos incógnitas el cual se puede resolver para conocer los valores de las constantes  $b_0$  y  $b_1$ .

La capa oculta es construida por funciones base de Bézier, cuya forma es descrita como:

$$B_{ij} = B_j(v)B_i(u) \tag{Ec.-3.14}$$

Dónde:

$$B_i(u) = \binom{n}{i} u^i (1-u)^{n-i}, B_j(v) = \binom{m}{j} v^j (1-v)^{m-j}$$

La salida es obtenida por medio de la suma de las neuronas en la capa oculta, la respuesta de la red neuronal para la altura  $h(u, v)$  es expresada como:

$$H(u, v) = \sum_{i=0}^n \sum_{j=0}^m w_{ij} h_i B_i(u) B_j(v) \quad (u, v) \in [0, 1] \tag{Ec.-3.15}$$



Donde  $w_{ij}$  son los pesos,  $h_i$  es la dimensión conocida del objeto y  $B_i(\mathbf{u})$  e  $B_j(\mathbf{v})$  son las funciones base de Bézier, para construir completamente la red neuronal es necesario conocer los pesos  $w_{ij}$  más convenientes que produzcan las salidas correctas  $h_i$ .

Para encontrar los pesos desarrollamos la ecuación 3.15 para cada una de las alturas  $h_i$ , donde queda un sistema de ecuaciones lineales como:

$$\begin{aligned}
 H_{00} = h_0 &= w_{00}h_0B_{0,0} + \dots + w_{0m}h_0B_{0,m} + \dots + w_{10}h_1B_{1,0} + \dots + w_{1m}h_1B_{1,m} + \dots + w_{nm}h_nB_{m,n} \\
 &\vdots \\
 H_{01} = h_n &= w_{00}h_0B_{0,0} + \dots + w_{0n}h_nB_{0,n} + \dots + w_{10}h_1B_{1,0} + \dots + w_{m0}h_0B_{1,m} + \dots + w_{mn}h_nB_{m,n} \\
 &\vdots \\
 H_{10} = h_0 &= w_{00}h_0B_{0,0} + \dots + w_{0m}h_0B_{0,m} + \dots + w_{10}h_1B_{1,0} + \dots + w_{1m}h_1B_{1,m} + \dots + w_{nm}h_nB_{m,n} \\
 &\vdots \\
 H_{11} = h_n &= w_{00}h_0B_{0,0} + \dots + w_{0m}h_0B_{0,m} + \dots + w_{10}h_1B_{1,0} + \dots + w_{1m}h_1B_{1,m} + \dots + w_{nm}h_nB_{m,n}
 \end{aligned}$$

Simplificando el sistema, lo podemos expresar como:

$$\begin{aligned}
 H_{00} &= w_{00}\beta_{0,0} + w_{01}\beta_{0,1} + \dots + w_{mn}\beta_{0,mn} \\
 H_{01} &= w_{00}\beta_{1,0} + w_{01}\beta_{1,1} + \dots + w_{mn}\beta_{1,mn} \\
 &\vdots \\
 H_{mn} &= w_{00}\beta_{mn,0} + w_{01}\beta_{mn,1} + \dots + w_{mn}\beta_{mn,mn}
 \end{aligned}$$

Este sistema podemos representarlo de forma matricial como:

$$\begin{bmatrix} \beta_{0,0} & \dots & \beta_{0,mn} \\ \vdots & \dots & \vdots \\ \beta_{mn,0} & \dots & \beta_{mn,mn} \end{bmatrix} \begin{bmatrix} w_{00} \\ \vdots \\ w_{mn} \end{bmatrix} = \begin{bmatrix} H_{00} \\ \vdots \\ H_{mn} \end{bmatrix}$$

Resolviendo este sistema lineal para la matriz de pesos  $w_{ij}$  por el método de Chelosky, se puede encontrar la curva de calibración.

### 3.4.2 Implementación de la red neuronal Bézier

En la figura 3.12 se tiene el algoritmo para encontrar la curva de calibración por medio de la red propuesta, en la línea 1 tenemos el vector  $S_i$  que contienen los desplazamientos y las alturas de la calibración, después de la línea 5 se tiene un ciclo para calcular el valor de la variable paramétrica  $u$  referente a los desplazamientos  $S_i$ , en la línea 10 se tiene el cálculo de los desplazamientos  $S(u)$  por medio de Bézier, por último se tiene el vector  $U(j)$  que contendrá los valores de las variables paramétricas.

```

Int n = Sj->cols-1;                                -[1]
float Su=0,Hu=0;
Mat U = Mat::zeros(1,Sj->cols,CV_32FC1);
//se trabaja con el sistema de ecuaciones de desplazamientos
for(int j=0;j<Sj->cols;j++){                        -[5]
    for(float u=0;u<=1.001;u+=0.001){
        Su = 0;
        for(int i=0;i<Sj->cols;i++){
            //CBIN(), es una función que calcula el coeficiente binomial
            Su += CBIN(n,i)*pow(u,i)*pow((1-u),n-i)*Sj(0,i);    -[10]
        }
        if(Su<=Sj->at<float>(0,j)){
            U(0,j) = u;                                        -[13]
            break;
        }
    }
}

```

**Figura.-3.12.- Algoritmo para calcular la curva de calibración.**

Para calcular el valor de la matriz de pesos por medio de la descomposición LU se tiene el código mostrado en la figura 3.13, en la línea 4 se tiene el ciclo para construir la matriz Beta, en la línea 11 se tiene la función para resolver el sistema por descomposición LU y donde w será una matriz que contendrá los pesos para ajustar la red neuronal.

```

//se trabaja con el sistema de ecuaciones de alturas
float u = 0;
Mat Beta = Mat::zeros(Sj->cols,Sj->cols,CV_32FC1);
Mat w;
for(int j=0;j<Sj->cols;j++){                        -[4]
    u = U(0,j);
    for(int i=0;i<Sj->cols;i++){
        Beta(j,i) = CBIN(n,i)*pow(u,i)*pow((1-u),n-i)*he(l,0);
    }
}

solve(Beta,he,w,DECOMP_LU); //DECOMP_LU          -[11]

```

**Figura.-3.13.- Algoritmo para resolver el sistema lineal por descomposición LU.**

### 3.4.3 Conversión de los datos a unidades de longitud

Conocida la relación entre desplazamientos y las alturas (curva de calibración), se puede conocer las dimensiones del objeto escaneado por medio de esta relación (figura 3.14), de acuerdo al arreglo mostrado en la figura 3.15 se puede ver que a partir de este sistema coordenado podemos recuperar solo dos dimensiones referentes al objeto, una es  $h_i$  que está relacionada a la altura del diente y el espesor  $t_i$  relacionado al ancho del diente.

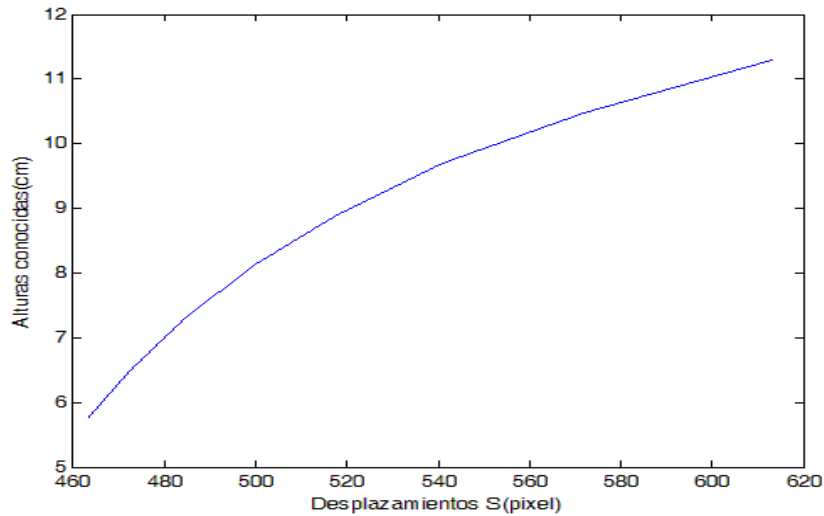


Figura.3.14- Curva de calibración para encontrar la relación desplazamiento (S) y Altura (h).

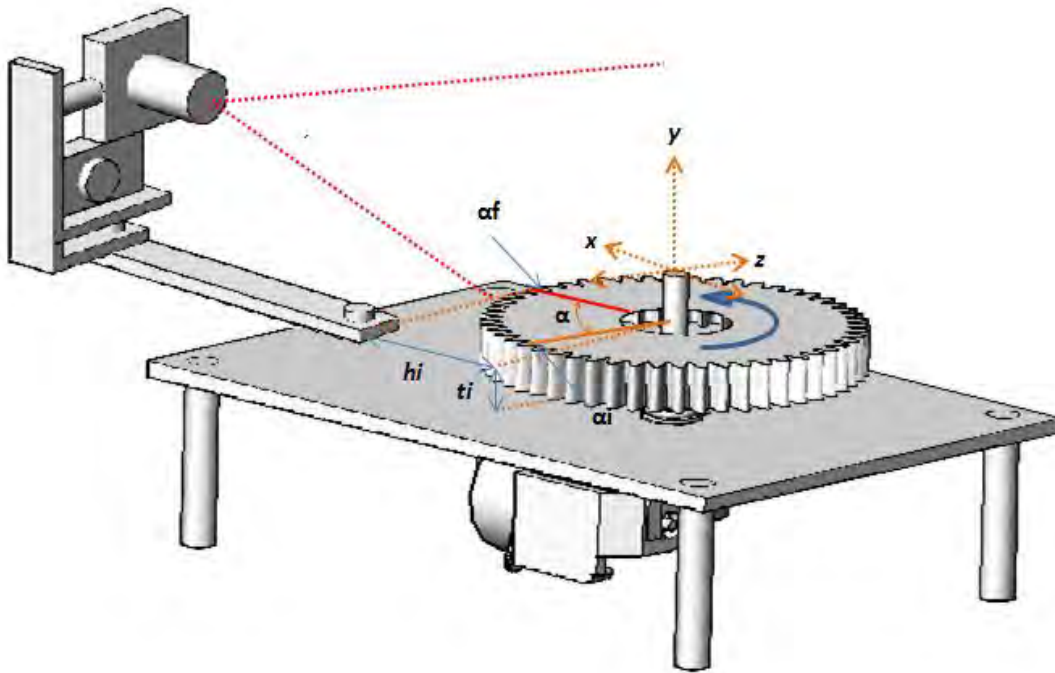
Al mover la pieza para poder escanear la superficie, se genera un ángulo  $\alpha$  relativo a una posición inicial  $\alpha_i$ , de tal modo que este ángulo crecerá a medida que el objeto es rotado y será este ángulo el que nos ayude a calcular las coordenadas de los puntos recuperados.

Cuando el sistema rota un ángulo relativo  $\alpha$  y que se encuentra en una posición  $\alpha_f$ , de acuerdo al esquema de la figura 3.15 se pueden deducir las ecuaciones 3.16 y 3.17 para calcular las coordenadas  $x$  e  $y$ .

$$z = h_i * \cos(\alpha) \tag{Ec.-3.16}$$

$$x = h_i * \sen(\alpha) \tag{Ec.-3.17}$$

Donde  $h_i$  corresponde a la altura calculada por medio de la curva de calibración,  $\alpha$  corresponde a la posición del sistema y se puede conocer por medio del sistema de posicionamiento.



*Figura.3.15- Esquema del sistema coordinado para recuperar las dimensiones del objeto.*

Para recuperar la coordenada  $y$  es necesario establecer el tamaño de cada pixel de acuerdo a un objeto referencia, en la imagen del objeto de calibración (figura 9) se tiene un perfil escalonado del cual se conocen perfectamente las dimensiones de cada escalón, la relación buscada para el tamaño de pixel se logra procesando la altura de cada uno de estos peldaños del perfil, los cuales se pueden visualizar como barras blancas.

Una vez conocidas las alturas en pixeles de cada uno de estos peldaños se procede con la relación de la dimensión del escalón y su altura en pixeles, esta relación proporciona el tamaño del pixel para cada uno de los desplazamientos  $S_{ij}$ .

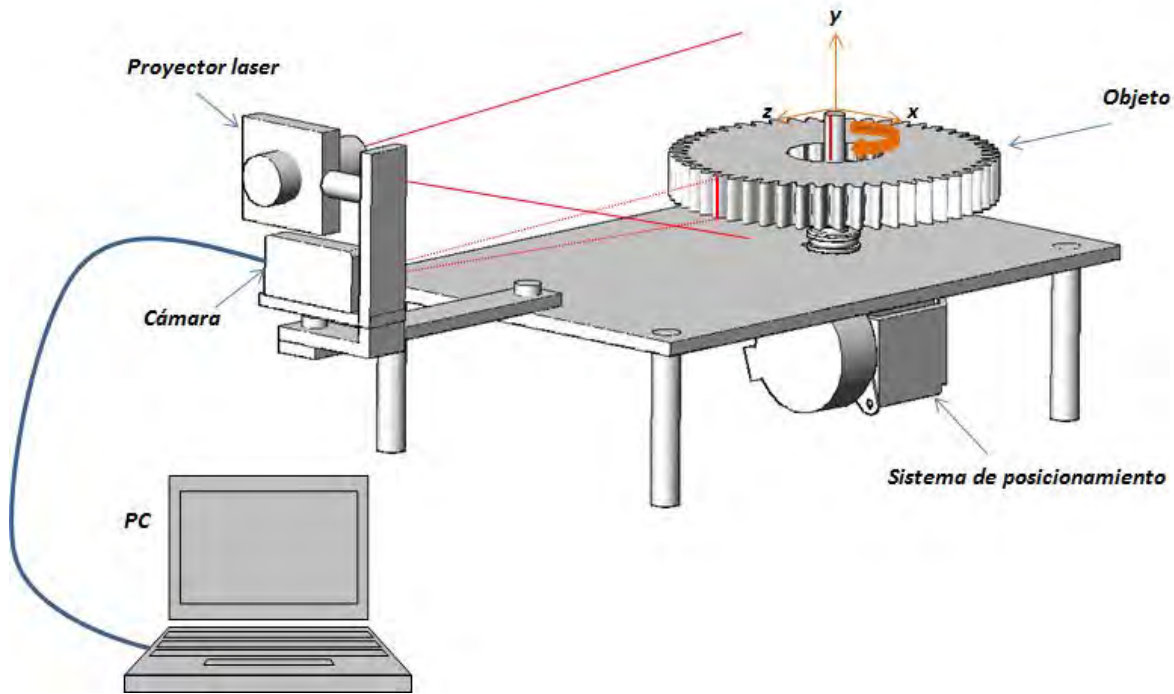
Con esto se tiene una curva que relaciona los desplazamientos  $S_{ij}$  y las dimensiones de pixel  $d_i$ , esta es igualmente modelada con funciones matemáticas para lograr una curva continua y así obtener las dimensiones de un pixel correspondiente a cada desplazamiento  $S_{ij}$ .

La curva anterior nos proporciona las dimensiones de la coordenada  $y$ , con esto se puede determinar las coordenadas  $(x, y, z)$  de cada punto perteneciente a las líneas capturadas por el sistema de visión artificial.

## 3.5 DESCRIPCION DEL SISTEMA OPTOMECANICO PARA ESCANEO DE PIEZAS

### 3.5.1 sistema mecánico y el proceso de reconstrucción

Dadas las características de la técnica aplicada para la recuperación de la topografía del objeto, se optó por la implementación de un sistema de rotación, donde el objeto en análisis estuviera sujeto a un eje de rotación y el sistema de visión permaneciera fijo, permitiendo así el escaneo de la superficie del objeto, en la figura 3.16 se muestra un esquema del sistema implementado.



**Figura.-3.16.- Sistema Óptico - mecánico para reconstrucción digital de objetos mecánicos**

El sistema de posicionamiento consta de un motor a pasos, un sistema mecánico de engrane y tornillo sinfín, una plataforma de montaje, el control electrónico y la interfaz gráfica.

El motor a pasos utilizado fue el M55SP-1 (figura 3.17), que posee las siguientes características:

- Voltaje de entrada: 21.6-26.4V
- Angulo de paso: 7.5°
- Torque Máximo: 43.6mN\*m
- Cantidad máxima de pulsos por segundo: 315pps



**Figura.-3.17.- Motor usado en el sistema de posicionamiento.**

Para la implementación del control se utilizó el PIC 18F4550 (figura 3.18), este es un dispositivo electrónico que ofrece una gama de funciones para la interfaz PC-Motor, para este proyecto se implementó una conexión USB entre el PIC y la PC, permitiendo el intercambio de información entre el programa para control del motor y el microcontrolador, el desarrollo del programa para el PIC se llevó a cabo en C y se utilizó la tarjeta MasterProg para la programación de este.



**Figura.3.18.- Microcontrolador 18F4550 de la compañía microchip.**

Las características del microcontrolador 18F4550 son:

- Voltaje de entrada: 5V
- Posee características para establecer una comunicación USB
- Corriente suplida en los periféricos: 25mA
- Numero de pines: 44
- Tamaño en memoria flash: 32K bytes
- Tamaño en memoria SRAM: 2048 bytes

El desarrollo de la interfaz gráfica de usuario y los algoritmos se llevó a cabo con librerías y clases de Qt, en la implementación de la comunicación USB se utilizó la librería Libusb\_win32, esta permite amplias aplicaciones para acceder de forma genérica a dispositivos USB en el sistema Operativo Windows.

### *3.5.2 Control del sistema de posicionamiento*

El objetivo de este control es facilitar el posicionamiento de la pieza y lograr una sincronía con el sistema óptico que permita un barrido de la superficie con la técnica de proyección de línea, en la

figura 3.19 se tiene el panel donde se integran los controles para el motor y la ejecución del programa.



Figura-3.19.- Panel de control del sistema de posicionamiento

Como se había comentado al inicio de la sección, la comunicación se realizó por medio de una conexión USB (Universal Serial Bus), en la figura 3.20 se muestra el algoritmo para establecer la conexión con el microcontrolador, donde las funciones son de la librería lib\_usb32, de la línea 1 a la 3 se tienen las funciones para iniciar la búsqueda del dispositivo USB, en la línea 4 se ejecuta la función para encontrar y abrir el dispositivo buscado.

En la línea 9 se tiene la función para establecer la configuración deseada donde **device** es el dispositivo abierto y **MY\_CONFIG** representa la configuración deseada. En la línea 10 se hace la petición de la interface con el sistema operativo y si se logra, el programa despliega en el panel (línea 16) la comunicación ha sido establecida.

```
usb_init(); /* inicializa la libreria */ -1  
usb_find_busses(); /* encuentra las conexiones usb existentes */  
usb_find_devices(); /* encuentra los dispositivos conectados */  
device = open_dev(); -4  
si el dispositivo no es abierto  
imprime "Error al abrir el dispositivo usb" y la función regresa un False  
si el dispositivo fue abierto correctamente  
imprime "el PID y VID del dispositivo USB"  
  
usb_set_configuration(devusb, MY_CONFIG) -9  
  
usb_claim_interface(devusb, 0) -10  
si la petición de la interfase no tuvo éxito  
usb_close(devusb); // se cierra el dispositivo  
si la petición de la interface tuvo éxito  
imprime "petición de interfase correcta"  
  
ui->line->setText("Comunicación establecida con el dispositivo usb"); -16  
La función regresa un TRUE
```

**Figura-3.20.- Algoritmo para establecer la interfaz del microcontrolador y la PC.**

### 3.6 INTERFAZ ELECTRÓNICA PARA EL CONTROL DEL MOTOR

La interfaz para el control del motor se implementó con el microcontrolador 18F4550, ya que este permite establecer una comunicación con el protocolo USB, el objetivo de esta interfaz es controlar el sistema mecánico para posicionar la pieza en la escena 3D, donde el sistema óptico extrae la información para la reconstrucción de las piezas.

En la figura 3.21 tenemos el diagrama del circuito electrónico para el desarrollo de la interfaz y control del motor.

La etapa de control contiene la implementación del circuito electrónico para establecer una conexión USB con el PC, esta conexión permite el envío y recepción de datos a través de los pines Tx y Rx, en esta etapa se tiene la recepción de los datos provenientes del programa para control del motor, así también el microcontrolador transmite la información necesaria para informar al programa del proceso en ejecución, en la figura 3.22 se muestra la imagen del control y donde se especifican todos sus elementos.



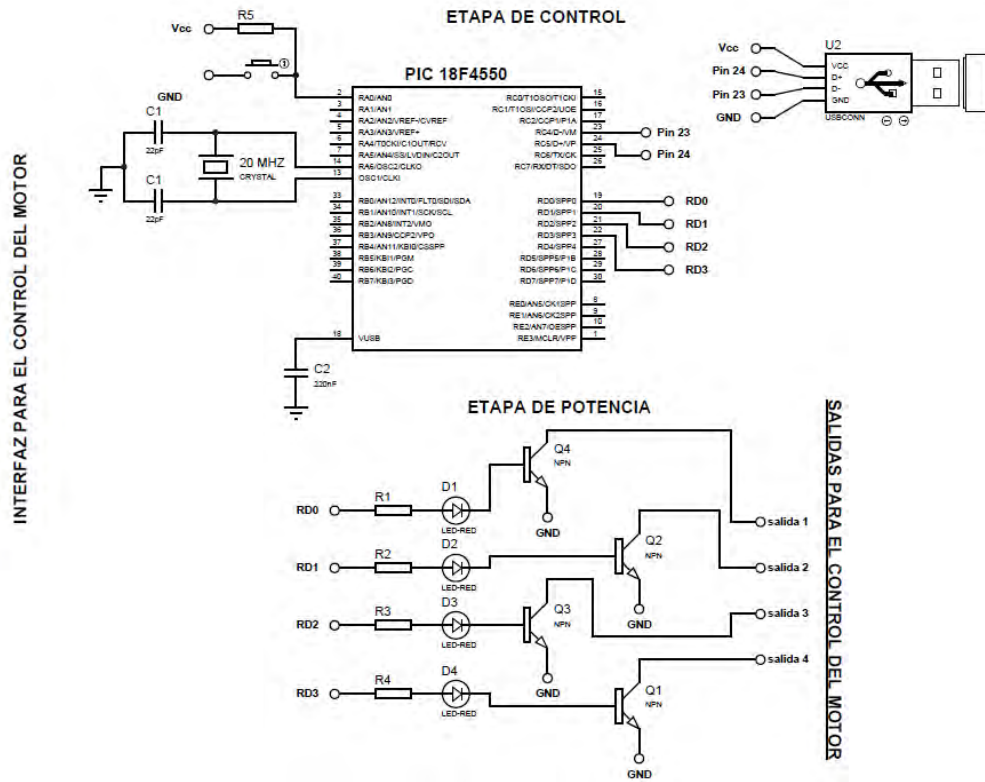


Figura-3.21.- Diagrama electrónico del circuito utilizado para el control del motor.

El objetivo primordial en la etapa de potencia es recibir las señales eléctricas del microcontrolador y amplificarlas por medio de transistores NPN TIP31C, con esto se provee una señal lo suficientemente potente para controlar el motor del sistema de posicionamiento.

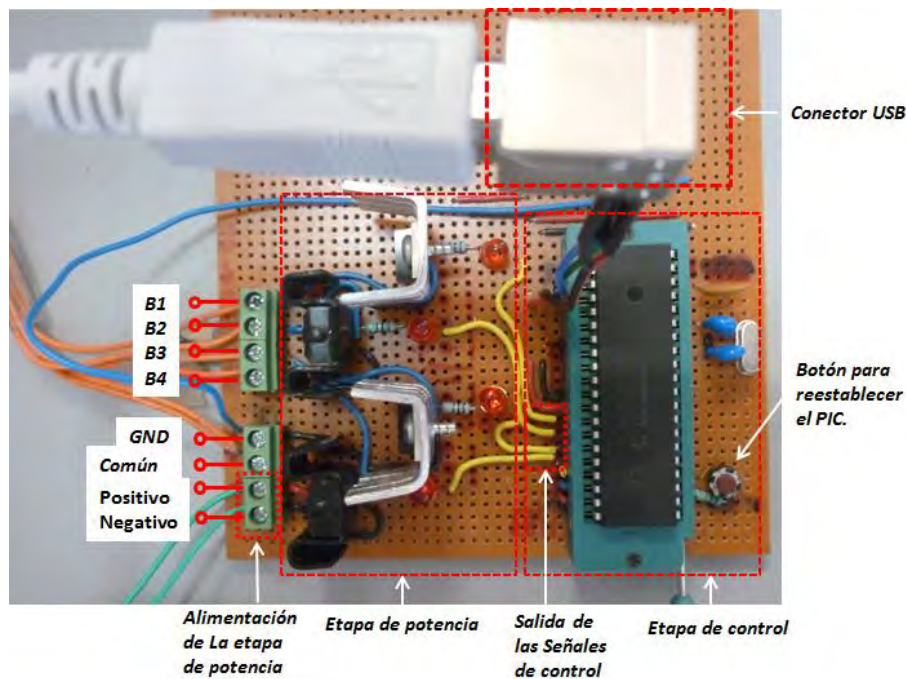
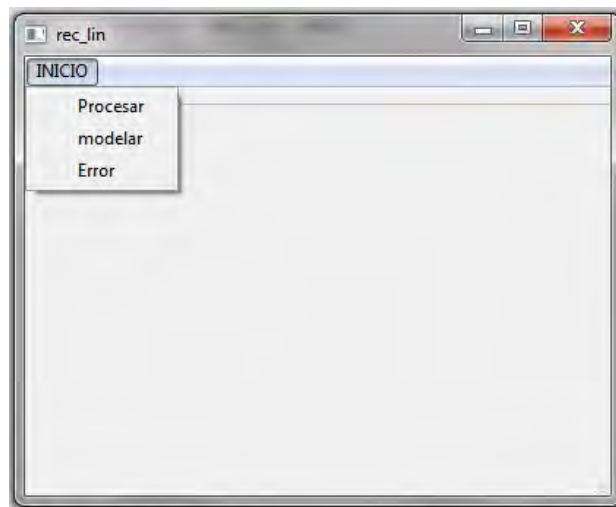


Figura-3.22.- Imagen del circuito electrónico utilizado para el control del motor.

### 3.7 DESCRIPCIÓN DE LA TECNICA PARA RECONSTRUCCIÓN DE OBJETOS

Para obtener los puntos que describirán la superficie a modelar se implementó el arreglo óptico descrito en la sección 2.3, este arreglo proporciona los puntos que describen la topografía del objeto, permitiendo que estos sean almacenados en arreglos de memoria para su post procesamiento.

En la figura 3.23 se muestra el panel para procesamiento de la información, el cual tiene tres opciones: Procesar que integra los procesos para el cálculo de la posición de la línea y la recuperación de las posiciones  $(x, y, z)$  de los puntos recuperados. Modelar que ejecuta el programa para generar la superficie modelada y la despliega en una ventana. Error que ejecuta los programas para el cálculo del error RMS y RMS%.



**Figura-3.23.- Interfaz gráfica para procesamiento de información y despliegado de la superficie.**

#### 3.7.1 Detección y procesamiento de la línea láser

La sección transversal de una proyección laser es una distribución gaussiana, tal sección es descrita por un conjunto de puntos representados como  $(x_0, z_0), (x_1, z_1), \dots, (x_n, z_n)$ , Donde  $x_i$  es la posición del pixel y  $z_i$  es la intensidad del pixel. La forma más conveniente de representar la información contenida en la sección transversal es por medio de una función gaussiana la cual es descrita por la ecuación 2.10 y se expresa como:

$$f(x) = \frac{Ni}{\sigma\sqrt{2\pi}} * \exp\left\{-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right\}$$

Para determinar la posición de la línea láser, el parámetro más importante de la función gaussiana es el valor de la esperanza matemática, debido a que este valor representa el punto central de tal función, la esperanza matemática es calculada por la ecuación 2.11 y se expresa como:

$$\mu = \frac{\sum_{i=1}^n z_i x_i}{\sum_{i=1}^n z_i}$$

En la figura 3.24 se muestra el algoritmo para el procesamiento y cálculo de la posición de la línea laser, en la primer línea se lee la información a procesar, a partir de la línea 4 se crea un ciclo **for** para escanear las filas de la imagen que contiene la proyección, en este caso cada fila contiene una sección transversal del perfil de la proyección, la función descrita en la línea 4 calcula la posición del pixel con el valor máximo, de la línea 17 a la 27 se calculan los limites derecho e izquierda de la proyección. A partir de la línea 30 se calcula la suma pesada para calcular el centro de la proyección, en la línea 36 se asigna el valor a la matriz que contendrá las posiciones de la proyección.

```

//leemos la imagen
improc = imread(FileName,0); -1
medianBlur(improc,improc,3); //Filtro para atenuar el ruido
Mat linescan = Mat::zeros(1,improc.cols,CV_32FC1);
for( int j = 180;j < 310; j++){ -4
    cont = 0; float val = 0; lineps = 0;
    line = Mat::zeros(20,33,CV_32FC1);
    linescan = improc.rowRange(j,j+1);
//Esta funcion encuentra la posición xi dela intensidad máxima
    maximum(&linescan,0,&lineps,&val,minn,450); -10
// si el pixel tiene un valor mayor que 90 y su posición es mayor que x0, el pixel pertenece a la proyección
    if(lineps >= minn&&val>90){
        float pos1=0,pos2=0,limite = 90;
//verificamos el ancho del perfil de la línea laser para calcular el centro geométrico
        for(int a=0;a<10;a++){ -17
            if(linescan.at<tip3>(0,lineps+a)<limite){
                pos2 = lineps+a;
                break;
            }
        }
        for(int a=0;a<10;a++){
            if(linescan.at<tip3>(0,lineps-a)<limite){
                pos1 = lineps-a;
                break;
            }
        }
//realizamos la suma pesada de los valores de los pixeles con la ecuación x
        float div = 0, divn = 0; -30
        for( int x = pos1;x <= pos2; x++){
            divn += x*float(linescan.at<tip3>(0,x));
            div += float(linescan.at<tip3>(0,x)); }
        if(div==0) line_pos.at<Vec3f>(j,n)[2]=0;
        else line_pos.at<Vec3f>(j,n)[2]= divn/div; -36
    }

```

Figura-3.24.- Algoritmo para la detección de la línea laser y cálculo de su posición.

### 3.8 MODELADO GEOMETRICO

En el modelado geométrico se involucran dos fases para lograr la representación de un objeto, en la primera fase se mide y extraen los puntos o datos pertenecientes a la superficie del objeto y en la segunda fase se realiza una interpolación de los puntos extraídos por medio de funciones NURBS, posteriormente se lleva un ajuste de los pesos para ajustar la superficie modelada y los datos extraídos en la primera fase.

#### 3.8.1 Parametrización de los puntos

Como se había mencionado en un principio para realizar el modelado es necesaria una medición de la superficie, por tanto el siguiente paso es la parametrización de los puntos medidos, esto se refiere a la asignación de puntos  $P(u,v)$  cuyas coordenadas paramétricas son conocidas, estos puntos serán los que representen a la superficie medida, en la figura 3.25 se muestra una superficie parametrizada.

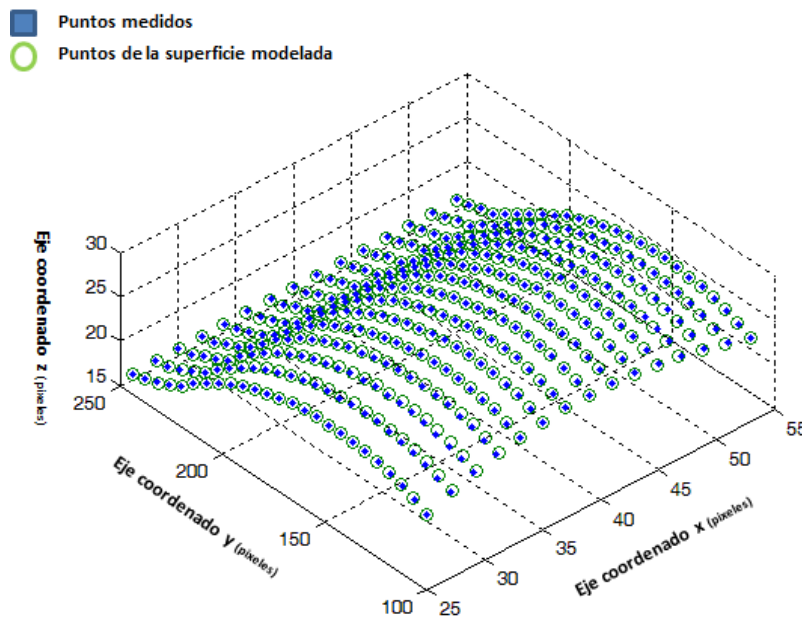


Figura-3.25.- Imagen de una superficie interpolada y parametrizada.

Para lograr una parametrización de los puntos de la superficie modelada es necesario desarrollar el sistema de ecuaciones lineales que resultan del desarrollo de la ecuación de una superficie NURBS, esta ecuación queda expresada como:

$$Q_{k,l} = S(\bar{u}_k, \bar{v}_l) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(\bar{u}_k) N_{j,q}(\bar{v}_l) P_{ij} \quad Ec. -3.18$$

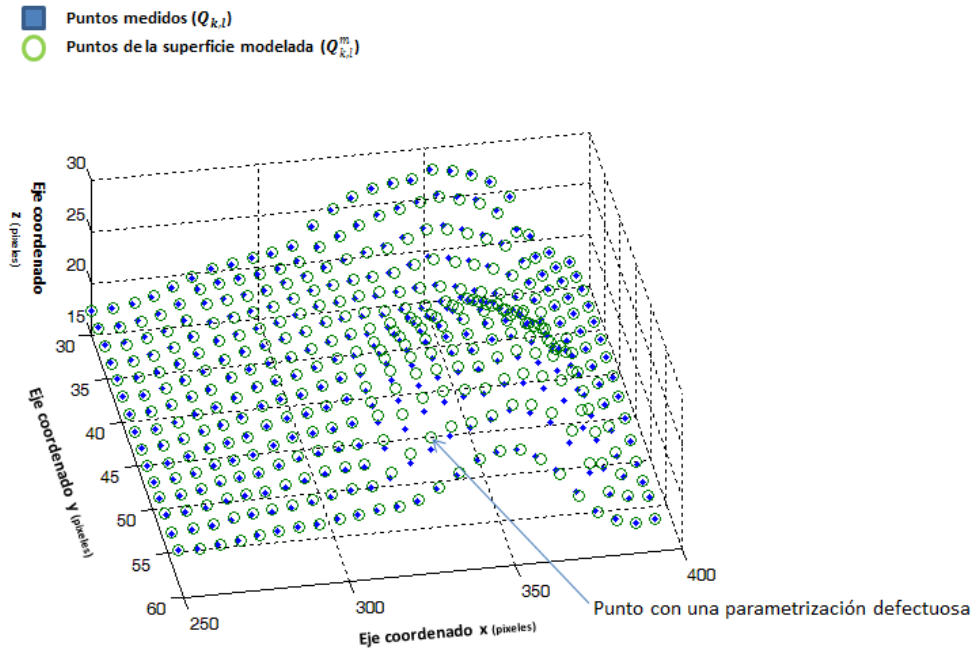
Donde  $(\bar{u}_k, \bar{v}_l)$  son las coordenadas paramétricas con distribución centripetal [14],  $P_{ij}$  son los puntos control,  $N_{j,q}(\bar{v}_l)$  y  $N_{i,p}(\bar{u}_k)$  son las funciones base evaluadas en las direcciones  $\bar{u}_i, \bar{v}_k$  respectivamente y  $Q_{k,l}$  representan los puntos medidos que van a ser interpolados.

Al resolver el sistema de ecuaciones lineales que resultan de la ecuación 3.18 se obtendrán los puntos control  $P_{ij}$ , estos puntos serán los que ayuden a formar la superficie NURBS, a este método se le da el nombre de interpolación global de una superficie [12], en la figura 3.26 se muestra el algoritmo implementado para la interpolación y parametrización de una superficie NURBS.

```
//Calculo de la superficie a partir de la matriz de puntos control P
for(int i=0;i<m1->rows;i++){
//almacenamos una fila de la matriz de puntos control en la matriz line
Mat line = m1->rowRange(i,i+1),v,t,ku;
//FittingL interpola la fila de datos line y regresa el vector
De nodos ku, que son las posiciones u de los puntos  $Q_{k,l}^m(u,v)$ 
u = FittingL(&line,L,&t,&ku);
//La matriz u posee los puntos de la línea interpolada y es asignada a la
matriz Rctrl, Ukv posee las posiciones u de todos los puntos  $Q_{k,l}^m(u,v)$ 
for(int j=0;j<m1->cols+L+1;j++){
if(j<m1->cols){
Rctrl.at<tip1>(i,j) = v.at<tip1>(0,j);
Ukv.at<tip>(i,j) = ku.at<tip>(0,j);
}
}
}
//Se almacenan las posiciones u para la parametrización de la superficie
Ukv.copyTo(*Uk);
//Calculo de la superficie a partir de R
for(int i=0;i<m1->cols;i++){
//almacenamos una columna de la matriz de puntos control en la matriz li-
ne
Mat line = Rctrl.colRange(i,i+1),v,t,kv;
transpose(line,line);
//FittingL interpola la columna de datos line y regresa el vector
De nodos kv, que son las posiciones v de los puntos  $Q_{k,l}^m(u,v)$ 
v = FittingL(&line,L,&t,&kv);
//La matriz v posee los puntos de la línea interpolada y es asignada a la
matriz Pctrl, Vkv posee las posiciones v de todos los puntos  $Q_{k,l}^m(u,v)$ 
for(int j=0;j<m1->rows+L+1;j++){
if(j<m1->rows){
Pctrl.at<tip1>(j,i) = v.at<tip1>(0,j);
Vkv.at<tip>(i,j) = kv.at<tip>(0,j);
}
}
Vv.at<tip>(i,j) = t.at<tip>(0,j);
}
}
//Se almacenan las posiciones v para la parametrización de la superficie
Vkv.copyTo(*Vk);
```

**Figura-3.26.- Algoritmo para la interpolación y parametrización de la superficie NURBS.**

Dado que la técnica de interpolación global solo es una aproximación se hace necesaria una reparametrización, en este caso al tener la superficie modelada y la superficie del objeto (figura 3.27) se puede apreciar que los puntos de la superficie modelada no coinciden en su totalidad con los puntos de la superficie medida, en este caso se debe realizar una parametrización, la cual consiste en encontrar las coordenadas  $(u, v)$  que nos permiten conocer la proyección del punto  $Q_{k,l}$  sobre la superficie modelada, en este caso la proyección se toma como el punto con la distancia mínima entre el punto  $Q_{k,l}$  y la superficie modelada.



**Figura-3.27.- Imagen de una superficie interpolada y con una parametrización deficiente.**

El punto interceptado por la proyección en la superficie modelada le daremos un sufijo  $m$  ( $Q_{k,l}^m$ ), ahora al modelar la superficie con puntos control  $P_{ij}$  y las coordenadas  $(u, v)$  obtenidas de la parametrización se obtiene una superficie cuyos puntos son  $Q_{k,l}^m(u, v)$  y estos puntos son proyecciones de los puntos  $Q_{k,l}$ , que son los puntos medidos de la superficie del objeto en análisis.

Dado que el error obtenido mediante este proceso aun es alto, es necesario un ajuste de los pesos  $w_{k,l}$  asociados a cada punto  $Q_{k,l}^m(u, v)$ , en este caso se aplicó el algoritmo mostrado en la figura 3.28, en donde se da un ajuste recursivo de los pesos  $w_{k,l}$  y donde el ajuste termina cuando el error se incrementa, la expresión 3.19 representa el error de ajuste y bajo la cual se determina si se sigue o se termina el ajuste.

$$f = \left| \frac{\sum_{i=0}^m \sum_{j=0}^n w_{ij} P_{ij} N_{i,m}(u) N_{j,n}(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{ij} N_{i,m}(u) N_{j,n}(v)} - Q_{ij} \right| \quad \text{Ec. -3.19}$$

Donde  $Q_{ij}$  son los puntos medidos y donde  $f$  representa el error de la superficie por el método de mínimos cuadrados [14].

```

//Ciclo para ajuste de pesos
// Función para calcular el error MSE de la superficie ajustada
e2 = ErrorMSE(Sfc, &Mdl);
for(int lop = 0; lop<=30; lop++){
    for(int j=0; j<Ku->rows; j++){
        for(int i=0; i<Ku->cols; i++){
            v = Kv->at<tip>(i, j);
            u = Ku->at<tip>(j, i);
// Calculamos el punto  $Q_{k,l}^m(u,v)$  y lo almacenamos en vec
            vec = NurbsUk(&Pctrl, &t, &t1, L, u, v);
            w = Weights(j, i);
            float err2;
            error = magntd(vec-Sfc(j, i));
// Modificación del peso Wij y cálculo del punto  $Q_{k,l}^m(u,v)$  que se almacena en vecs, se calcula el error y si el error se incrementa el peso no se modifica más.
            Weights.at<tip>(j, i) += 0.5;
            vecs = NurbsUk(&Pctrl, &t, &t1, L, u, v);
            err2 = magntd(vecs-Sfc(j, i));
            if(err2>error){
                Weights.at<tip>(j, i) = w;
                vecs = vec;
            }
// almacenamos en memoria los puntos  $Q_{k,l}^m(u,v)$  de la superficie modelada
            Mdl.at<Vec3f>(j, i) = vecs;
        }
    }
}
// si el error MSE total bajo sigue el ciclo de lo contrario se rompe
float e = ErrorMSE(Sfc, &Mdl);
if(e>e2) break;
else e2 = e;

```

---

**Figura-3.28.- Algoritmo para el ajuste de pesos de la superficie NURBS.**

Finalmente una vez ajustados los pesos estos se almacenan en un arreglo matricial para ser usados posteriormente en el modelado.

### 3.8.2 Modelado con superficies NURBS

Una vez calculados los puntos control y los pesos asociados a cada punto control, el siguiente paso es el modelado de la superficie, en este caso se aplica el algoritmo para modelado mediante NURBS (figura 3.29).

Este algoritmo se esquematiza de manera breve para comprender la estructura del programa de modelado, el primer paso para el modelado es calcular los vectores de nodos para ambas direcciones de la superficie  $u$  y  $v$  (proceso descrito en la sección 2.5.1), estos vectores establecen la relación entre las funciones base y los puntos control de la superficie.

El siguiente paso es calcular las funciones base para obtener la superficie modelada, en este caso se aplican las ecuaciones 3.20 y 3.21 para el cálculo:

$$N_{i,k}(u) = \frac{(u-t_i)*N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)*N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}} \quad \text{Ec.-3.20}$$

$$N_{i,L}(v) = \frac{(v-t'_i)*N_{i,L-1}(v)}{t'_{i+L-1}-t'_i} + \frac{(t'_{i+L}-v)*N_{i+1,L-1}(v)}{t'_{i+L}-t'_{i+1}} \quad \text{Ec.-3.21}$$

Donde  $N_{i,k}(u)$  representa las funciones base de orden  $k$  en la dirección  $u$  y  $N_{i,L}(v)$  representa las funciones base de orden  $L$  en la dirección  $v$ ,  $t_i$  y  $t'_i$  representa el vector de nodos en la dirección  $u$  y  $v$ , con esto se tendrían los parámetros suficientes para el cálculo de la superficie NURBS (Ec.-2.20) cuya expresión es:

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n w_{ij} P_{ij} N_{i,m}(u) N_{j,n}(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{ij} N_{i,m}(u) N_{j,n}(v)}$$

En las figuras 3.30, 3.31 y 3.32 se muestran los algoritmos para el cálculo de cada uno de los parámetros.

```

Calculo del vector de nodos en la dirección u
Calculo del vector de nodos en la dirección v
For(u=0;pm;1){
    For(v=0;pn;1){
        Calculo de las funciones base en la dirección u
        Calculo de las funciones base en la dirección v
        Calculo de la curva
    }
}
    
```

**Figura-3.29.- Estructura del algoritmo para modelar una superficie mediante NURBS, donde pm y pn son los pasos que definen la densidad de puntos en dirección u y v.**

En la figura 3.30 se muestra el algoritmo para el cálculo del vector de nodos, en la línea 7 se declaran las matrices que contendrán la superficie a modelar y los pesos asociados a los puntos control, a partir de la línea 10 se construye el vector de nodos para la dirección  $u$ , la construcción para el vector en la dirección  $w$  se haría de igual manera.

En la figura 3.31 se muestra el algoritmo para el cálculo de las funciones base en la dirección  $u$ , El cálculo de las funciones base requiere que se empiece con las funciones de menor grado, por tanto en la línea 1 se ejecuta un ciclo para buscar la función base activada, es decir cuando el valor de la variable paramétrica  $u$  este en el espacio nodal  $(t_i, t_{i+1}]$ , la posición  $p_{sc}$  nos dice la función base ( $N(p_{sc}, grado)$ ) activada, a partir de aquí se calculan las funciones base de grado mayor con la ecuación 3.15, de la línea 8-24 se realiza este cálculo, en la línea 19 se almacena el valor de la función de orden  $k$ , para posteriormente utilizarla en el cálculo de la superficie.



```

//NURBS(superficie a modelar, paso en dirección m, paso en dirección n)
Mat NURBS(Mat*m1,float pm,float pn){
//-----surfaces NURBS- "not uniform rational B-Splines"
int L = 3,K = 3; //GRADO DE LA SUPERFICIE
Mat Pctrl,h,Mdl;
int n = Pctrl.cols, m = Pctrl.rows;
//inicializamos las matrices Mdl y h, [7]
Mdl = Mat::zeros(ceil((m-K+1)/pm)+1,ceil((n-L+1)/pn)+1,CV_32FC3);
h = Mat::ones(m,n,CV_32FC1);
//creamos el vector de knots mat.create(rows,cols,type) [10]
cv::Mat t(1,n+L-1,CV_16SC1);
for(int j=0;j<=n+L-1;j++){
    if(j<L) t.at<tip2>(0,j)=0;
    if(j>=L&&j<=n-1) t.at<tip2>(0,j)=j-L+1;
    if(j>n-1) t.at<tip2>(0,j)=n-L+1;
}

```

Figura-3.30.- Algoritmo para crear los vectores de nodos de la superficie NURBS.

```

//llenamos la matriz N(i,p) [1]
for(int i=0;i<n;i++){
    if(t(0,i)<=u&&t(0,i+1)>u){psc = i; break;}
}
//----- calculamos la funcion base activada
N.at<tip>(psc,1)=1;
//----- cálculo de las funciones de grado mas alto
int grado = 2;
for(int i =psc;i<psc+1;i--){ [8]
    if(i<0) break;
    for(int k=count;k<=L;k++){
        a = ((u-t(0,i))/(t(0,i+k-1)-t(0,i)))*N(i,k-1);
        si a es un valor NAN
        a=0

        b = (((t(0,i+k)-u))/(t(0,i+k)-t(0,i+1)))*N(i+1,k-1);
        si b es un valor NAN
        b=0;

        N.at<tip>(i,k) = a + b; [19]
        N.at<tip>(i+1,k-1) = 0;
    }
    if(i==psc-1) count += 1;
    if(count>L) break; [24]
}

```

Figura-3.31.- Algoritmo para calcular las funciones Base de la superficie NURBS.

En la figura 3.32 se muestra el algoritmo para obtener la superficie NURBS, donde para el cálculo de esta se utiliza la ecuación 2.20, en la línea 1 y 2 se tiene dos ciclos **for**, limitados por dos variables que restringen el cálculo solo para los puntos control involucrados tanto en la dirección **u** como **v**, en la línea 3 se realiza la suma para el cálculo de la superficie y en la 6 se almacena en memoria, para posteriormente ser mostrada.

```
//-----cálculo de la curva-----  
for(int j=lim1;j<=psc1;j++){ -1  
  for(int i=lim;i<=psc;i++){  
    zum += w(j,i)*N(i,L)*N1(j,K)*Pctrl(j,i); -3  
    zum1 += w(j,i)*N(i,L)*N1(j,K);  
  }  
}  
Mdl.at<Vec3f>(y,x)= zum/zum1; -6
```

Figura-3.32.- Algoritmo para calcular la superficie NURBS

### 3.9 INTEGRACIÓN DEL SISTEMA ÓPTICO Y MECÁNICO

En la figura 3.33 se muestra el sistema implementado para la recuperación de la forma de objetos mecánicos, en este caso se observa el escaneo de un engrane de 52 dientes.

Este sistema realiza una rotación del objeto para lograr el escaneo de su superficie, el posicionamiento es controlado por medio de un software que permite ajustar la velocidad de rotación así como ejecutar movimientos a diferentes ángulos, permitiendo con esto un escaneo uniforme y con diferentes resoluciones.

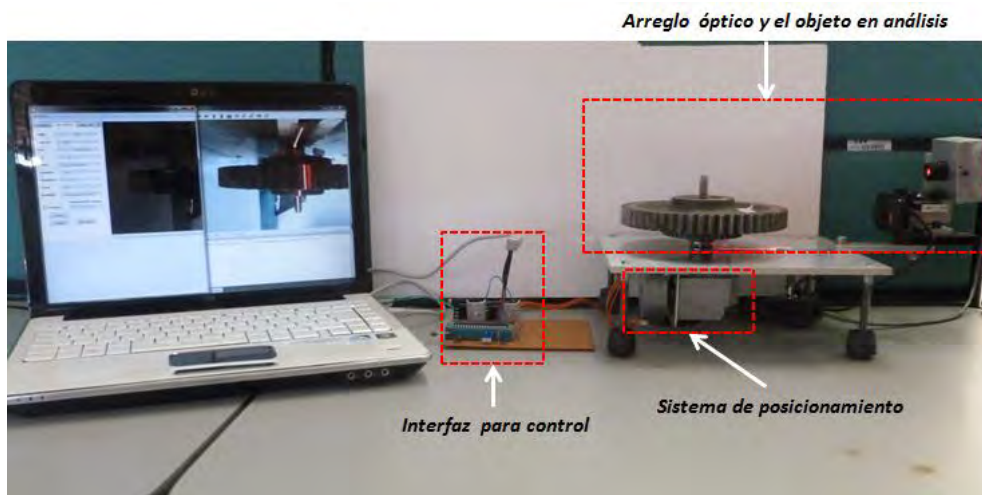


Figura-3.33.- Sistema para reconstrucción de piezas mecánicas.

La última parte del proceso se constituye de un software para procesamiento digital de las imágenes tomadas, un proceso de reconstrucción y un modelado geométrico de la superficie.

El resultado de este proceso son formas geométricas que pueden ser llamados objetos virtuales [17], estos objetos pueden ser analizados matemáticamente, ya que quedan descritos matemáticamente por la superficie modelada mediante NURBS.

La computadora utilizada para este proceso es una PC portátil de doble núcleo con procesador a 2.1 Ghz, sistema operativo Windows 7 a 64 bits, marca Hewlett Packard.

# CAPITULO IV.- RESULTADOS EXPERIMENTALES

---

En el presente capítulo se darán a conocer los resultados obtenidos de la implementación del sistema para reconstrucción y modelado de objetos mecánicos, se hará un análisis de la incertidumbre del sistema y se calcularán los errores referentes a los datos recuperados.

Para mostrar los gráficos y superficies 3D se hizo uso de Qt con la implementación de las librerías de Qwt3Dplot, esta es una librería libre para uso en ambientes de programación de c++ y la cual está basada en OpenGL, esta proporciona una gran variedad de widgets para mostrar gráficos y formas 3D en QtCreator.

## 4.1 RECUPERACIÓN DE SUPERFICIES TRIDIMENSIONALES.

El requerimiento de sistemas para mediciones más complejas y sofisticadas, además del gran desarrollo tecnológico y la creación del láser, marcaron la nueva tendencia tecnológica para realizar mediciones [18].

La metrología óptica comprende una serie de métodos para realizar mediciones dimensionales de objetos físicos, tales métodos pueden ser basados en luz coherente o incoherente, las ventajas de estos procesos es que son procesos en tiempo real, poseen gran exactitud y se pueden implementar por medio de computadoras.

Debido a las características antes mencionadas, es que la metrología óptica es un método viable para cubrir las necesidades metrológicas de la industria, en este trabajo de tesis particularmente se implementó la técnica de proyección de línea, que es un método basado en un haz coherente y que a su vez posee grandes cualidades, tanto en exactitud como en la facilidad de su implementación.

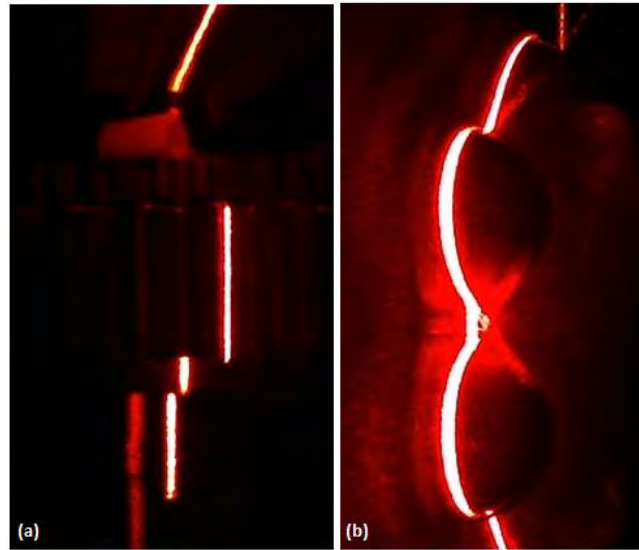
### *4.1.1 Reconstrucción de la superficie de objetos mecánicos.*

Como se había mencionado para adquirir la información necesaria para la reconstrucción 3D, es necesario tomar una serie de imágenes de la línea láser, cada una de estas ha de contener información de una sección transversal de la superficie, procesando de manera digital cada una de las filas pertenecientes a cada imagen, se ha de recuperar la posición de la línea laser por medio de la aproximación gaussiana (sección 2.4.1).

Cada una de las posiciones recuperadas se han de guardar en arreglos de memoria, ya que esta información está en pixeles, es necesario procesarla posteriormente para obtener mediante la curva de calibración las dimensiones de la superficie recuperada.

La figura 4.1 muestra imágenes de la proyección de la línea sobre la superficie de objetos de forma libre, cada una de estas imágenes se convierten del formato RGB a un formato de escala de grises, esto facilitara el cálculo del posicionamiento de la línea. Por último cada una de las imágenes tiene un formato preestablecido de 640x480 pixeles, donde para el engrane también se utilizó el formato de 1280x720 pixeles.

Para escanear el engrane se debe tener una separación pequeña entre la cámara y el proyector, esto disminuye las oclusiones presentes en la imagen extraída por el sistema de visión y consecuentemente se tiene una mejor visión de la línea.



*Figura.-4.1.- Escaneo laser de (a) un engrane recto, (b) un objeto con forma de hipopótamo.*

Como se había mencionado en capítulos anteriores, las redes neuronales basadas en funciones de Bézier son de interés para el ajuste de la curva de calibración, en este sentido el proceso para obtener los parámetros de calibración es omitido, ya que el sistema se calibra de acuerdo a un objeto referencia y el procesamiento digital. Con esto se tiene un proceso de calibración más exacto, ya que se evitan los errores de medición externos al sistema.

Para convertir los datos almacenados en memoria a valores dimensionales es necesario introducir el desplazamiento  $S_{ij}$  a la red neuronal expuesta en 3.4.1, obteniendo así la variable paramétrica  $u$  que nos relacionara el valor de la altura  $h(x, y)$  al desplazamiento  $S(u)$ , con esta conversión se tienen los valores dimensionales de la superficie almacenados en memoria.

Los objetos analizados en este trabajo de tesis fueron un objeto con forma de hipopótamo y un engrane de dientes rectos, en el primer objeto solo se aplicó el modelado para corroborar la eficiencia del algoritmo NURBS, mientras que en el engrane se aplicó un proceso de reconstrucción, dimensionamiento y modelado de su superficie, con la finalidad de obtener la exactitud de representación del objeto.

El tiempo de reconstrucción para un engrane de 52 dientes y diámetro = 6.5in es de 10min, en este caso se toman una serie de imágenes cada 0.6°, siendo un total de 600 imágenes las que describen la superficie del objeto escaneado.

Debido a que el engrane posee gran cantidad de detalles en la forma de sus dientes, fue necesario tomar una mayor cantidad de imágenes, por tanto se propone otro método alternativo para la reconstrucción de este tipo de objetos mecánicos, el método propuesto es la reconstrucción parcial en donde se escanea uno o varios dientes que al ser reproducidos  $n$  veces estos reconstruyen la totalidad de la superficie, esto lo podemos aplicar ya que se trata de una estructura periódica y predecible.

Para realizar la reconstrucción parcial se escaneo solo un par de dientes y esta porción de superficie se replicó para reconstruir el engrane en su totalidad, en este caso el error rms se mantuvo.

#### 4.1.2 Determinación de la exactitud de la red neuronal Bézier

Para el cálculo de la exactitud de la red neuronal de Bézier, es necesaria la comparación de los datos arrojados por la red neuronal y los datos de la superficie medidos con la máquina CMM (Coordinate Measure Machine). La expresión para calcular la precisión será la del error rms expresado como:

$$Erms = \left( \frac{1}{n} \sum_{i=1}^n (ho_i - hc_i)^2 \right)^{1/2} \quad \text{Ec. -4. 1}$$

Donde  $ho_i$  son los datos medidos en la maquina CMM,  $hc_i$  son los datos calculados por la red neuronal y  $n$  es el número de muestras. Para los datos recuperados del engrane (figura 4.2) el número de datos fue  $n = 700$  con un error rms = 0.6206 mm, teniendo  $h_m = 77.581 \text{ mm}$  de los datos antes mencionados y la expresión 4.2 podemos calcular el error  $MSE(\%)$ .

$$MSE(\%) = \frac{Erms}{h_m} * 100 \quad \text{Ec. -4. 2}$$



Figura.-4.2.- Escaneo laser de (a) un engrane recto y (b) un objeto con forma de hipopótamo.

El error MSE(%) resultante es de 0.8% para el engrane de la figura 4.2, este error corresponde a la reconstrucción del objeto por medio de la línea laser.

#### 4.1.3 Calculo de la resolución del sistema

El concepto de la resolución nos ayuda a cuantificar la mínima cantidad de una magnitud que el sistema puede discriminar, en este caso se tiene la resolución en las alturas  $h_i$  referente a la topografía del objeto, la resolución del sistema de posicionamiento y la resolución de la cámara.

Para calcular la resolución del sistema se proyectó una línea laser en el plano referencia del sistema, donde se posicionaron 4 patrones que poseen diferentes alturas, los desplazamientos para cada patrón quedan expresados en la tabla 4.1.

**Tabla 4.1.- Desplazamientos de los patrones usados para calcular la resolución del sistema**

Patrón	Desplazamiento medido
1	437.32píxeles
2	429.57 píxeles
3	429.52 píxeles
4	429.35 píxeles

En el cálculo de la resolución para el eje  $x$  que corresponde al eje de los desplazamientos  $S_{ij}$ , se tomaron una serie de imágenes de los 4 patrones, de acuerdo a los desplazamientos obtenidos se pudo calcular que el mínimo desplazamiento discriminado por el sistema fue entre el patrón 2 y 4, lo cual dio una resolución de 0.22 píxeles.

Introduciendo los desplazamientos de los patrones 2 y 4 en la red neuronal Bézier podemos calcular la resolución para las alturas  $h_i$ , por tanto las alturas calculadas por la red neuronal para los patrones 2 y 4 son  $h_2 = 6.665cm$  y  $h_4 = 6.693cm$ , con estos datos podemos calcular la resolución como  $resh = h_i - h_f = (66.93 - 66.65)mm = 0.28mm$ , por tanto la mínima cantidad medible por el sistema es de 0.28mm.

Respecto a la resolución angular del sistema de posicionamiento se conoce que se tiene un motor a pasos de  $15^\circ$  por paso y un engrane con acoplamiento de tornillo sin fin, en su conjunto nos da una resolución angular de  $0.3^\circ$ , permitiendo obtener hasta un total de 1200 imágenes por cada superficie escaneada.

Dado que la resolución en el eje  $y$  (eje donde se determina la relación mm/píxel) depende de la medición del patrón de calibración, es necesario calcular la mínima cantidad discriminada por la cámara.

Para poder realizar el cálculo de esta resolución es necesario conocer varios detalles de la cámara, en este caso se conoce que para la cámara C525 logitech, se tiene un tamaño de píxel =  $1.75 \mu m$  y donde el tamaño del sensor corresponde a  $1/5in$ .

El primer factor a calcular es *PMAG* (potencia de magnificación) la cual queda descrita por la relación del tamaño del sensor (magnitud horizontal) y el campo de visión (horizontal), ecuación 4.3 [4].

$$PMAG = \frac{\text{Tamaño sensor (mm)}}{FOV (mm)} \quad \text{Ec.-4.3}$$

Con un campo de visión de 185.25mm y un tamaño de sensor de 2.8mm se obtiene una *PMAG* de:

$$PMAG = \frac{2.8}{185.25} = 0.01511$$

La resolución del objeto puede ser calculada a partir de la *PMAG* y la resolución de la imagen proporcionada por el sistema de visión.

*Calculo de la resolución de la cámara*

Esta resolución depende esencialmente de la composición del sensor de imagen, es decir que entre más pequeños sean los pixeles, esta resolución mejorara y queda expresada por la ecuación 4.4.

$$Res_{cam} = lp * tamaño\_pixel \quad \text{Ec.-4.4}$$

Donde los pares de líneas  $lp = 2$ , se refieren a dos filas o columnas de pixeles adyacentes que son capaces de discriminarse figura 4.3.

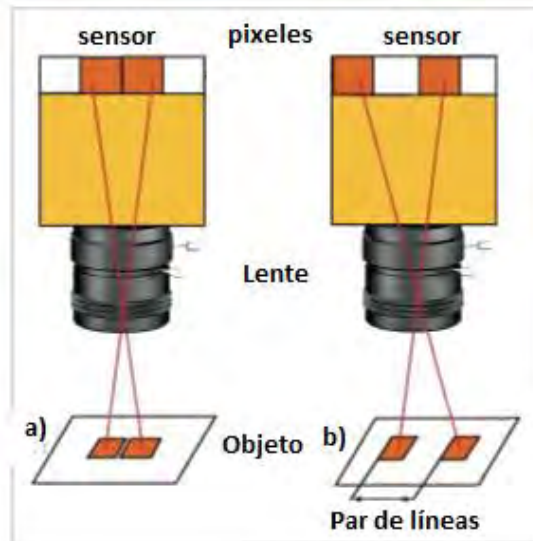


Figura.-4.3.- Imagen referente a la resolución espacial de una cámara electrónica, imagen tomada de referencia [35]



Sustituyendo los valores en la ecuación 4.4 para conocer la resolución de la cámara.

$$Res_{cam} = 2 * 1.75\mu m = 3.5\mu m$$

Con este resultado ahora se puede calcular la resolución del sistema, en este caso llamada resolución del objeto y se refiere a la distancia mínima que el sistema puede discriminar del objeto en escena.

*Calculo de la resolución del eje y*

La resolución del eje y queda descrita por la ecuación 4.5.

$$Res_{obj} = Res_{cam}/PMAG \quad \text{Ec.-4.5}$$

Donde la  $PMAG = 0.01511$  y  $Res_{cam} = 3.5\mu m$ , sustituyendo los valores en la ecuación 4.5.

$$Res_{obj} = \frac{3.5\mu m}{0.01511} = 231.56\mu m$$

En este caso la resolución en el eje y es de  $0.231mm$

## 4.2 EVALUACIÓN DEL SISTEMA Y LOS DATOS EXTRAIDOS PARA EL ENGRANE.

Para la evaluación de los datos obtenidos por medio del sistema se tomaron 67 mediciones de la superficie del objeto analizado, el cual se basó en un arreglo de medición por contacto o palpador. El objetivo era extraer datos con una exactitud mayor que la que posee el sistema de visión implementado.

La expresión para calcular el error rms de los datos respecto a una referencia es la ecuación 4.6, en esta expresión  $h_o$  es la altura medida por el sistema de contacto y  $h_c$  es la altura medida por el sistema de visión.

$$rms = \sqrt{\frac{1}{n} \sum_{i=1}^n (h_o - h_c)^2} \quad \text{Ec.-4.6}$$

En la figura 4.4 se tiene la imagen de la superficie medida por medio el sistema de contacto, esta superficie será la que se tome como referencia para el cálculo del error rms de las superficies recuperadas por el sistema de visión.

El sistema de posicionamiento utilizado para las mediciones por contacto fue el sistema de medición de coordenadas TECHNO, este sistema realiza movimientos de hasta 0.0005 in, lo cual permite buena exactitud y precisión para medir la superficie del objeto.

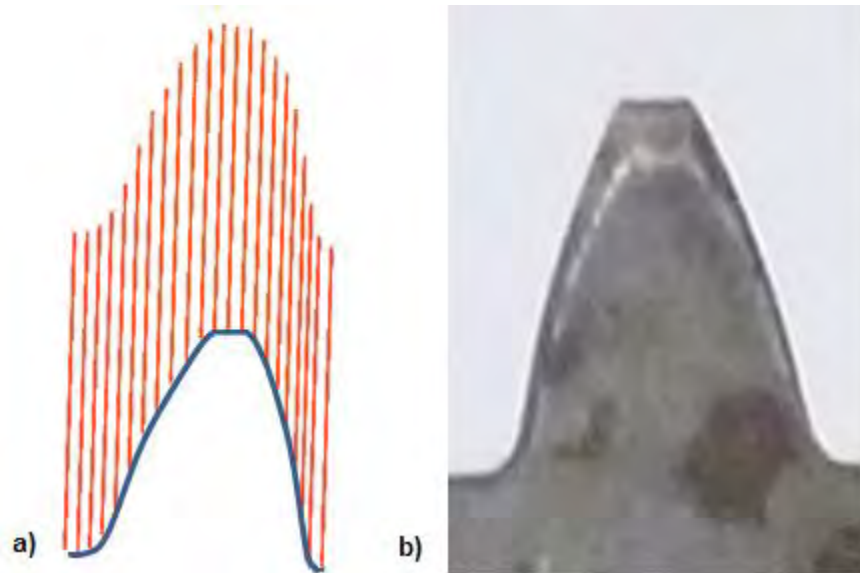


Figura.-4.4.- a) superficie escaneada, b) perfil del diente escaneado.

Dado que el nivel de confianza debe ser de 95%, es necesario saber si el número de muestras tomadas son suficientes para alcanzar este nivel de confianza, la ecuación 4.7 muestra la relación que nos permitirá conocer el número de muestras de acuerdo al nivel de confianza.

$$n = \left( z_{\alpha} \frac{\sigma_x}{e} \right)^2 \quad \text{Ec.-4.7}$$

En esta expresión  $n$  es el número de muestras,  $z_{\alpha}$  es la confianza,  $e$  es el error y  $\sigma_x$  corresponde a la desviación estándar de las alturas recuperadas.

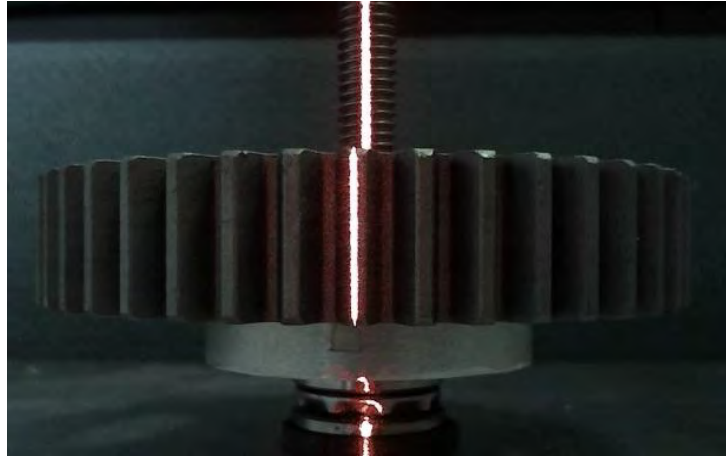
De los datos obtenidos para la superficie de un diente de engrane, se tiene un error rms de 0.93962%, donde la desviación estándar corresponde a 2.3215mm y con un nivel de confianza de 95%, sustituyendo en la ecuación 4.2 tenemos una  $n$  de:

$$n = \left( 1.96 * \frac{2.3215}{0.67981} \right)^2 \cong 44.7996$$

Como se puede ver el número de datos extraídos es suficiente para tener una confianza del 95% de acuerdo a la tabla de confianza [19].

#### 4.2.1 repetibilidad del sistema.

La repetibilidad se calculó de acuerdo a una serie de imágenes de una línea laser proyectada sobre el perfil del engrane (fig. 4.5), se analizó la posición de la misma teniendo en cuenta un proceso repetitivo que estuviera bajo las mismas condiciones, el número de muestras tomadas fueron 90 imágenes para el análisis de la repetibilidad en el sistema.



**Figura.-4.5.- proyección laser usada para calcular la repetibilidad del sistema**

De las muestras tomadas se obtuvo una desviación estándar de 0.1625 pixeles y un valor promedio de  $x = 9.6965$  pixeles.

La ecuación 4.8 muestra la relación para calcular la repetibilidad del sistema [20], en este caso tendríamos una medida cuantificable de la capacidad del sistema de visión para repetir una medida, en esta expresión  $x_{prom}$  se refiere a la esperanza matemática del mesurando y  $\sigma x_i$  es la desviación estándar del error en los datos.

$$repetibilidad_i(\%) = \frac{x_{prom_i} - \sigma x_i}{x_{prom_i}} * 100 \quad \text{Ec.-4.8}$$

Sustituyendo los valores obtenidos en la expresión anterior:

$$repetibilidad_i(\%) = \frac{9.6965 - 0.1625}{9.6965} * 100 = 98.32\%$$

#### 4.2.2 incertidumbre del sistema

Como parte esencial de este proyecto, se debe calcular la exactitud con que el sistema de visión realiza las mediciones del mesurando  $h_i$ , para lograrlo se debe determinar la incertidumbre del sistema de medición, debido a que el mesurando depende de otras variables se analizara la incertidumbre característica combinada [19], esto nos dará una idea del valor real del mesurando  $h_i$ .

La ecuación 4.9 representa la propagación del error del mesurando y que depende de otras variables  $x_i$ , que son medibles y cuantificables.

$$u_c^2(y) = \left(\frac{\partial y}{\partial x} u(x)\right)^2 + \left(\frac{\partial y}{\partial x_1} u(x_1)\right)^2 + \dots + \left(\frac{\partial y}{\partial x_n} u(x_n)\right)^2 \quad \text{Ec.-4.9}$$

Donde  $u^2(x_i)$  es la varianza,  $\frac{\partial y}{\partial x}$  son los coeficientes de sensibilidad y estos cuantifican el grado de influencia de la variable  $x_i$  sobre el valor del mesurando  $y$ . Por otro lado para una magnitud de entrada  $x_i$ , obtenida a partir de  $n$  observaciones repetidas e independientes, la incertidumbre típica  $u(x_i)$  de su estimación  $x_i$  queda definida por la ecuación 4.10.

$$U_x = \sqrt{\left(\frac{\sum(x_i - x_{prom})^2}{n-1}\right)} \quad \text{Ec.-4.10}$$

Debido a que en este trabajo se implementó una red neuronal para reducir el número de variables involucradas en el cálculo del mesurando  $h_i$ , se conoce que los únicos parámetros involucrados en el cálculo del mesurando serian la variable paramétrica  $v$  y el valor de la posición de la línea, que en este caso se representa por  $x$ .

La expresión de la propagación del error para las alturas  $h_i$  queda expresadas como:

Desarrollando la expresión 4.9 para encontrar la incertidumbre se tiene:

$$u_c^2(h) = \frac{\partial h^2}{\partial v} * u^2(v) \quad \text{Ec.-4.11}$$

$$u^2(v) = \frac{\partial v^2}{\partial x} * u^2(x) \quad \text{Ec.-4.12}$$

Para calcular la incertidumbre  $u^2(v)$  se tiene que la relación entre la variable paramétrica  $v$  y la posición  $x$  es una función lineal, de modo tal que el coeficiente de sensibilidad queda expresado como:

$$\frac{\partial v^2}{\partial x} = \left(\frac{1}{x_{max} - x_{min}}\right)^2$$

De los datos para la repetibilidad se tiene que la desviación estándar de  $x$  es de 0.110555 pixeles, por tanto se tiene sustituyendo los resultados en la ecuación 4.12:

$$\frac{\partial v^2}{\partial x} * u^2(x) = \left(\frac{1}{417.274 - 509.067}\right)^2 * (0.110555)^2 = 1.45056 * 10^{-6}$$

Para el cálculo de  $u_c^2(h)$  se tiene que  $h$  representa la función generada por la red neuronal de Bézier, entonces el coeficiente de sensibilidad de la expresión 4.7 se calcula a partir de la curva de calibración, donde se tomó el valor mayor del coeficiente y se obtuvo un coeficiente de 93.496 mm, sustituyendo los valores en la ecuación 4.11:

$$u_c^2(h) = (93.496 \text{ mm})^2 * 1.45056 * 10^{-6} = 12.68 * 10^{-3} \text{ mm}^2$$

Donde la incertidumbre total del sistema es:

$$u_c(h) = 0.1126 \text{ mm}$$

Con esta incertidumbre se tiene un sistema que posee poco error en la recuperación de la topografía de los objetos, además se considera un valor bajo ya que representa la incertidumbre total del sistema.

### 4.3 RESULTADOS DE LA RECONSTRUCCIÓN Y MODELADO DE OBJETOS MECÁNICOS.

La figura 4.6 muestra la reconstrucción total de un engrane de dientes rectos en un formato de 480x640 pixeles, para esta reconstrucción se utilizaron 7200 puntos, la cual corresponde a un total de 600 imágenes de la línea para describir la superficie del objeto, en este caso la superficie recuperada obtuvo un error de forma del 0.8%.

El modelado de la superficie antes descrita se puede ver en la figura 4.7, para esta se calculó un error de modelado  $RMS = 0.09306mm$ , con una altura media  $hm = 7.76106cm$  se tiene un  $RMS(\%) = 0.1199\%$  y donde el tiempo transcurrido desde el escaneo hasta el modelo de la pieza fue de aproximadamente 7.2 min.

El error de la representación del objeto se obtiene sumando los errores obtenidos tanto por la reconstrucción como por el modelado, por tanto para el engrane se obtiene una exactitud en su representación de 0.6923mm que representa un error relativo de 0.91%.

Ya que un engrane posee un patrón repetitivo, se consideró el reconstruir este objeto a partir de un escaneo parcial de la pieza, la figura 4.8 muestra una reconstrucción parcial del engrane en donde la superficie está formada a partir del escaneo de un solo diente del engrane, para reconstrucciones de este tipo el error  $RMS$  se mantuvo, el tiempo promedio para una reconstrucción por medio de esta técnica rondaba los 2.3 minutos, con lo cual se tiene una reducción importante en el tiempo de la reconstrucción y modelado de engranes.

Es importante especificar que el tiempo de modelado puede variar de acuerdo a que tan definida queremos la superficie, las imágenes en esta tesis se presentan con un formato de puntos para poder visualizar la densidad del modelado.

En la figura 4.9 se muestra un acercamiento de la superficie del engrane, procesada y en un formato de imagen de 1280x720 pixeles, la cual esta modelada por medio de la técnica NURBS. Para esta reconstrucción se obtuvo un error  $MSE\% = 0.77\%$ , el error de modelado de esta superficie fue  $RMS = 0.09621mm$  con una altura media  $hm = 7.7616cm$  se tiene un error relativo de  $RMS(\%) = 0.1239\%$ , por tanto la exactitud en su representación es de 0.6838mm lo cual equivale a un error relativo de 0.8939%.

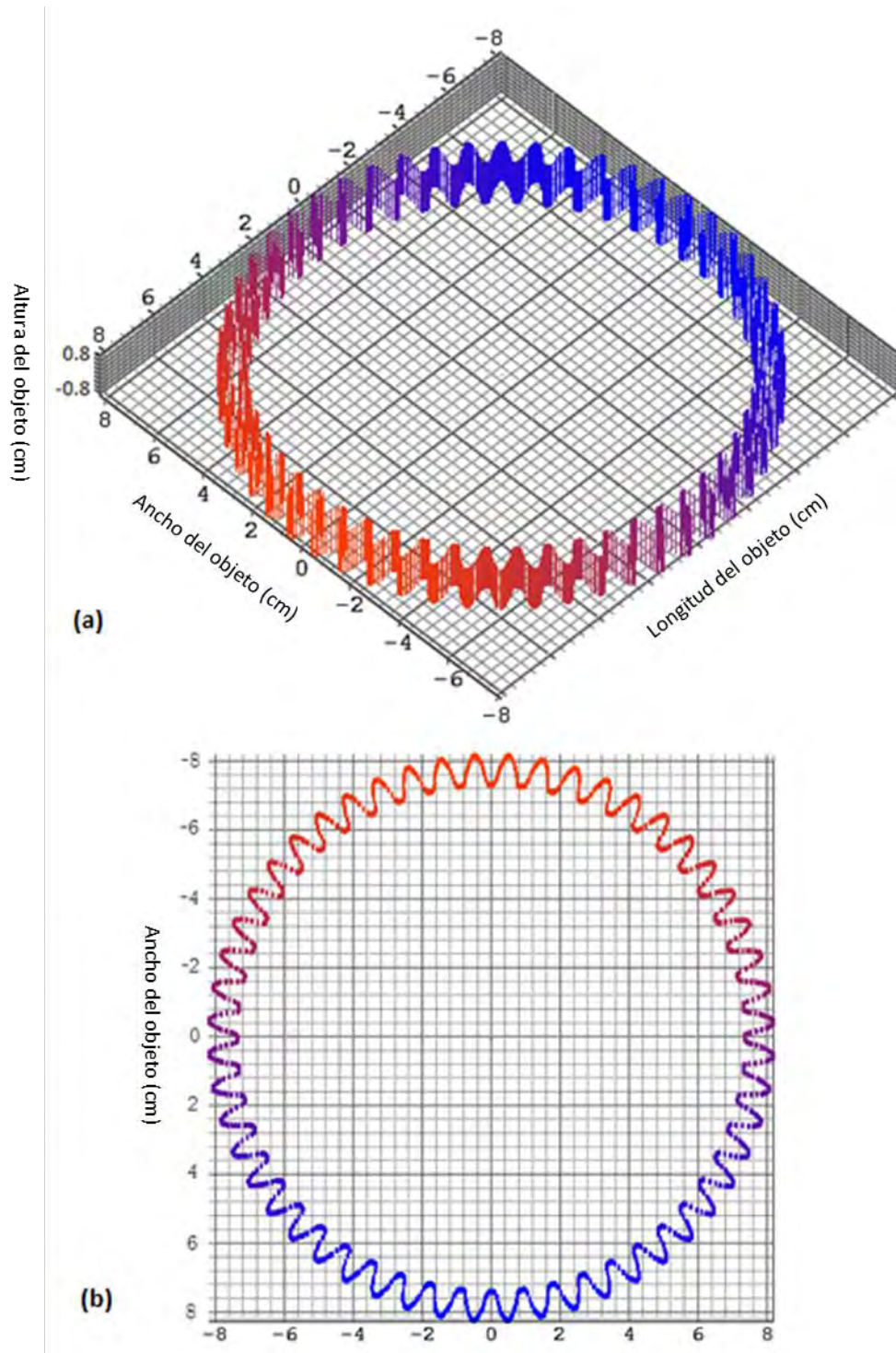


Figura.-4.6.- nube de puntos para un engrane de dientes rectos, reconstruida por medio de la proyección de línea, (a) vista isométrica del engrane y (b) vista superior del engrane.

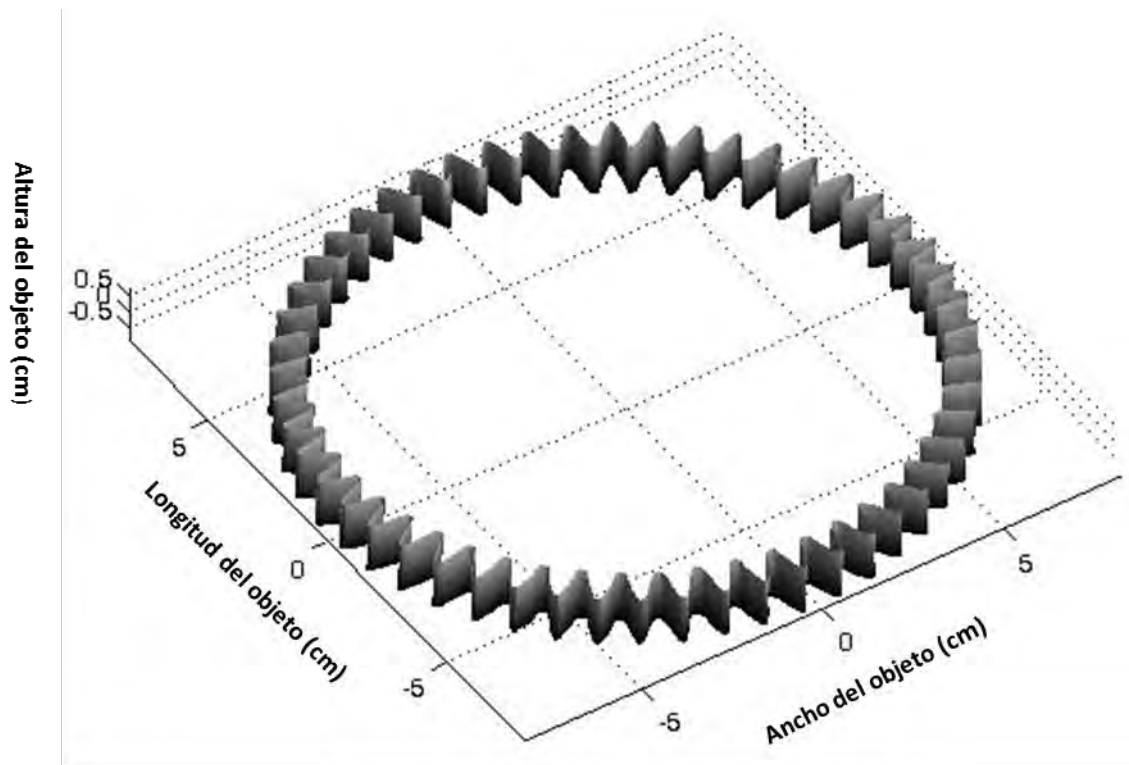


Figura.-4.7.- Superficie reconstruida y modelada para un engrane de dientes rectos.

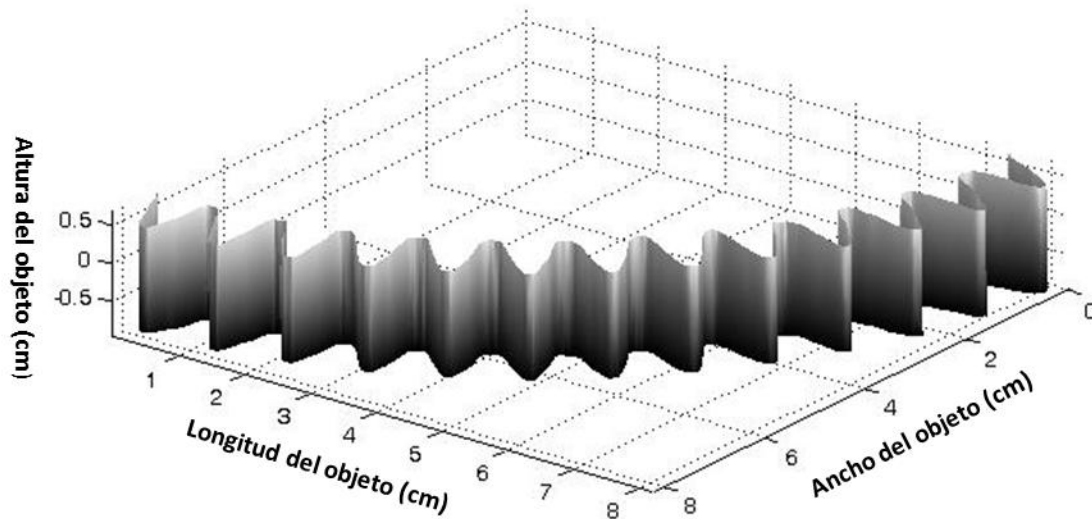
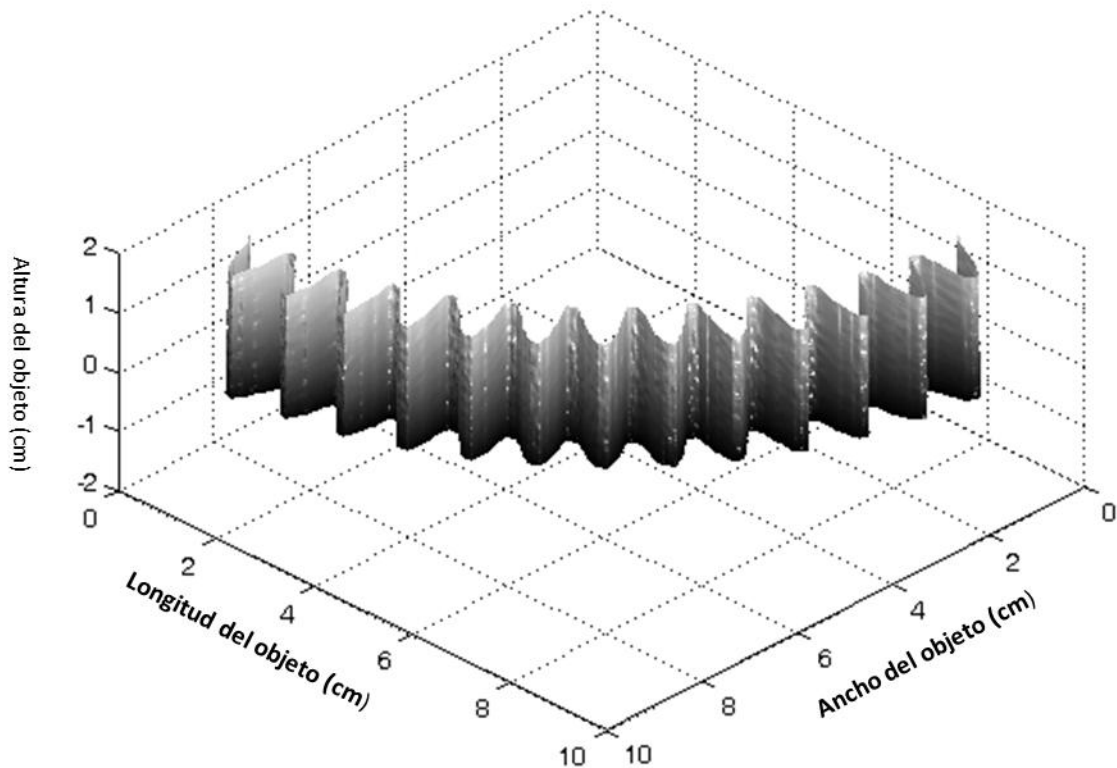


Figura.-4.8.- Superficie reconstruida parcialmente y modelada para un engrane de dientes rectos.

En la figura 4.9 se muestra el mapeo de una sección de la superficie de un engrane de dientes rectos, en esta figura se puede observar un mayor número de detalles debido a que existe más información de la superficie que la obtenida en el formato de menor resolución (640x480 pixeles).

En esta superficie se puede constatar que el efecto de "smoothing" generado por el modelado se ve disminuido, esto se debe a que el número de puntos ajustados es mayor y por tanto se pueden apreciar más las características de la superficie recuperada.



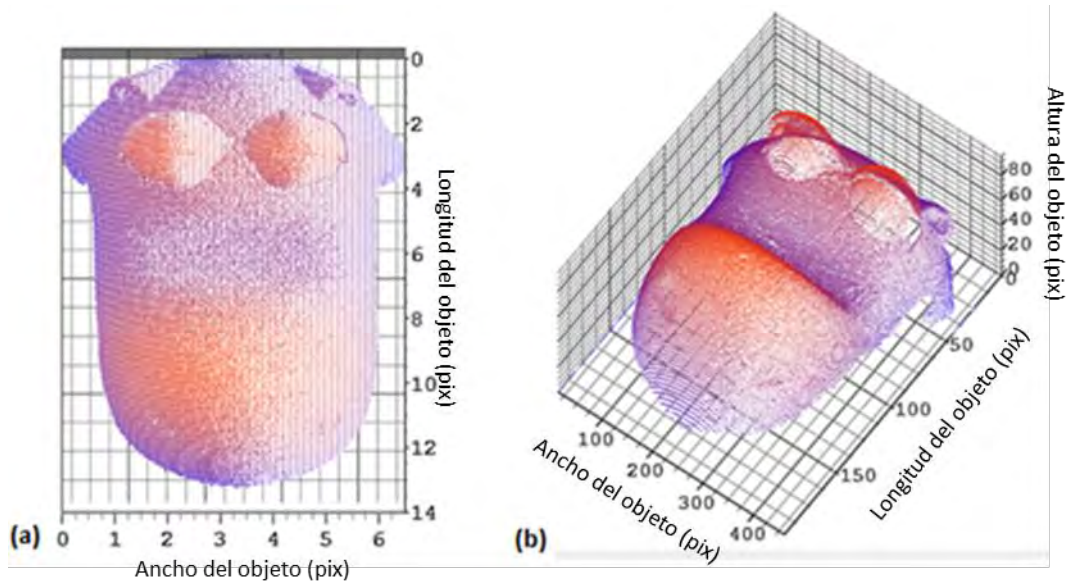
*Figura.-4.9.- Superficie modelada para un engrane de dientes rectos.*

Como se puede apreciar en las imágenes, una característica importante del modelado es su capacidad de interpolar los datos de acuerdo al grado que se desee, es decir si se quiere obtener una superficie más lisa es necesario generar un modelado de un grado alto (por ejemplo 3°), donde los NURBS deberán mantener un continuidad  $C_2$  y por tanto no pasara exactamente por todos los puntos de la superficie modelada, en consecuencia se tendrá un incremento en el error de modelado.

Otro factor importante en el modelado con NURBS es el ajuste de pesos, ya que este permite modelar una superficie lo más exacto posible, a diferencia del caso anterior este proceso sirve para disminuir el error del modelo generado.

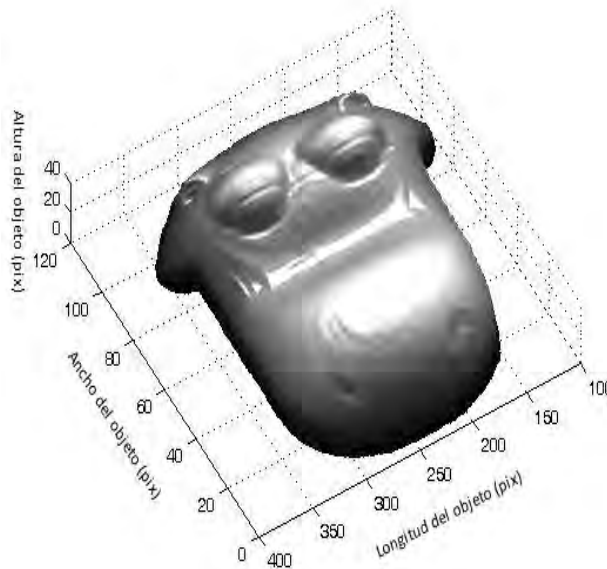


En la figura 4.10 se muestran los puntos recuperados para la superficie de un objeto en forma de hipopótamo cuyo material del objeto es plástico, en este caso la superficie se forma de datos recuperados por medio de la proyección de línea.



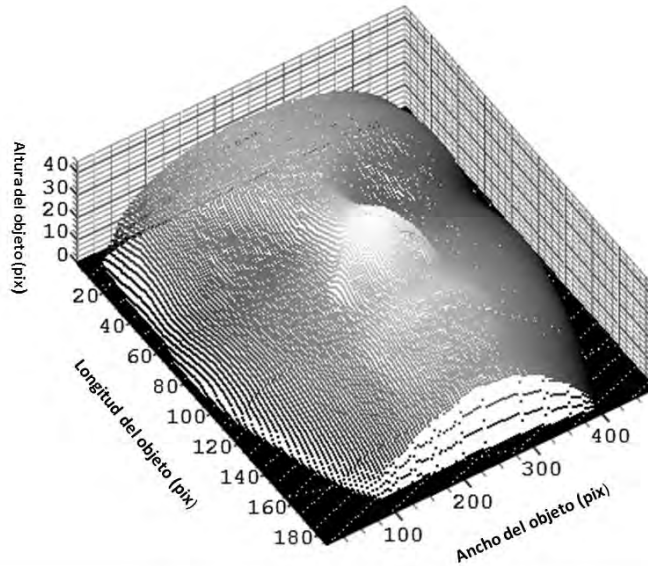
**Figura.-4.10.- Superficie reconstruida por medio de la técnica de luz estructurada, (a) vista superior y (b) vista isométrica.**

En la figura 4.11 se tiene el modelado de la superficie del hipopótamo, el error RMS para esta superficie fue de 0.1128425 mm, con una altura media  $hm = 13.949\text{ cm}$  se tiene un error relativo  $RMS(\%) = 0.08089\%$ .



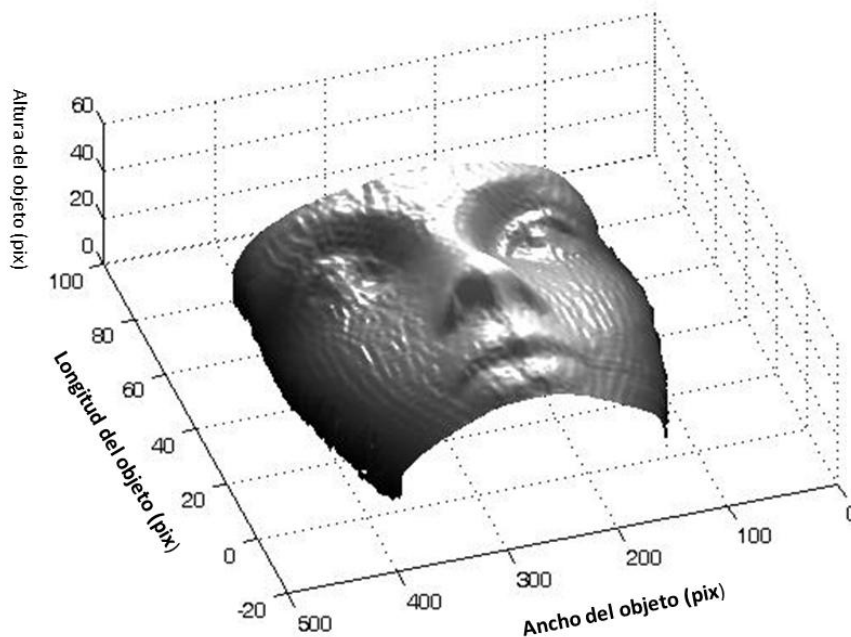
**Figura.-4.11.- Superficie modelada para un objeto en forma de hipopótamo.**

En la figura 4.12 se tiene la forma de un rostro conformado en unicel (poliestireno expandido), fue reconstruida por medio de la técnica de proyección de línea, esta forma posee gran cantidad de detalles debido a la textura del unicel y a los rasgos propios de la forma.



**Figura.-4.12.- Perfil de un rostro reconstruido por medio de la técnica de luz estructurada.**

En la figura 4.13 se muestra el modelado de la forma de rostro, el error  $RMS = 0.32979 \text{ mm}$  , en este caso se tiene un  $hm = 18.696 \text{ cm}$  y el error  $RMS(\%) = 0.1763\%$ .



**Figura.-4.13.- Superficie modelada para una forma de rostro.**

De acuerdo a los resultados obtenidos se demuestra que la técnica de modelado con NURBS es un método viable, el cual es muy natural y fácil de implementarse en procesos de ingeniería inversa y control dimensional, esta técnica presenta un buen comportamiento en el modelado de superficies con texturas no uniformes, en general las texturas presentes en los objetos modelados son debido a la información presente en el eje y que corresponde a la proyección de la línea laser, respecto a la textura contenida entre imágenes de la línea se pierde debido al modelado.

La capacidad de las funciones NURBS para manipular tanto su continuidad geométrica como sus funciones base, permite un modelado eficiente de las superficies generadas y a su vez contribuye en la eliminación del ruido de los datos modelados y los cambios erráticos en la superficie trabajo, dando una textura suavizada de las formas generadas.

La diferencia entre otras técnicas de modelado que se basan en funciones polinomiales uniformes (tal como Bézier) y los NURBS, es que estos poseen una facilidad para el modelado de las superficies, donde se pueden manejar parches o secciones de superficie mucho más grandes que las manejadas por Bézier. Debido a que la naturaleza no uniforme de los NURBS hace más fácil interpolar los puntos con funciones cuyo grado es independiente del número de datos, al contrario de otras técnicas con funciones polinomiales de grado  $n$ , donde  $n$  depende del número de puntos a interpolar.

Dado que Bézier pasa a través de todos los puntos que forman su curva de control, puede llegar a errores del orden 0.001%, pero a diferencia de los NURBS ésta no mantiene su continuidad geométrica, lo cual hace poco natural su representación para un análisis geométrico.

Se usaron objetos diferentes al engrane para demostrar que la técnica de modelado implementada puede representar cualquier forma con errores por debajo del 0.2%. En superficies con mucho ruido y formas agudas, se pudo observar que el error tendía a incrementarse debido a que la técnica NURBS no puede modelar el error presente en los datos y a su característica de suavizar las superficies.

El tiempo de procesamiento para reconstruir y modelar un objeto real es bastante viable (tiempos obtenidos menores a 10 min), lo cual genera un método accesible para prototipado de objetos 3D cuyo propósito principal es su análisis geométrico.

## CONCLUSIONES

La integración del láser y el desarrollo de la tecnología en estos días, ha permitido la implementación de técnicas metrológicas no invasivas en la industria moderna. A medida que se integran sistemas más robustos y precisos para el desarrollo de nuevos productos, es necesaria la creación de procedimientos para control de calidad más precisos, sistemas para hacer ingeniería inversa, aplicaciones que permitan mejorar tales procesos, que sean lo suficientemente versátiles para lograr una mejoría en tiempos de rediseño y lograr a su vez reducción de costos en el desarrollo de estos productos.

Con el desarrollo de este trabajo se ha mostrado que la digitalización de formas por medio del escaneo con patrones de luz estructurada y la aplicación de técnicas de modelado, se puede crear un sistema que analiza su entorno y que puede ser usado para la creación de modelos virtuales de objetos 3D, permitiendo la implementación de la técnica para propósitos de control de calidad, análisis para realizar ingeniería inversa e implementación en sistemas de desarrollo CAD/CAM/CAE.

Con los errores obtenidos (errores $<0.2\%$ ) se muestra que el sistema provee una forma fiable para digitalizar piezas mecánicas. Ya que éste se calibra por medio de redes neuronales basada en funciones Bézier y procesamiento computacional, su exactitud de medición se ve incrementada ya que se eliminan los errores de medición referentes a los parámetros del sistema de visión y a que el cálculo para el posicionamiento de la línea se realiza a un nivel de sub-píxel, aportando así formas alternativas al diseño.

La repetibilidad obtenida en este trabajo fue de 95.8%, lo cual es una buena cifra para propósitos de reconstrucción y análisis de superficies, debido a que el proceso para la creación del modelo se realiza en pocos pasos, el tiempo para la recuperación de superficie es considerablemente bajo (tiempos menores a 10 min.).

## PRINCIPALES APORTACIONES

El trabajo aquí presentado muestra formas alternativas al diseño CAD/CAM/CAE, donde la principal ventaja se tiene en los pocos pasos para la reconstrucción de superficies objetos.

Ya que en este trabajo se genera el modelo con funciones NURBS, se tiene una representación estándar que facilita el diseño e intercambio de datos geométricos en la industria, donde varios estándares internacionales como IGES, STEP and PHIGS, reconocen a los NURBS como herramientas poderosas para el diseño geométrico.

Con esto se provee un sistema dinámico para el rediseño de objetos, que permiten formas alternativas para el control de calidad y mejoramiento de productos.

Además que este sistema se muestra como una forma viable de implementar la reconstrucción, ya que es un sistema de bajo costo y de construcción sencilla.

## TRABAJO A FUTURO

Debido a que este trabajo es dirigido principalmente al modelado y procesos de ingeniería inversa, es de gran relevancia continuar con:

- El desarrollo de un sistema de visión más robusto que permitan la reconstrucción de objetos más grandes.
- Crear el software necesario para generar un vínculo entre la reconstrucción generada en este trabajo y los programas de análisis y diseño CAD/CAM/CAE.
- Implementar un software que permita la manipulación de las formas, con esto se generaría un sistema más dinámico en el ámbito del diseño.
- Por otro lado también convendría desarrollar este tipo de sistemas para aplicaciones en el campo, es decir miniaturizar este tipo de equipos.

## REFERENCIAS BIBLIOGRAFICAS

- [1]. - Jean Gallier, *Curves and Surfaces in Geometric Modeling*, The Morgan Kaufmann Series in Computer Graphics, 2<sup>nd</sup> edition.
- [2]. - J. Apolinar Muñoz-Rodríguez, *Laser imaging and approximation networks for calibration of three-dimensional vision*, *Optics & Laser Technology* Volume 43, Issue 3, Abril 2011, Pages 491–500.
- [3]. - Milan Sonka, Vaclav Hlavac, Roger Boyle, *Image Processing Analysis and Machine Vision*, Editorial THOMSON, 3RD EDITION, 2008.
- [4]. – Electronic Imaging Resource Guide, Edmund optics, <http://www.edmundoptics.com/technical-resources-center/application-notes>, section imaging.
- [5]. - Gerard Medioni, Sing Bing Kang, *Emerging Topics in computer vision*, University of Southern California, Microsoft Research, November, 2003.
- [6]. - Chris Solomon, Toby Breckon, *Fundamentals of digital image processing*, EDITORIAL WILEY-BLACKWELL, 1<sup>ST</sup> edition, 2011.
- [7] K.S. FU, R.C. Gonzalez and C.S.G. Lee, *Robotics: control, sensing, vision and intelligence*, Mc Graw Hill, 1987.
- [8] Z. Huang, *Finge skeleton extraction using adaptive refining*, *Optics and Laser in Engineering*. Vol 18, 281-295, (1993).
- [9] M. Servin and R. Rodríguez-Vera, *Two-dimensional phase locked loop demodulation of interferograms*, *Journal of Modern Optics* Vol. 40 No. 3, 2087-2094, (1993).
- [10] Ch. Doignon and D. Knittel, *A structured light vision system for out-of-plane vibration frequencies location of a moving web*, *Machine Vision and Applications*, Vol 16 No. 5, 289-297, (2005).
- [11]. - Christopher M. Bishop, *Pattern Recognition and Machine Learning*, editorial Springer, 1st edition (2006).
- [12]. – Les Piegl, Wayne Tiller, *THE NURBS BOOK*, LES PIEGL, EDITORIAL SPRINGER, 2<sup>nd</sup> edition (1996).
- [13]. - Paul Siebert, *An Introduction to 3D computer vision techniques and algorithms*, editorial Wiley, 1<sup>st</sup> edition (2009).

- [14] Jiang Dan, Wang Lancheng, An algorithm of NURBS surface fitting for reverse engineering, International Journal of Advanced Manufacturing Technology, Vol. 31, issue 1-2 ,pp 92-97 , (2006).
- [15]. – Howard Anton, Elementary Linear Algebra, Editorial Anton Textbooks, Inc., 10<sup>th</sup> edition, 2010.
- [16]. - J. Apolinar Muñoz-Rodríguez, Modeling of a mobile setup by networks for object contouring, Opt. Eng.Vol. 47, pag. 053605-1 (may 2008).
- [17]. - MICHAEL E. MORTENSON, GEOMETRIC MODELING, EDITORIAL Wiley Computer Publishing, 1st edition (1997).
- [18]. – Rastogi, Pramod K., Optical Measurement Techniques and Applications, EDITORIAL Artech House, 1<sup>st</sup> edition (1997).
- [19]. - Evaluación de datos de medición, Guía para la expresión de la incertidumbre de medida, Centro español de Metrología, 1ra Edición, 2008.
- [20]. – Luis Enrique Llamosa, Estudio de la repetibilidad y reproducibilidad utilizando el método de promedios y rangos para el aseguramiento de la calidad de los resultados de calibración de acuerdo con la norma técnica ntciso/iec, Scientia et Technica Año XIII, No 35, Agosto de 2007.
- [21] W. Ma, J. P. Kruth, NURBS Curve and Surface Fitting for Reverse Engineering, International Journal of Advanced Manufacturing Technology, Vol. 14, pp 918-927 , (1998).
- [22].-J. Apolinar Muñoz-Rodríguez, Anand Asundi, Ramon Rodríguez-Vera, Shape detection of moving objects based on a neural network of a light line, Optics Communications, Volume 221, Issues 1–3, 1 June 2003, Pages 73–86
- [23]. - J. Apolinar Muñoz-Rodríguez, A. Asundi, Ramon Rodríguez-Vera, G. Garnica Campos , Shape detection using light line and Bezier approximation network, Imaging Science Journal, Volume 55, Number 1, March 2007 , pp. 29-39(11).
- [24]. - Qiu Guan, Sheng Yong Chen, Wanliang Wang, Y.F. Li, Grid- Pattern Design for Fast Scene Reconstruction by a 3D Vision Sensor, Optical Measurement for industrial inspection IV, SPIE Volume 5856, pag. 201-209 (2005).
- [25]. - Qifeng Yu, Xiaohua Ding, Hongjun Oyang, A Calibration Method of Structured Light System With a Planar Object, SPIE Volume 5144, pag. 420-430 (May 2003).
- [26]. - Georg H. Notni, Gunter Notni, Digital Fringe Projection in 3D Shape Measurement-an Error Analysis, SPIE Volume 5144, pag. 372-380 (May 2003).

- [27]. - Soung R. Yee, Paul M. Griffin, Three-dimensional imaging system, *Optical Engineering* 33(6), 2070-2075 (June 1994).
- [28]. - Zhan Song, Ronald Chung, Grid point extraction for structured light system, *SPIE, Volume 50(9), Optical engineering* (September 2011).
- [29]. - Yoshinori Hatanaka, Yasushi Kobayashi, Hidetoshi Takahashi, A 3-D Measurement System using a CCD Sensor and a Laser Scanner, *SPIE Volume 2183*, pag. 250-256 (March 1994).
- [30]. - Song Zhang, *Handbook of 3D Machine Vision (optical Metrology and Imaging)*, editorial CRC Press, 1<sup>st</sup> edition 2013.
- [31]. - L.A. Piegl, W. Tiller, Parametrization for surface fitting in reverse engineering, *Elsevier Science Direct, Computer aided design* 33, pag. 593-603, (august 2000) .
- [32]. - Wolfgang Heidrich, Richard Bartels, George labahn, *Fitting Uncertain Data with NURBS*, Vanderbilt University Press, Nashville, TN, pp 1-8, 1996.
- [33]. - Anke Guenther, Karin Kniel, Frank Hartig, Ingo Lindner, Introduction of a new bevel gear measurement standard, *Elsevier Science Direct, Manufacturing Technology* 62, Pag. 515-518 (2013).
- [34]. - A. Guenther, Evaluation of Runout at Bevel Gears based on Pitch Measurements, *Annals of the CIRP, Volume 55/1/2006*.
- [35]. - A. Guenther, Interpretation of bevel gear topography measurements, *Elsevier Science Direct , CIRP Annuals- Manufacturing technology* 60(2011), pag. 551-554.
- [36]. - Yohan Kodo, Sonko Osawa, Osamu Sato, Masaharu Komori, Toshiyuki Takatsuji, Evaluation of instruments for pitch measurement using a sphere artifact, *Elsevier Science Direct, Precision Engineering* 36(may 2012), pag. 604-611.
- [37]. - Paulo Cauchick-Miguel, Tim King, Jim Davis, *CMM verification: a survey*, Elsevier Science Direct, *Measurement Vol. 17*, pp. 1-16 (1996).
- [38]. - C. Baudouin, R.Blgot, S. Leleu, N. Muller, P. Martin, *Gear geometric control software: Approach by entities*, hal-00750724, version1 -12 nov 2012.
- [39]. - Naser Amini, B.G. Rosén, H. Westberg, *Optimization of Gear Tooth Surfaces*, Elsevier Science Direct, *Tools Manufacture*, vol. 38, pag. 425-435 (1998).
- [40]. - C.H. Gao, K. Cheng, D. Webb, *Investigation on sampling size optimization in gear tooth surface measurement using a CMM*, Springer-Verlag London Limited, *manuf Technol* 24(2004), pag. 599-606.



- [41]. - Gary Sroka, Lane Winkelman, Superfinishing Gears- The State of the Art, Gear technology, December 2003, .
- [42]. - Pajares Gonzalo, Visión por Computador: Imágenes Digitales y Aplicaciones, Editorial ALFAOMEGA, 1st edition (2002).
- [43]. - Kjell j. Gasvik, Optical Metrology, EDITORIAL John Wiley & Sons, Ltd., 3<sup>RD</sup> EDITION 2002.
- [44]. - Nils Carlson, NURBS Surface Fitting with Gauss-Newton, Lulea University of Technology, Computer Science and Engineering Department of Mathematics, 2009.