

SISTEMA PARA LA RECONSTRUCCIÓN GRÁFICA TRIDIMENSIONAL DE  
OBJETOS FÍSICOS EN COMPUTADOR.

HUGO FRANCO TRIANA

ADOLFO REYES CUEVAS

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

FRANCISCO MELUK

Ingeniero Electrónico, M. Sc.

UNIVERSIDAD NACIONAL DE COLOMBIA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE SISTEMAS

SANTAFÉ DE BOGOTÁ

2000

*A todos aquellos que dieron sus vidas para defender la  
existencia de la educación pública...*

Hugo Franco

*A María Rosaura, mi madre.*

Adolfo Reyes

## **AGRADECIMIENTOS**

Los autores expresan sus agradecimientos a:

El ingeniero Francisco Meluk, por hacer de la búsqueda del conocimiento una labor agradable y emocionante.

El Licenciado Víctor Hugo Franco, por su valioso aporte en la elaboración de la herramienta de ubicación de coordenadas.

Los ingenieros Manuel Guillermo Forero y Fabio González, por su interés y colaboración en la solución de diversas cuestiones durante el desarrollo del proyecto.

Y a nuestros padres por su apoyo incondicional, sin el cual este proyecto no habría sido posible.

## CONTENIDO

### OBJETIVOS.

### INTRODUCCIÓN.

<b>1. NOCIONES PRELIMINARES</b>	<b>1</b>
1.1. NOCIONES GEOMÉTRICAS ELEMENTALES	1
1.2. ELECTRÓNICA APLICADA A LA DIGITALIZACIÓN DE SEÑALES	5
1.3. COMUNICACIÓN CON EL COMPUTADOR	6
<b>2. UBICACIÓN DE COORDENADAS MEDIANTE ARTICULACIONES</b>	<b>12</b>
2.1. PROBLEMA CINEMÁTICO DIRECTO	13
2.1.1. Modelo Geométrico para la Ubicación de Puntos en el Espacio	14
2.1.2. Matrices de Transformación Geométrica	18
2.1.2.1. Matrices de Transformación Bidimensionales	18
2.1.2.2. Matrices de Transformación Tridimensionales	19
2.2. CONFIGURACIÓN FÍSICA DE UNA HERRAMIENTA DE UBICACIÓN DE COORDENADAS MEDIANTE ARTICULACIONES	22
2.2.1. Espacio de Trabajo	23
2.2.2. Grados de Libertad	23
<b>3. GENERACIÓN DE ENMALLADOS</b>	<b>24</b>
3.1. CONCEPTOS BÁSICOS DE GEOMETRÍA COMPUTACIONAL	25
3.1.1. Búsqueda Geométrica	25
3.1.1.1. Localización de Puntos	27
3.1.1.2. Búsqueda de Rango	28
3.1.2. Simplexes y Politopos	30
3.1.3. Envolturas Convexas	31
3.1.4. Relaciones de Vecindad	32
3.1.5. Enmallados	33
3.1.6. Diagrama de Voronoi – Triangulación de Delaunay	35.
3.2. LA TRIANGULACIÓN DE DELAUNAY	38
3.2.1. Lema General de Delaunay	40
3.2.2. Algoritmos para la Construcción de la Triangulación de Delaunay	41

3.2.2.1. Fuerza Bruta	42
3.2.2.2. Algoritmo Incremental	42
3.2.2.3. Otros Métodos	43
3.2.3. Triangulaciones Restringidas	44
<b>4. MODELAMIENTO GRÁFICO TRIDIMENSIONAL</b>	<b>46</b>
4.1. TRAZO EN COMPUTADOR DE PRIMITIVAS GEOMÉTRICAS	47
4.1.1. Discretización de primitivas geométricas	47
4.1.2. Eliminación de artefactos de discretización	50
4.1.3. Relleno de polígonos	51
4.2. MODELAMIENTO DE LA GEOMETRÍA TRIDIMENSIONAL	52
4.2.1. Transformaciones Geométricas Aplicadas al Modelamiento Tridimensional	52
4.2.1.1. Translación	53
4.2.1.2. Rotación	53
4.2.1.3. Escalamiento	54
4.2.2. Coordenadas Homogéneas	54
4.2.3. Proyecciones	56
4.2.4. Recorte de Polígonos	58
4.3. DESPLIEGUE EN PANTALLA	59
4.3.1. Determinación de superficies visibles	59
4.3.1.1. Métodos Objeto–Espacio	59
4.3.1.2. Métodos Imagen–Espacio	60
4.3.1.3. Métodos Mixtos	62
4.3.2. Modelos de Color	63
4.3.2.1. Modelo RGB	63
4.3.2.2. Modelo CYMK	63
4.3.2.3. Modelo YIQ	64
4.3.2.4. Modelo HSB	64
4.3.3. Modelos de Iluminación	64
4.3.3.1. Componente Ambiental	65
4.3.3.2. Componente Difuso	65
4.3.3.3. Componente Especular, Modelo de Phong	66
4.3.4. Sombreado	67
4.3.4.1. Sombreado de Gouraud	68
4.3.4.2. Sombreado de Phong	69
4.3.5. Detalle de Superficie	69
4.3.5.1. Correspondencia de Patrones	70
4.3.5.2. Correspondencia de Protuberancias	70

4.3.6.	Transparencia	71
4.3.6.1.	Transparencia de malla	72
4.3.6.2.	Transparencia interpolada	72
4.3.6.3.	Transparencia filtrada	73
4.3.6.4.	Transparencia Refractiva vs. Transparencia No–Refractiva	73
<b>5.</b>	<b>IMPLEMENTACIÓN DEL SISTEMA</b>	<b>75</b>
5.1.	REGISTRO DE POSICIONES TRIDIMENSIONALES.	75
5.1.1.	Configuración de la herramienta de ubicación de coordenadas	76
5.1.1.1.	Grados de libertad y número de enlaces	76
5.1.1.2.	Plano de rotación de las articulaciones	76
5.1.2.	Dispositivo para la detección del desplazamiento angular	77
5.1.2.1.	Sensor óptico de rotación	77
5.1.2.1.1.	Descripción	78
5.1.2.1.2.	Canales de salida	79
5.1.2.1.3.	Funcionamiento	80
5.1.2.2.	Transformación de las señales	81
5.1.2.2.1.	Etapa de aislamiento	81
5.1.2.2.2.	Etapa de acoplamiento	82
5.1.3.	Registro de posiciones angulares	83
5.1.3.1.	El Microcontrolador	84
5.1.3.1.1.	Puertos	84
5.1.3.1.2.	Interrupciones	85
5.1.3.1.3.	Registros Asociados	86
5.1.3.1.4.	Código Assembler Manejado por el Microcontrolador	86
5.1.3.1.5.	El Compilador MPC de ByteCraft	88
5.1.3.2.	Algoritmo para el registro de rotaciones en el PIC	90
5.1.3.2.1.	Variaciones de los contadores	90
5.1.3.2.2.	Validaciones de sumas y restas	91
5.1.3.2.3.	Cadenas de estados	91
5.1.3.2.4.	Diagrama de flujo	92
5.1.4.	Comunicación con el computador	94
5.1.4.1.	La interfaz serie	95
5.1.4.2.	Interfaz RS–232	96
5.1.4.2.1.	Especificación Eléctrica	96
5.1.4.2.2.	Especificación Mecánica	96
5.1.4.2.3.	Especificación Funcional	96
5.1.4.2.4.	Circuitos principales	97

5.1.4.3. Tipos de puerto serie en un computador	99
<b>5.2. SOFTWARE PARA EL MODELAMIENTO Y REPRESENTACIÓN GRÁFICA DE OBJETOS TRIDIMENSIONALES</b>	<b>101</b>
5.2.1. Descripción del sistema	101
5.2.1.1. Entradas y salidas	101
5.2.1.2. Módulos principales de la aplicación	105
5.2.1.2.1. Solución al problema cinemático directo para la configuración del hardware	105
5.2.1.2.2. Generación de enmallados poligonales	108
5.2.1.2.3. Despliegue en pantalla del modelo tridimensional	108
5.2.2. Aspectos generales de la implementación	111
<b>6. RESULTADOS Y CONCLUSIONES</b>	<b>112</b>
6.1. DESEMPEÑO DEL HARDWARE	112
6.1.1. Maniobrabilidad de la herramienta de ubicación de coordenadas	112
6.1.2. Lectura de datos	113
6.1.3. Resolución	113
6.2. DESEMPEÑO DEL SOFTWARE	114
6.2.1. Desempeño del proceso de triangulación	114
6.2.2. Desempeño del proceso de despliegue	114
6.3. EJEMPLOS	115
<b>7. BIBLIOGRAFÍA.</b>	

## **OBJETIVOS**

### **OBJETIVO PRINCIPAL**

Desarrollo de un prototipo de sistema que permita el registro de puntos sobre la superficie de un objeto físico y obtener a partir ellos una representación gráfica tridimensional de ese objeto en computador mediante técnicas de generación de mallas poligonales y *renderizado*.

### **OBJETIVOS ESPECÍFICOS**

- Desarrollo de un prototipo de herramienta en hardware que permita la ubicación de puntos específicos sobre la superficie de un objeto dado.
- Desarrollo de un sistema (hardware y software) que, por medio del uso de sensores, registre digitalmente las coordenadas espaciales de los puntos seleccionados, y las envíe a un computador para su posterior procesamiento.
- Implementación del software necesario para la generación de la representación gráfica tridimensional del objeto, a través del uso de técnicas de generación de mallas poligonales y *renderizado*.

## INTRODUCCIÓN

Este trabajo, como proyecto final de la Carrera de Ingeniería de Sistemas, ha pretendido rescatar el carácter eminentemente práctico de la ingeniería en diferentes ámbitos de las ciencias de la computación, como la graficación por computador y la geometría computacional. Además, se hizo necesario recurrir a otras áreas del conocimiento, como la electrónica digital o el diseño industrial, aunque con un cierto grado de informalidad.

El principal propósito del presente proyecto es el de explorar, en sus múltiples facetas, la posibilidad de desarrollo de una herramienta tecnológica que permita la reconstrucción gráfica en formato tridimensional de objetos reales, utilizando un modelo geométrico obtenido mediante su digitalización. La digitalización de imágenes y la digitalización de objetos tridimensionales son radicalmente diferentes. La digitalización de imágenes consiste en obtener un muestreo de la imagen a través de un barrido sobre ésta con celdas fotosensibles, obteniéndose matriz de colores o mapa de bits. Entre tanto, la digitalización de objetos tridimensionales se fundamenta en la obtención de una representación geométrica del objeto haciendo uso de un muestreo de puntos sobre su superficie, cuyas coordenadas son obtenidas mediante algún proceso de registro electrónico.

La digitalización tridimensional de objetos reales es una labor compleja, tanto por los problemas que pueden presentarse en la obtención de la morfología del objeto a digitalizar como por las dificultades que se pueden presentar en el proceso que hace uso de esa información para la generación de un modelo de datos que represente dicha morfología para su manipulación y presentación en el computador. La elaboración de sistemas destinados a la digitalización de objetos tridimensionales se remonta aproximadamente quince años atrás, y existen diversos enfoques tecnológicos para su implementación.

Algunos utilizan un sistema de coordenadas cartesianas generado por tres bobinas a manera de antenas receptoras dispuestas ortogonalmente; el sensor es una bobina emisora conectada a un sensor táctil. Así, cuando el sensor detecta un contacto con la superficie del objeto, una señal es emitida por la bobina emisora, siendo recibida por las tres antenas con una intensidad que depende de la distancia entre la bobina emisora y cada una de las receptoras. Usando métodos de interpolación, que dependen del comportamiento de las bobinas, la intensidad de la señal en cada una de ellas es convertida en una coordenada rectangular en tres dimensiones. Herramientas más recientes utilizan una filosofía similar, pero reemplazando el sensor táctil por un sistema de emisión y recepción de un rayo láser, el cual determina la distancia entre la superficie del objeto y el sensor midiendo el tiempo que emplea el rayo en ser emitido y regresar. Una opción diferente es usada con cierta frecuencia en la actualidad por ser más económica y, en cierta medida, reducir las posibilidades de error en las mediciones debido a perturbaciones externas producidas por equipos eléctricos o electrónicos. Esta es la implementación una herramienta articulada no-autónoma<sup>1</sup> cuya morfología la asemeja a un brazo de robot: un conjunto de enlaces rígidos unidos de manera secuencial mediante articulaciones que giran sobre uno o dos planos. En el extremo final del último enlace se ubica un sensor táctil, cuya señal indica que su posición en el momento del pulso corresponde a un punto sobre la superficie del objeto. Las coordenadas de cada punto registrado se obtienen a partir de la aplicación de una serie de operaciones geométricas partiendo de la longitud de cada enlace y el ángulo que forma con el enlace anterior en la configuración que permitió al extremo final llegar a su posición actual. Esta última fue la aproximación escogida para la elaboración del prototipo de herramienta de ubicación de coordenadas que hace parte del proyecto, dada su factibilidad desde el punto de vista económico y tecnológico.

Una vez el hardware ha generado un muestreo de lugares sobre la superficie del objeto, es necesaria la generación de una representación de su geometría por medio de alguna estructura de datos que pueda ser utilizada posteriormente para la reconstrucción gráfica del objeto en pantalla. Diferentes esquemas para la representación computacional de superficies en tres dimensiones han sido planteados, dependiendo de los requerimientos de

---

<sup>1</sup> Es decir, que requiere para su movimiento de la intervención de un operario humano.

la aplicación en los que son aplicados. Existen representaciones con un elevado grado de exactitud basadas en funciones paramétricas en el espacio tridimensional; ellas requieren de muy poca memoria para su almacenamiento, y los resultados obtenidos para en el modelamiento de superficies curvas es de gran calidad, pero la complejidad de los algoritmos de despliegue de superficies paramétricas los hace poco atractivos cuando la velocidad de presentación en pantalla es un factor determinante. En estos casos es preferible hacer una aproximación más burda de la superficie, sí bien más eficiente en función de la velocidad del proceso de despliegue. Ésta consiste en la generación de lista de polígonos en tres dimensiones, cuyos vértices corresponden a los puntos registrados por la herramienta de ubicación de coordenadas. Procesos para la generación automática de conjuntos de polígonos son estudiados en el contexto de la geometría computacional, aunque su aplicación al modelamiento tridimensional está en sus primeras etapas de estudio. Sin embargo, algoritmos para la construcción de triangulaciones son útiles cuando el problema es restringido apropiadamente. El software que acompaña este trabajo utiliza un tipo de triangulación especial, denominado Triangulación de Delaunay, para relacionar los puntos registrados en el proceso de muestreo, obteniéndose una aproximación de la superficie representada por un enmallado triangular de levantamiento, cuyos detalles serán explicados en capítulos posteriores. Otra forma de tratar con este problema, que escapa al alcance del presente trabajo, busca la generación del espacio que ocupa el objeto, usando el muestreo para la creación de primitivas tridimensionales (esferas por ejemplo), y, a partir de la superficie de las esferas aproximar la superficie del objeto mediante un proceso de suavizado iterativo.

La representación geométrica seleccionada es utilizada por las aplicaciones en el computador que utilizan el objeto tridimensional para sus cálculos. Dichas aplicaciones pueden ir desde la simulación de procesos físicos complejos, como en el análisis del comportamiento aerodinámico del objeto, hasta el establecimiento de las características geométricas de piezas mecánicas complejas para su posterior producción en serie. La utilización más sencilla del modelo obtenido (y también la más común en la actualidad) consiste en emplear el objeto digitalizado para su visualización, como sucede con los modelos obtenidos a partir de imágenes médicas, o para su presentación en composiciones artísticas o secuencias animadas. El proceso que conduce a la visualización de un objeto

tridimensional involucra una serie de procedimientos, desde la manipulación de la representación geométrica en el espacio tridimensional, hasta la implementación de algoritmos que emulen el comportamiento físico del objeto en su interacción con la luz, garantizando un resultado creíble.

Tanto la graficación por computador como la geometría computacional se encuentran entre las áreas de la informática de más reciente y vertiginoso desarrollo, gracias a los avances de algunos campos que son soporte tecnológico de la informática. Indiscutiblemente, el principal aspecto es este desarrollo, que ha dado paso al trabajo práctico en dichas áreas, es el sensible incremento en la velocidad de procesamiento y, en el caso de la computación gráfica, la evolución del hardware de vídeo, aunque aspectos teóricos como el análisis de algoritmos han hecho un valioso aporte en cuanto a la optimización de los procedimientos asociados a sus tareas fundamentales.

El impacto de los adelantos tecnológicos citados es evidente cuando se hace una retrospectiva del desarrollo histórico de estas áreas. Pocas décadas atrás, la graficación por computador y la geometría computacional apenas se estaban consolidando como áreas de trabajo estructuradas, y su delimitación, alcance y objeto de estudio estaban aún por ser definidos de manera estricta y con claridad. Lo más notable es que sus fundamentos teóricos, heredados de las matemáticas, la geometría clásica y la física, estaban prácticamente perfeccionados casi un siglo atrás.

En el caso de la Geometría Computacional, los lemas y postulados geométricos fundamentales que le dieron origen, e incluso algunos aspectos algorítmicos, fueron establecidos por investigadores como Dirchlett, Voronoï, Lemoine, Hilbert o Delaunay. Dichos lemas y postulados no serían retomados hasta los primeros años de la década de 1970, siendo utilizados como parte de las investigaciones en análisis de algoritmos. La graficación por computador por su parte ha recorrido su propio camino. Los primeros computadores que implementaban un entorno gráfico sólo se construyeron a finales de la década de 1960 y principios los años 70. Para aplicar las técnicas de computación gráfica se requerían grandes super-computadores pertenecientes a las agencias gubernamentales norteamericanas y europeas dedicadas a la investigación en cuestiones militares, astronómicas o meteorológicas, así como algunas grandes empresas dedicadas a la

exploración petrolera. Los estudios sobre la presentación de gráficos se concentraban en la discretización y el trazo de primitivas geométricas, mientras que las técnicas para el despliegue tridimensional eran dedicadas principalmente a la visualización de funciones de dos variables. Se hacía uso de pantallas de televisión convencionales, tal como los primeros computadores personales, que empezaron a ser elaborados a mediados de la década de 1970. Pero la limitada capacidad de cómputo de estos últimos impedía la implementación de aplicaciones de entorno gráfico, por lo que la interfaz generalmente se elaboraba a partir de un patrón de filas y columnas ocupadas por caracteres. Solo hasta mediados de los años 80 que se popularizaron los computadores personales con posibilidades de ejecutar un entorno completamente gráfico. Mientras tanto, el desarrollo de aplicaciones de diseño asistido por computador en la gran industria avanzaba a pasos enormes, acelerando el desarrollo de nuevas técnicas para la presentación en pantalla de modelos y fenómenos. En los últimos años, y gracias a los avances tecnológicos mencionados anteriormente, se ha hecho factible la posibilidad de utilizar las técnicas de computación gráfica en aplicaciones desarrolladas para los computadores personales con un desempeño aceptable.

La Geometría Computacional y la Graficación por Computador, a pesar de ser campos independientes, cada uno con su propia evolución y objeto de estudio, convergen en la elaboración de aplicaciones en diversas áreas, además de colaborar mutuamente en casos específicos, como la visualización de estructuras geométricas o el estudio analítico de primitivas geométricas para optimizar su despliegue en pantalla.

Con este proyecto se pretende hacer una primera aproximación, desde una perspectiva académica, a las diferentes técnicas cuya aplicación coordinada converge en la elaboración de un prototipo de sistema para la reconstrucción gráfica tridimensional de objetos físicos, el cual es su principal resultado práctico; no se intenta llegar a la implementación de una herramienta completamente funcional para la digitalización y presentación profesional de objetos tridimensionales complejos, ni la optimización de los procedimientos comprendidos en el camino hacia la obtención del modelo gráfico.

## 1. NOCIONES PRELIMINARES

Para la comprensión global y coherente del problema planteado es necesario recurrir a diversos áreas de las matemáticas, las ciencias de la computación y la tecnología, que comprenden múltiples ideas, enfoques y aproximaciones en diferentes contextos. Para un desarrollo comprensible y articulado de todos los aspectos que se aplican a la solución del problema propuesto es necesario realizar ciertas definiciones y delimitaciones cuyo dominio es indispensable para la comprensión y apreciación de las implicaciones y los resultados del proyecto. Tres tópicos fundamentales sustentan las temáticas tratadas por los capítulos posteriores; dichos tópicos son: el marco geométrico aplicado desde distintas perspectivas por las rutinas y procedimientos del software destinado al modelamiento de objetos tridimensionales (sección 1.1), la electrónica aplicada al registro digital de posiciones (sección 1.2) y la tecnología de las comunicaciones que permiten la interacción entre el computador y el mundo real (sección 1.3).

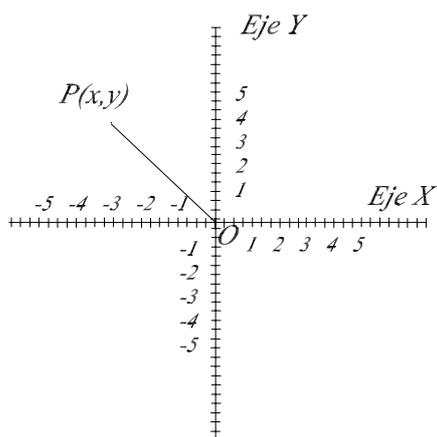
### 1.1. NOCIONES GEOMÉTRICAS ELEMENTALES.

Al hablar del tratamiento de modelos tridimensionales en computador es imposible eludir el tema de la representación numérica de la geometría de los objetos. Para una solución satisfactoria a este problema, es esencial el planteamiento de una abstracción coherente y apropiada de los conceptos básicos inherentes al modelamiento geométrico de los objetos para su procesamiento y almacenamiento en el computador.

En este trabajo se hace uso del enfoque tradicional para el tratamiento computacional de modelos geométricos. Dicho enfoque, utilizado por la gran mayoría de trabajos teóricos y aplicaciones prácticas, consiste en especificar una serie de objetos elementales, denominados *primitivas geométricas*, definidas sobre un *espacio Euclidiano*. El término ‘primitiva geométrica’ envuelve todos aquellos conceptos (puntos, líneas, polígonos,

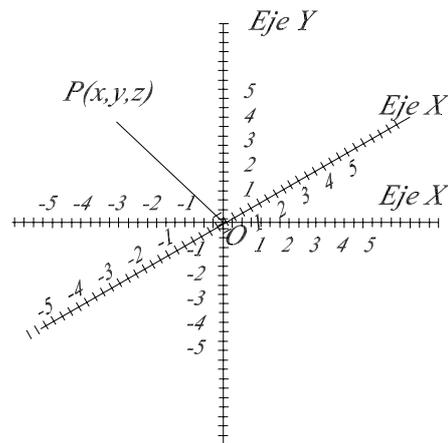
cónicas, curvas y superficies paramétricas, etc.) de relevancia para la ejecución de las tareas propias de cada aplicación.

Es importante recalcar que el espacio Euclidiano es un caso especial de espacio, en el cual la métrica está determinada por el concepto clásico de distancia geométrica entre puntos, planteado por Euclides. La ubicación de lugares en el espacio puede ser obtenida de múltiples formas, utilizando diversos conceptos geométricos. La más importante de todas, por su utilización generalizada y su relativa sencillez teórica, es aquella atribuida a Descartes: un sistema de referencia consistente en  $n$  ejes coordenados ortogonales, siendo  $n$  la dimensión del espacio. Los ejes coinciden exactamente en un punto, llamado origen, y los puntos en el espacio son representados por una  $n$ -tupla de números reales,  $(x_1, x_2, \dots, x_n)$  donde cada  $x_i$  ( $i$ -ésima coordenada) es un número real igual a la distancia de la proyección del punto sobre el  $i$ -ésimo eje hasta el origen.



**Figura 1.1** Sistema de coordenadas cartesianas rectangulares en dos dimensiones

La correspondencia uno a uno entre todas las  $n$ -tuplas posibles de  $\mathbf{R}^n$  y todos los puntos pertenecientes al espacio Euclidiano, generado por  $n$  ejes coordenados ortogonales, se conoce como *sistema de coordenadas cartesianas rectangulares*. Cada eje coordenado se numera positivamente de un lado del origen y negativamente del otro. El lugar de las coordenadas positivas y las negativas se define arbitrariamente, aunque existen convenciones para los casos más conocidos,  $\mathbf{R}^2$  y  $\mathbf{R}^3$  (Figuras 1.1 y 1.2).



**Figura 1.2** Sistema de coordenadas cartesianas tridimensionales.

La noción computacional de arreglo para tipos básicos fue introducida en los primeros lenguajes de alto nivel a partir de este concepto geométrico. Por lo tanto, es apenas natural identificar la representación cartesiana de un punto en  $\mathbf{R}^n$  mediante un arreglo de valores de punto flotante con  $n$  posiciones, aunque enfoques posteriores como el diseño orientado a objetos, pueden conducir a una abstracción igualmente satisfactoria y más poderosa.

El punto es la primitiva geométrica fundamental, pues a partir de ella pueden ser creadas las demás primitivas. Así, por ejemplo, la construcción de una línea tendrá como parámetros los dos puntos que la definen; el plano se obtendrá utilizando tres puntos no colineales; una esfera se puede representar a través del punto en el que se ubica su centro y otro punto cualquiera sobre su superficie; un polígono puede verse como una colección secuencial de puntos coplanares; etc. A su vez, estas primitivas pueden tener como atributos características propias de su modelamiento algebraico clásico, como los coeficientes de la ecuación general o las matrices representativas de su forma paramétrica. Vinculado a los conjuntos de primitivas geométricas que pueden ser establecidos bajo ésta perspectiva se encuentra el concepto de convexidad. Para los propósitos de este trabajo basta con decir que un conjunto convexo definido por alguna serie de parámetros geométricos es convexo sí y sólo sí cualquier par de puntos pertenecientes al conjunto pueden ser unidos por una línea recta completamente contenida por éste.

Entre los primeros estudios sistemáticos de geometría aplicada de forma numérica a disciplinas como la ingeniería, después de muchos cientos de años de oscuridad, se

encuentra el trabajo de Klein, a mediados del siglo XIX, sobre el efecto de ejecutar una serie de operaciones algebraicas sobre objetos geométricos en el espacio Euclidiano. El estudio de Klein condujo al planteamiento del concepto de transformación geométrica, además del establecimiento de sus bases matemáticas, relacionadas principalmente del álgebra lineal. Una transformación geométrica es una operación efectuada sobre uno o varios lugares del espacio para modificar su posición, orientación, o proporción respecto al origen del sistema de coordenadas o a otros objetos. Esto se consigue mediante operaciones matriciales básicas, como sumas vectoriales o multiplicaciones matriciales. Las transformaciones geométricas son utilizadas por una amplia gama de aplicaciones; una de las más conocidas es el proceso de determinación de lugares en el espacio a partir de ciertas configuraciones físicas. Este caso está íntimamente relacionado con áreas como la robótica y sus problemas asociados de determinación de posiciones relativas y ubicación de coordenadas. En el capítulo número dos se hace una revisión más detallada de la aplicación de transformaciones geométricas para la obtención de las coordenadas cartesianas de puntos *muestreados* sobre la superficie de un objeto tridimensional, como parte del proceso de digitalización.

La Geometría Computacional y la Graficación por Computador también hacen uso del modelo numérico de la geometría cartesiana para la implementación de sus propias técnicas y algoritmos, aunque desde perspectivas diferentes. Mientras que la geometría computacional busca la solución de problemas estrictamente geométricos mediante relaciones entre conjuntos de primitivas relacionados a través de variables como distancias y ángulos, la graficación por computador emplea el conjunto de operaciones definidas sobre dichas primitivas para la emulación de los fenómenos ópticos que hacen parte de la percepción visual de los objetos en la naturaleza. En la implementación del sistema propuesto hubo una convergencia de la solución a problemas pertenecientes a estas dos disciplinas. En el capítulo tres hará una descripción general de los principales temas de estudio en el ámbito de la Geometría Computacional, para concluir con una referencia global al enfoque utilizado por el prototipo de aplicación desarrollada para la generación automática de mallas poligonales usando el concepto de Triangulación de Delaunay. El capítulo cuatro expone los procedimientos y técnicas más relevantes de la Graficación por Computador relacionadas con el modelamiento de objetos tridimensionales.

## 1.2. ELECTRÓNICA APLICADA A LA DIGITALIZACIÓN DE SEÑALES.

La incursión de los computadores en las esferas científica y tecnológica ha planteado la necesidad de revisar una serie de enfoques, y plantear otros nuevos, para que la solución analítica de los problemas de la vida real sea compatible con el carácter numérico y digital de las soluciones entregadas por metodología basadas en la computación. En este contexto, el principal reto es el de conseguir una representación de las variables asociadas a un fenómeno dado que el computador pueda recopilar y procesar. La solución a problema contempla la utilización de ciertas técnicas, como el muestreo y la digitalización de los parámetros relevantes de un fenómeno, para su posterior introducción al computador a través de la electrónica. Entre los conceptos electrónicos de aplicabilidad utilidad práctica, se encuentra el amplificador operacional. Este ha sido estudiado por su particular comportamiento, que lo hace útil tanto en problemas de carácter análogo como digital. Los amplificadores operacionales ideales, descritos en la teoría, poseen características como la ganancia infinita de voltaje ( $A_{V \rightarrow \infty}$ ), la impedancia infinita de entrada ( $Z_{ent \rightarrow \infty}$ ), la impedancia de salida igual a cero ( $Z_{sal \rightarrow 0}$ ), el voltaje de salida  $V_{sal} = 0$  cuando los voltajes de entrada son iguales y el ancho de banda infinito (no hay retraso de la señal a través del amplificador). Los amplificadores operacionales reales (*amp-op's*) no pueden llegar a comportarse como los ideales, pues se violarían leyes físicas, como la velocidad máxima de la luz; sin embargo estos pueden lograr una buena aproximación para el desarrollo de las aplicaciones de la vida real. El *amp-op* es un amplificador modular de etapas múltiples en cuya primera etapa se comporta como un amplificador diferencial que proporciona una alta ganancia a entradas diferenciales y baja ganancia a las señales aplicadas simultáneamente a ambas entradas (señales en modo común). El amplificador diferencial presenta una gran impedancia a cualquier señal de entrada que se le aplique. En esta etapa se establece la impedancia de entrada y se minimiza la respuesta en modo común y los voltajes de desajuste. Las siguientes etapas intermedias permiten cambiar a cero el nivel de voltaje estático del punto de operación a la salida, y proporcionar ganancia tanto de voltaje como de corriente para lo cual se requiere una ganancia de voltaje adicional para obtener una alta ganancia general de voltaje, y la ganancia de corriente es necesaria para suministrar corriente impulsora a la etapa de salida, sin cargar la etapa de entrada. La etapa de salida debe presentar una baja impedancia de salida y proporcionar corriente suficiente

para impulsar la carga esperada. Debe tener una impedancia de entrada lo suficientemente alta para no cargar la última etapa de amplificación intermedia. La etapa de salida es normalmente un emisor seguidor ó una configuración complementaria.

En un amplificador operacional se distinguen varios componentes entre los cuales están:

- Los terminales para los voltajes de la fuente de alimentación. (+V, -V).
- Compensación de frecuencia: estas terminales (llamadas a veces de avance, de retraso o de régimen de atenuación) sirven para impedir la oscilación del amplificador.
- Entrada no inversora: si la entrada está puesta a tierra y se aplica una señal a la entrada no inversora, la salida estará en fase con la señal de entrada.
- Entrada inversora: si la entrada no inversora está puesta a tierra y se aplica una señal a la entrada inversora, la salida estará 180° fuera de fase con respecto a la señal de entrada.
- Salida: donde aparece el voltaje amplificador.

Básicamente el amplificador operacional trabaja en tres regiones: Región de voltaje de saturación positiva, región lineal y región de saturación de voltaje negativo. Los voltajes de saturación tanto positivo como negativo corresponden a los voltajes aplicados por los terminales desde la fuente de alimentación, mientras que en la región lineal el voltaje es un valor intermedio entre los voltajes de saturación:  $V_0 = A_0(V^+ - V^-)$

El amplificador operacional utilizado puede ser utilizado como un comparador cuando se utiliza en lazo abierto, haciendo que voltajes menores de un voltaje de referencia se convierten en 0 y voltajes superiores se convierten en el voltaje de saturación del amplificador. De esta forma se tiende un puente entre la realidad continua y la representación computacional digitalizada.

### **1.3. COMUNICACIÓN CON EL COMPUTADOR.**

En el proceso de control que el sistema operativo realiza sobre los dispositivos externos e internos se establecen comunicaciones, por ejemplo entre la unidad central de proceso y los dispositivos periféricos como impresoras, escáner, teclados, ratón, etc. Estas

comunicaciones permiten el flujo de información no solo para controlar, sino para intercambio de información cuando se necesita trabajar con datos externos al computador como lo hacen los sistemas de control.

La información para este efecto se estructura en bytes. Si la distancia entre los computadores que se pretenden comunicar es pequeña (*menos de 15 metros*), se puede utilizar una transmisión de datos de forma paralela en donde se utilizan 8 hilos para transmitir por cada uno de estos un bit, logrando velocidades de transmisión muy altas. En la red telefónica y en la mayoría de sistemas de larga distancia se utiliza otro tipo de comunicación llamada comunicación serial, que básicamente consiste en transmitir todos los bits por un solo canal y por medio de un dispositivo llamado UART (*Universal Asynchronous Receiver Transmitter*) ó Transmisor Receptor Asíncrono Universal que convierte cada byte en una serie de ocho bits y viceversa. Por facilidad de interface entre el computador y el microcontrolador se utiliza la comunicación serial.

Para el caso de las comunicaciones tanto serial como paralela, el computador tiene conectores especiales llamados puertos. Los puertos paralelos (*Parallel Port*) se identifican por *LPT1* y *LPT2*. Estos puertos son conectores hembras de 25 pines por donde generalmente se conecta la impresora. Los puertos serie (*Serial Port*) son máximo cuatro y se identifican como *COM1*, *COM2*, *COM3*, *COM4*. Estos pueden tener 9 ó 25 pines.

Dependiendo de la longitud del cable, la información que viaja a través de él puede sufrir alteraciones debido a fenómenos relacionados con la constitución del medio de transmisión (Inductancia, resistencia, etc.). La distancia a partir de la cual se comienza a generar errores en la transmisión de los datos es de 15 metros. Una de las soluciones propuestas para esto (compatible con la red telefónica existente) es acoplar las señales de salida de los puertos del computador al tipo de señal para los cuales fue construida inicialmente la red telefónica, esto es, señales analógicas. El dispositivo que permite realizar estas conversiones se denomina módem. El proceso de transformación de señales digitales a analógicas se llama modulación y el proceso contrario demodulación.

Existen tres formas de realizar una comunicación:

**Simplex.** También conocida como unidireccional, en esta la transmisión de información se realiza en un solo sentido. El más claro de los ejemplos es la TV; la comunicación se realiza del estudio de grabación a los hogares siempre en el mismo sentido.

**Semidúplex (Half-Dúplex).** La transmisión de información en este caso se realiza en ambos sentidos pero no simultáneamente. Es una comunicación bidireccional donde no hay cruce de información en la línea. Esta es la forma más común para transmisión de datos y se realiza incluso sobre circuitos que permiten el modo duplex.

**Dúplex (Full-Dúplex).** La comunicación se puede realizar bidireccionalmente y simultáneamente. El más claro ejemplo de este tipo de comunicación es la comunicación telefónica en donde las dos personas que se comunican pueden hablar en cualquier momento, incluso simultáneamente. Por esto, una comunicación debe contar con procedimientos que nos permitan identificar los bits, los caracteres y las tramas; esta técnica se conoce con el nombre de sincronismo existiendo el sincronismo de bit, sincronismo de carácter y sincronismo de trama. El sincronismo de bit es responsabilidad del módem, mientras que el sincronismo de carácter y de trama es responsabilidad del protocolo de comunicaciones utilizado.

Existen dos métodos para realizar el sincronismo de bit: el método asíncrono el método sincrónico. El método de transmisión asincrónica (también llamado de arranque y espera ó start-stop) fue establecido para los terminales TTY (Teleimpresoras) usados en los años cincuenta y sesenta. Este sistema utiliza un bit de arranque (start bit), bit 0, el cual ponía en funcionamiento a los dos motores simultáneamente. A continuación se transmiten los bits de información (5, código Baudot), luego de éste se transmitía un bit de parada (stop bit), bit 1, para a continuación volver a transmitir el bit de arranque del nuevo carácter. Actualmente no se utiliza código Baudot sino ASCII.

En una transmisión asincrónica se tiene por tanto, que por cada carácter transmitido se tiene un bit de arranque, seguido por los ocho bits de información y un bit de parada, en total 10 bits. De esta forma se resuelve el problema de la sincronización de bit y de carácter pero, se presenta el inconveniente de transmitir por cada carácter enviado, 2 ó 3 bits de

control. Esto es, el sistema asincrónico desperdicia entre el 23.8% y el 30% del tiempo en enviar caracteres de control de sincronismo de bit.

La búsqueda de mayores velocidades de transmisión llevó a los diseñadores de sistemas de comunicaciones de datos a idear un sistema que aprovecharse mejor el tiempo de lo que lo hace el sistema asíncrono. La idea es producir dispositivos que envíen el máximo posible de bits por unidad de tiempo utilizando un mismo canal de comunicación. El resultado fue el sistema sincrónico.

El problema que se pretende resolver es cómo hacer que la base de tiempos usada por el terminal receptor para fijar la cadencia con la que se deben leer los datos, sea lo suficientemente parecida a la base de tiempos del terminal emisor como para que no se produzcan errores de lectura. Al utilizar los terminales de origen y destino distintas bases de tiempo, distintas señales de reloj, y dado que los circuitos electrónicos no son perfectos, una pequeñísima desviación puede producir, como se ha indicado anteriormente un error en la lectura de los datos.

Para asegurar la sincronización, antes de empezar a transmitir los datos de información, el terminal emisor transmite uno ó más caracteres de sincronización llamados *SYN*. Esos caracteres están formados por una combinación de 0 y 1 alternos (01010101). Por otro lado, los bytes de información son enviados agrupados en tramas de forma que después de cada trama se envía de nuevo el carácter *SYN*.

La ventaja de estos sistemas es que no desperdician tiempo en hacer el sincronismo. Los módems sincrónicos son equipos más complejos que los asincrónicos pero supone una mejor utilización de la línea y permite mayores velocidades, por ser menos sensible al ruido e imperfecciones de los medios de transmisión.

Existe una gran variedad de sistemas de modulación, cada una pensando en una circunstancia distinta y con propósitos distintos. Las técnicas básicas de modulación están basadas en el hecho que a través de los medios de comunicación, como los canales telefónicos, se pueden transmitir de una manera fiable señales analógicas que señales digitales. Por este motivo, cuando se va a transmitir una información digital, lo que transmitimos es una señal analógica, llamada portadora, a la cual se le modifica una de sus características de acuerdo con la información binaria que se pretende transmitir. La señal

portadora (*carrier*) es normalmente una onda senoidal, la cual está definida por tres características: Frecuencia, amplitud máxima, y fase. Eso quiere decir que si conocemos estas características, en cualquier momento podemos saber el estado de la señal aplicando una sencilla formula matemática:

$$a = A \text{sen}(2\pi ft + \theta)$$

donde  $a$  es el valor instantaneo de la tensión en el tiempo  $t$ ,  $A$  es amplitud minima,  $f$  es la frecuencia y  $\theta$  es la fase.

Si transmitimos la señal portadora sin ninguna modificación, estaremos transmitiendo una señal senoidal constante, la cual no transporta ninguna información. Ahora bien, podemos transmitir una señal de frecuencia y fase constante, pero enviando dos amplitudes distintas, una para representar informaciones 0 y otra para representar informaciones 1. De la misma forma se puede enviar una señal de amplitud y fase constantes pero con dos frecuencias y, una señal de amplitud y frecuencia constante pero con distinta fase. Esos simples procesos son lo que se conocen con el nombre de modulación. Dicho de otra forma, modular quiere decir modificar las características de una señal llamada portadora, de forma que contenga la información que se pueda transmitir.

Los tres sistemas básicos de modulación son:

- Modulación de amplitud
- Modulación de frecuencia
- Modulación de fase

La lógica TTL es utilizada internamente en el computador para representar los estados binarios por medio de voltajes, donde un cero binario (0) es 0 volts y un uno (1) es representado por 5 volts. Para efectos de transmisión hacia el exterior la UART trabaja 0 y 5 como 3 y -3 volts. Es así como un voltaje inferior a -3 volts equivale a un uno (1) y, uno superior a tres volts equivale a un cero (0). Normalmente estos voltajes se encuentran en los rangos de [-12, -3] y [3,12].

Como se mencionó anteriormente la comunicación serial es empleada para comunicaciones a larga distancia empleando por ejemplo la red telefónica, mientras que la comunicación

paralela se utiliza en distancias cortas. Nótese que para la comunicación serial se deben tener en cuenta diferentes aspectos en la coordinación de la transmisión y la recepción de los datos. Entre estos están:

Los datos transmitidos desde el emisor son cadenas de bits que el receptor debe leer de tal forma que no repita el mismo bit ni tampoco pierda alguno, es decir; el envío y la recepción de las cadenas deben conservar ciertas características de sincronismo.

En la transmisión paralela los bits son transmitidos de ocho en ocho y se diferencia perfectamente un byte de otro. En la transmisión serie se debe diferenciar cada byte de información ya que estos se transmiten como secuencias de bits.

Por último, cuando se tiene que transmitir grandes volúmenes de información, ésta no se transmite toda de una vez, ya que eso provocaría que de haber un error se tendría que retransmitir todo desde el principio. Para evitar eso, la transmisión no se transmite de un solo golpe, sino que se divide en secciones más pequeñas llamadas tramas, bloques ó paquetes. Esto quiere decir que hay que establecer un procedimiento que permita identificar que carácter, de todos los recibidos, es el primero de cada trama.

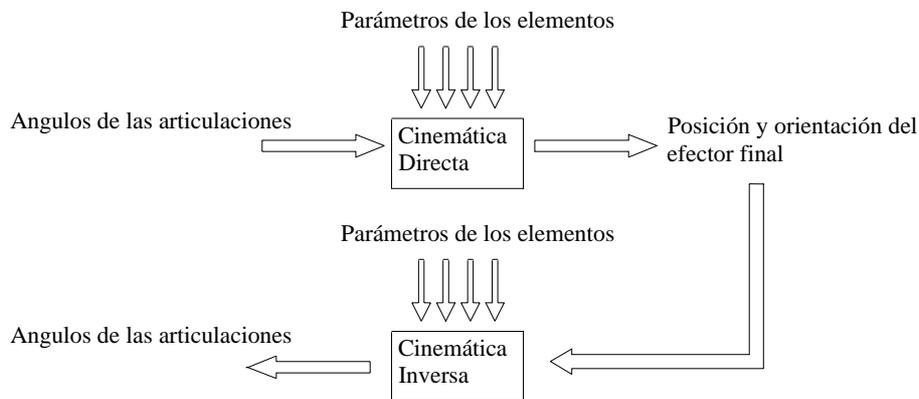
## 2. UBICACIÓN DE COORDENADAS MEDIANTE ARTICULACIONES

Dentro de la gran variedad de alternativas para la ubicación de puntos en el espacio utilizando medios físicos, una de las opciones más accesibles es, sin duda, la ubicación de coordenadas mediante articulaciones. El problema de ubicación de coordenadas puede definirse rápidamente de la siguiente manera: Dado un conjunto enlaces rectos unidos por sus extremos mediante articulaciones, encontrar las coordenadas cartesianas del extremo final del último enlace conociendo el ángulo que forma cada enlace con su antecesor y el ángulo que forma el primer enlace con el origen del sistema de coordenadas.

La ubicación de coordenadas mediante articulaciones podría ser considerada, de alguna manera, el problema inverso de uno de los principales objetivos de la robótica: dado un conjunto de enlaces como el anterior y un punto en el espacio, determinar el ángulo que debe formar cada enlace con su antecesor, y el primer enlace con el origen del sistema de coordenadas, para que el extremo final del último enlace del conjunto se ubique exactamente sobre el punto dado.

Los dos problemas anteriormente descritos conforman un par principal–dual, conocido como *Problema cinemático*; el primero es llamado generalmente *Problema cinemático directo*, en tanto que el segundo se denomina *Problema cinemático inverso* (Figura 2.1). Sin embargo existe una diferencia significativa entre los dos casos, ya que el problema cinemático directo tiene solución única, en tanto que el problema cinemático inverso puede tener múltiples soluciones (casi siempre un número infinito de soluciones) aún a pesar de las restricciones impuestas por la configuración del conjunto de enlaces. Esta dificultad es solucionada con base en los fenómenos físicos implícitos en el funcionamiento de un sistema autónomo de enlaces y articulaciones (brazo robot) añadiendo al proceso de

determinación del ángulo para cada articulación algún método para minimizar el efecto de variables como momentos, esfuerzos o recorrido angular desde la posición inicial.



**Figura 2.1** Problema cinemático directo e inverso.

Para la reconstrucción de objetos físicos mediante la recuperación de un muestreo de puntos en el espacio, obtenidos desde su superficie mediante un sistema de enlaces y articulaciones, el interés se centra en la solución del problema cinemático directo, al cual se dedica la sección 2.1. La sección 2.2 se concentra en la estructura física seleccionada y la aplicación del problema cinemático directo al prototipo de herramienta para la ubicación de coordenadas resultado del presente trabajo.

## 2.1. PROBLEMA CINEMÁTICO DIRECTO.

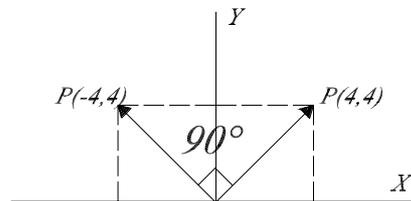
La cinemática directa de un brazo articulado involucra el estudio analítico de la geometría del movimiento de un conjunto de cuerpos rígidos (enlaces), conectados en serie mediante articulaciones de revolución. El estudio de la variación de la posición de cada enlace, respecto a un sistema fijo de coordenadas de referencia, se efectúa como una función del tiempo, sin considerar las fuerzas ni los momentos que originan dicho movimiento. De esta manera se describe el desplazamiento espacial del extremo del elemento final del brazo articulado, denominado *efector final*.

Para el estudio de la cinemática directa, la principal pregunta a tener en cuenta, por su importancia teórica y práctica es la siguiente: para un *manipulador* determinado, dado el

vector de ángulos de las articulaciones  $q(t) = (q_1(t), q_2(t), q_3(t), \dots, q_n(t))$  y los parámetros geométricos del elemento, donde  $n$  es el número de grados de libertad ¿Cuál es la orientación y posición del efector final del manipulador con respecto a un sistema de coordenadas de referencia? Para contestar satisfactoriamente esta pregunta debe recurrirse a ciertos conceptos básicos geometría cartesiana, la trigonometría y el álgebra lineal, que convergen en el concepto de las transformaciones geométricas para el cambio de sistemas de coordenadas. Debido a un cierto grado de abstracción requerido para plantear un modelo geométrico coherente, que conduzca eficientemente a la solución del problema cinemático directo, es conveniente hacer una aproximación desde los casos más elementales en dos dimensiones, para luego, por medio de generalizaciones, acercarse progresivamente a la situación real de una estructura de múltiples enlaces y articulaciones en el espacio tridimensional.

### **2.1.1. Modelo Geométrico para la Ubicación de Puntos en el Espacio**

El caso más sencillo de transformación de coordenadas en el que se puede pensar es el de un único enlace en el espacio bidimensional. Dado un enlace  $L_1$  definido como la recta que une al origen con el punto  $P = (x, y)$  sobre el plano, al efectuar una rotación de  $P$  sobre el origen, es de esperar que sus coordenadas se desplacen a un nuevo lugar del plano. Por ejemplo, al realizar una rotación de  $\alpha = 90^\circ$  sobre  $P = (4, 4)$ , el punto resultante será  $P = (-4, 4)$  (Figura 2.2). El signo del ángulo de rotación sobre el plano determina el sentido en el que se debe efectuar el giro. Si el ángulo es positivo la rotación se efectúa en el sentido contrario al de las manecillas del reloj, si es negativo la rotación tendrá el mismo sentido de las manecillas del reloj.



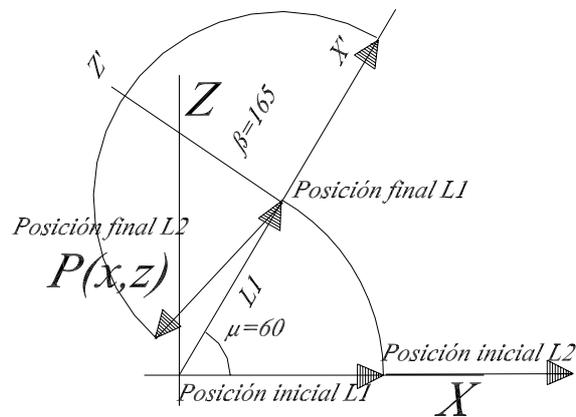
**Figura 2.2.** Rotación de un punto bidimensional sobre el origen.

Supóngase ahora que en el punto P, el cual es el extremo final del enlace  $L_1$  se añade un nuevo enlace  $L_2$ . De este modo se obtiene una nueva articulación, ubicada sobre P. El extremo al final del conjunto de enlaces, ubicado en el extremo final de  $L_2$  (efector final) adquiere una nueva posición que depende de las rotaciones realizadas tanto por  $L_1$  como  $L_2$ <sup>1</sup>. La medición de los grados de rotación de cada uno de los enlaces se realiza con respecto a una posición inicial determinada (Figura 2.3).

La esencia de la solución del problema de cinemática directa para un sistema de múltiples enlaces consiste en asignar a cada articulación su propio sistema de coordenadas, y definir la posición de cada enlace respecto a la articulación que lo une con el enlace inmediatamente anterior. En este caso, la posición inicial del primer enlace se encuentra en el origen del sistema de coordenadas  $XY$  con coordenadas  $(l_1, 0)$ , donde  $l_1$  es la longitud del enlace  $L_1$ ; para el segundo enlace la posición inicial se encuentra en el punto  $(l_2, 0)$ , definido sobre el sistema de coordenadas auxiliares  $X'Y'$ , siendo  $l_2$  la longitud del enlace  $L_2$ .

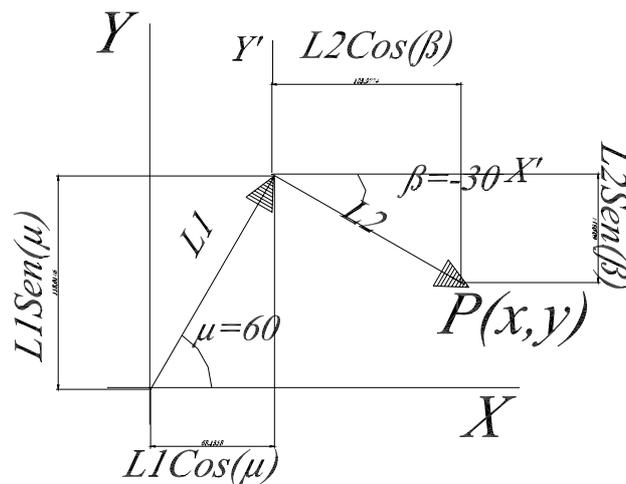
---

<sup>1</sup> Se asume que una rotación efectuada sobre el enlace  $L_1$  afectará simultáneamente la orientación de  $L_2$  respecto al origen del sistema de coordenadas.



**Figura 2.3.** Rotación sobre un plano con dos articulaciones

A partir de lo anterior se generalizan las componentes del punto final para el caso bidimensional, dadas las longitudes de los enlaces y los ángulos de rotación de cada una de ellos respecto a la posición inicial.



**Figura 2.4** Rotación en dos dimensiones con dos articulaciones

La Figura 2.4 muestra el cálculo de las coordenadas del efector final  $E = (x, y)$ , obteniéndose la expresión:

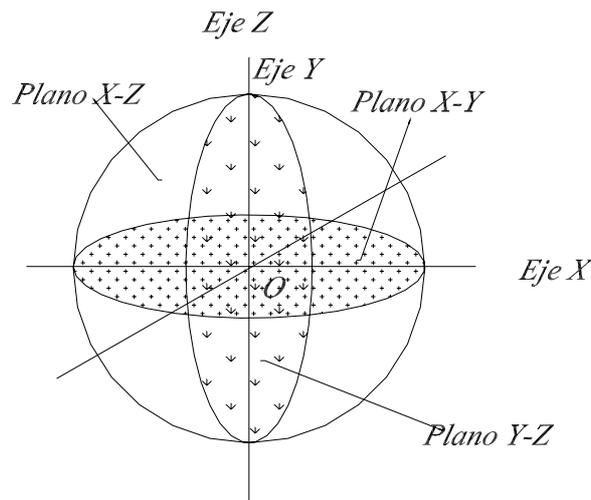
$$x = l_1 \cos(\mu) + l_2 \cos(\beta)$$

$$y = l_1 \text{sen}(\mu) + l_2 \text{sen}(\beta)$$

Esta expresión se puede generalizar fácilmente debido a que los sistemas de coordenadas para cada articulación son equivalentes. Así para un conjunto de  $n$  enlaces en el plano, las coordenadas del efector final tienen la forma general:

$$\begin{aligned}x &= L_1 \cos(\beta_1) + L_2 \cos(\beta_2) + \dots + L_n \cos(\beta_n) \\y &= L_1 \sin(\beta_1) + L_2 \sin(\beta_2) + \dots + L_n \sin(\beta_n)\end{aligned}$$

A diferencia del caso bidimensional, en el espacio tridimensional los vectores que representan a los enlaces pueden efectuar rotaciones sobre tres planos diferentes:  $XY$ ,  $XZ$  y  $YZ$  (Figura 2.5). Entonces, para la solución del problema cinemático directo en tres dimensiones se debe identificar tanto los enlaces que conforman al sistema articulado como el plano de rotación sobre el cual rota cada uno de ellos.



**Figura 2.5.** Planos de rotación en tres dimensiones.

Análogamente al caso bidimensional, para determinar la posición del efector final es necesario calcular cada una de sus componentes en  $\mathbf{R}^3$ . Los ejes de rotación son elegidos arbitrariamente, aunque es preferible asociarlos a las características geométricas del sistema de enlaces articulados. Antes de plantear un modelo geométrico en tres dimensiones es necesario formalizar el tratamiento matemático de las transformaciones geométricas para hacer una descripción de un manejo matemático más uniforme (y por ende más comprensible) de la aplicación de transformaciones a lugares geométricos en  $\mathbf{R}^3$ . El

principio de esta formalización consiste en la utilización de representaciones matriciales en vez de simples ecuaciones algebraicas.

### 2.1.2. Matrices de Transformación Geométrica.

En la sección anterior se hizo una introducción al desarrollo trigonométrico que conduce a la solución del problema cinemático directo en dos dimensiones. Una estrategia que perfecciona tal desarrollo y permite abordar el problema desde el punto de vista computacional con mayor claridad, consiste en representar las transformaciones geométricas como multiplicaciones matriciales. Bajo esta perspectiva, un punto de  $\mathbf{R}^n$  es representado como un vector  $n$ -dimensional, y la transformación se aplica multiplicando la matriz que representa la transformación por el vector.

#### 2.1.2.1 Matrices de Transformación Bidimensionales.

Supóngase que el vector  $V$  con magnitud  $r$  en el plano  $XY$  (Figura 2.6) gira un ángulo  $p$  en sentido positivo; el vector resultante de la rotación de  $V$  es  $V'$ . Entonces:

$$v = \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{y} \quad v' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

por trigonometría se sabe que

$$\begin{aligned} x &= r \cos \beta & y &= r \operatorname{sen} \beta \\ x' &= r \cos(p + \beta) & y' &= r \operatorname{sen}(p + \beta) \end{aligned}$$

Utilizando las identidades trigonométricas se obtiene

$$\begin{aligned} r \cos(p + \beta) &= r \cos p \cos \beta - r \operatorname{sen} p \operatorname{sen} \beta \\ r \operatorname{sen}(p + \beta) &= r \operatorname{sen} p \cos \beta + r \cos p \operatorname{sen} \beta \end{aligned}$$

De donde

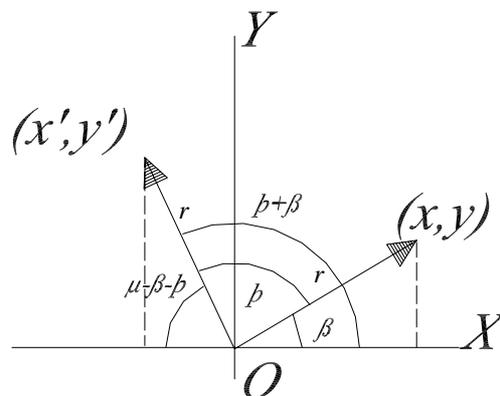
$$\begin{aligned} x' &= x \cos p - y \operatorname{sen} p \\ y' &= x \operatorname{sen} p + y \cos p \end{aligned}$$

La determinación para las coordenadas de  $V'$  se puede integrar en una sola operación matricial a partir de las ecuaciones anteriores

$$\begin{bmatrix} \cos b & -\operatorname{sen} b \\ \operatorname{sen} b & \cos b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

La forma matricial de la rotación de un punto cualquiera en un ángulo  $b$  alrededor del origen es entonces

$$\begin{bmatrix} \cos b & -\operatorname{sen} b \\ \operatorname{sen} b & \cos b \end{bmatrix}$$



**Figura 2.6.** Obtención de la Matriz de rotación en dos dimensiones

### 2.1.3.2 Matrices de Transformación Tridimensionales.

En el espacio tridimensional, el procedimiento para encontrar las matrices de transformación para la rotación de un punto sobre un plano determinado tiene la misma estructura que el expuesto anteriormente para dos dimensiones. La idea de una matriz de rotación tridimensional sobre un plano dado es que altere las componentes del punto sobre los dos ejes que generan el plano sin alterar la tercera componente. Entonces, la rotación sobre el plano  $XY$ <sup>2</sup> de un punto  $P = (x, y, z)$ , en un ángulo  $\theta$  produce un punto  $P' = (x', y', z')$ , cuyas coordenadas se obtienen mediante las ecuaciones

$$x' = x \cos \theta - y \operatorname{sen} \theta$$

$$y' = x \operatorname{sen} \theta + y \cos \theta$$

$$z' = z$$

---

<sup>2</sup> Lo que equivale a decir alrededor del eje Z.

Lo anterior, expresado en forma matricial equivale a

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\operatorname{sen} \theta & 0 \\ \operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

La matriz de rotación alrededor del eje z es por lo tanto

$$\operatorname{Rot}(z, \theta) = \begin{bmatrix} \cos \theta & -\operatorname{sen} \theta & 0 \\ \operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

donde  $\operatorname{Rot}(z, \theta)$  denota la matriz de rotación alrededor del eje z en un ángulo  $\theta$ .

Las matrices para las rotaciones sobre los ejes x y y se elaboran de forma similar. Así, para la rotación alrededor del eje X se tiene que

$$\operatorname{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\operatorname{sen} \theta \\ 0 & \operatorname{sen} \theta & \cos \theta \end{bmatrix}$$

y sobre el eje Y

$$\operatorname{Rot}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & -\operatorname{sen} \theta \\ 0 & 1 & 0 \\ \operatorname{sen} \theta & 0 & \cos \theta \end{bmatrix}$$

Vale la pena resaltar que al aplicar a un punto la matriz de rotación sobre un plano del sistema de coordenadas, solo son alteradas las componentes que generan a dicho plano. Por ejemplo, dado el punto  $P = (1, 1, 1)$ , el resultado de aplicar independientemente las transformaciones de rotación sobre los tres ejes de coordenadas un ángulo de  $180^\circ$  se obtienen los siguientes resultados:

En el caso de la transformación sobre el eje X, solo cambian las componentes y y z

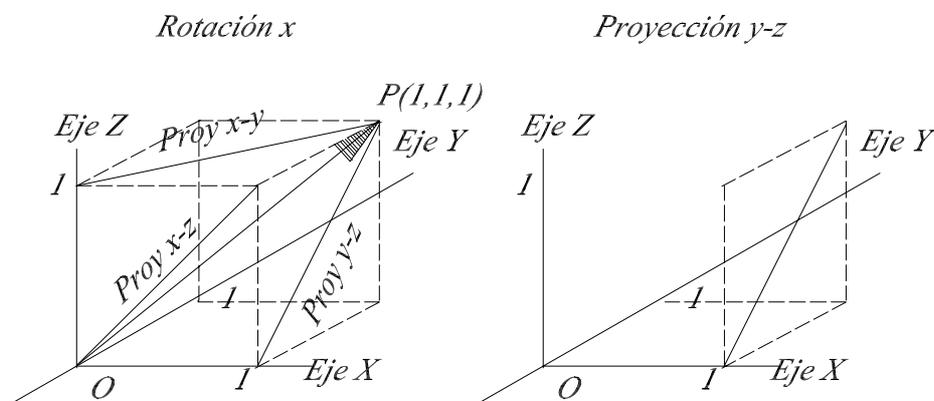
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 180 & -\operatorname{sen} 180 & 0 \\ 0 & \operatorname{sen} 180 & \cos 180 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

en la rotación alrededor de  $Y$  la componente  $y$  permanece constante, mientras que  $x$  y  $z$  cambian

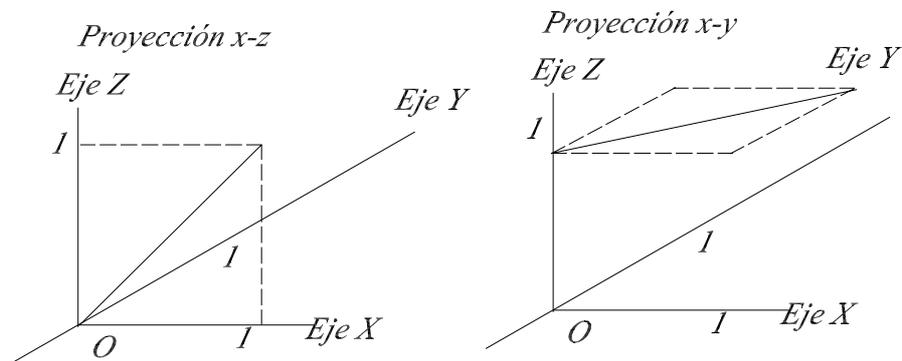
$$\begin{bmatrix} \cos 180 & 0 & \sin 180 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin & 0 & \cos 180 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 0 \end{bmatrix}$$

y, en la rotación alrededor del eje  $Z$  sólo la componente  $z$  permanece constante

$$\begin{bmatrix} \cos 180 & -\sin 180 & 0 & 0 \\ \sin 180 & \cos 180 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 0 \end{bmatrix}$$

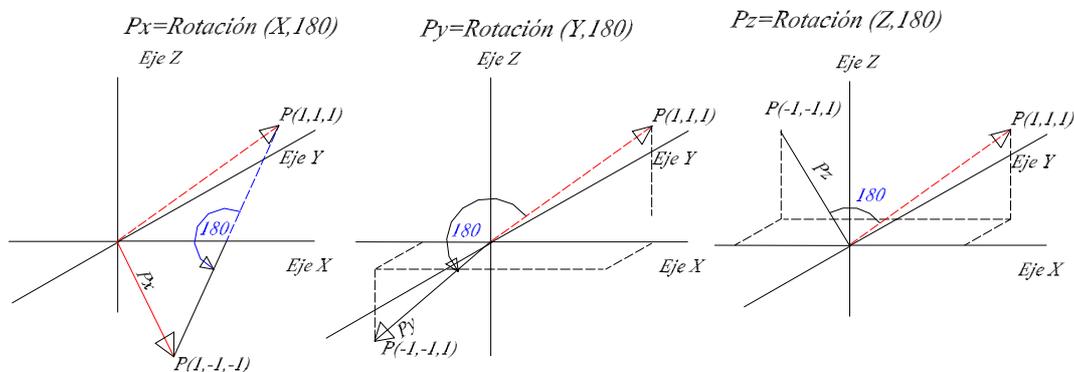


**Figura 2.7.** Posición inicial. Proyección plano  $YZ$



**Figura 2.8.** Proyecciones en los planos XZ y XY.

En cada uno de los puntos resultantes la componente asociada con la transformación no sufre ninguna variación. En este ejemplo, las figuras 2.7 y 2.8 muestran gráficamente las proyecciones correspondientes al punto  $P = (1, 1, 1)$ , cuando cada transformación es aplicada. Los resultados se identifican como  $P_x$ ,  $P_y$ ,  $P_z$ . Estos corresponden al efecto de las matrices de transformación sobre cada uno de los ejes  $X$ ,  $Y$ , y  $Z$  respectivamente, aplicando  $180^\circ$  como ángulo de giro (Figura 2.9).



**FIGURA 2.9.** Efectos de las Transformaciones sobre el punto  $P = (1, 1, 1)$

## 2.2. CONFIGURACION FISICA DE UNA HERRAMIENTA DE UBICACIÓN DE COORDENADAS MEDIANTE ARTICULACIONES

Como se mencionó anteriormente, una de las alternativas para la implementación de un sistema de ubicación de coordenadas corresponde a un conjunto de enlaces unidos mediante articulaciones, similar en apariencia a los brazos robóticos usados en la industria automatizada, con la diferencia de que el movimiento de una herramienta de ubicación de

coordenadas de este estilo es efectuado por un operador humano. Este enfoque fue el utilizado para la elaboración de la herramienta desarrollada como parte de este proyecto.

Las características principales que permiten establecer el desempeño de una herramienta de ubicación de coordenadas análogamente a los brazos robóticos son el espacio de trabajo y los grados de libertad.

### **2.2.1 Espacio de trabajo**

Dadas las restricciones de la longitud de los enlaces y los planos de rotación de las articulaciones, existe una región limitada del espacio dentro de la cual el efector final de la herramienta puede desplazarse. Esta región es denominada espacio de trabajo de la herramienta y es única para una configuración determinada.

### **2.2.2 Grados de Libertad.**

Este aspecto hace referencia al número de articulaciones que posee la herramienta. Las articulaciones se ubican en la unión de cada par de enlaces de la herramienta y se considera como articulación el extremo final del primer enlace asumiendo que este tiene la posibilidad de girar sobre algún plano. Cada articulación tiene asociado un plano de rotación; un caso particular se presenta cuando una articulación puede girar sobre más de un plano, en cuyo caso el movimiento se descompone en las proyecciones de los ángulos formados con cada plano del sistema de coordenadas.

Al momento de determinar la solución del problema cinemático directo para una posición específica, se asigna un sistema auxiliar de coordenadas tal como fue expuesto en secciones anteriores. El sistema principal de coordenadas en el cual se desea ubicar la posición del efector final generalmente se asigna a la posición del extremo inicial del primer enlace.

La maniobrabilidad de la herramienta, definida como la facilidad con la que el efector final accede a cualquier posición del espacio de trabajo, aumenta en la medida en que se incrementa el número de articulaciones aunque la solución del problema cinemático directo se hace más complejo.

### 3. GENERACIÓN DE ENMALLADOS

Existen varias alternativas para modelar geoméricamente objetos tridimensionales en el computador. Algunas hacen uso de superficies definidas paraméricamente, tales como las superficies de Bézier, los B-splines o los  $\beta$ -splines. Este tipo de representaciones es muy exacto pues se modela estrictamente la curvatura de la superficie punto a punto mediante una serie de ecuaciones paraméricas, si bien la carga computacional que implica el cálculo de tales ecuaciones para cada punto es bastante pesada. Un enfoque menos exacto pero más eficiente es utilizar una aproximación de la superficie del objeto mediante una malla de polígonos convexos, dejando el trabajo de búsqueda del realismo visual a los procesos de despliegue en pantalla del modelo tridimensional.

Sin embargo, los algoritmos para la generación automática de mallas poligonales que representen objetos tridimensionales apenas están comenzando a ser explorados, su complejidad es bastante elevada y se presentan múltiples dificultades y limitaciones para generar un modelo tridimensional a partir de un conjunto de datos (como puntos sobre la superficie del objeto) sin un conocimiento previo de la geometría del objeto a generar o alguna introducción de datos adicionales por parte del usuario. A pesar de tales dificultades, las tecnologías computacionales encargadas de solucionar este tipo de problemas avanzan vertiginosamente desde su reciente aparición hace algunas décadas.

La generación automática de mallas de polígonos es tan solo uno de los aspectos de un campo de la computación recientemente desarrollado: la Geometría Computacional. A dicho campo compete el análisis, diseño, implementación y evaluación de procedimientos destinados a la ejecución de tareas de índole geométrica, requeridas por ciertas áreas de la ingeniería y las ciencias que hacen un uso intensivo de los computadores, como la Simulación, el análisis mediante Elementos Finitos, la Programación Lineal, el Análisis de

Grafos, etc. El auge de la Geometría Computacional ha brindado una serie de alternativas computacionales para el modelamiento y la solución satisfactoria de múltiples problemas que anteriormente podían ser tratados únicamente mediante técnicas analíticas demasiado complejas o aproximaciones numéricas que implican la existencia de un cierto grado de error.

Aunque en principio las técnicas de geometría computacional no tienen un vínculo explícito con la presentación de modelos gráficos por computador, las técnicas automáticas de generación de mallas poligonales, empleadas ordinariamente para simulaciones de fenómenos físicos complejos mediante elementos finitos, podrían ser empleadas, en algunos de sus aspectos, para la generación de cubrimientos poligonales que aproximen una superficie tridimensional dada a partir de un conjunto de puntos ubicados sobre la superficie del objeto tridimensional a representar.

### **3.1. CONCEPTOS BÁSICOS DE GEOMETRÍA COMPUTACIONAL**

Los procedimientos estudiados por la Geometría computacional van desde la simple determinación de distancias entre lugares de un espacio dado, hasta la descripción geométrica de la intersección de poliedros complejos en espacios  $n$ -dimensionales. La solución de este tipo de problemas implica la utilización de ciertos lemas geométricos que se han propuesto a lo largo del último siglo sobre la base de las características comunes a los espacios  $n$ -dimensionales, partiendo de los postulados de la geometría analítica y el álgebra lineal; sin embargo, la aplicación práctica de dichos lemas a la elaboración de procedimientos que llevan a cabo tareas de Geometría Computacional se remonta aproximadamente 25 años atrás, cuando el avance de la capacidad de procesamiento y velocidad de los computadores permitió pensar seriamente en la posibilidad de implementar tales programas con los parámetros de rendimiento y fiabilidad deseados para problemas de ingeniería e investigación científica de la vida real.

#### **3.1.1. Búsqueda geométrica**

Los problemas de búsqueda geométrica son los más comunes y “simples” de la geometría computacional, además de estar entre los primeros tratados por esta disciplina. Un problema de búsqueda geométrica consiste generalmente en la determinación de la relación

entre un conjunto inicial de datos, conocido como ‘archivo’, y un nuevo dato insertado al conjunto, conocido como ‘muestra’. El conjunto inicial de datos no es una simple colección de registros, sino que puede ser la representación de estructuras geométricas complejas como polígonos o poliedros o regiones definidas de forma paramétrica o funcional.

La solución a los problemas de búsqueda geométrica constituye la base de la geometría computacional, pues cualquier problema de índole geométrico, sea cual sea su complejidad, puede ser solucionado a partir de su descomposición en problemas simples de búsqueda geométrica y la determinación de la solución a cada *sub-problema* (convergiendo así a la solución del problema original), si bien este enfoque no es el único que puede conducir a una solución correcta y, en ocasiones, la solución óptima al problema original se obtiene a partir de la aplicación de técnicas diferentes. En muchos casos, la demanda de un orden de complejidad determinado para el algoritmo de búsqueda geométrica implica la utilización de procedimientos accesorios de pre-procesamiento, especialmente ordenamientos según alguna variable geométrica particular al problema en cuestión.

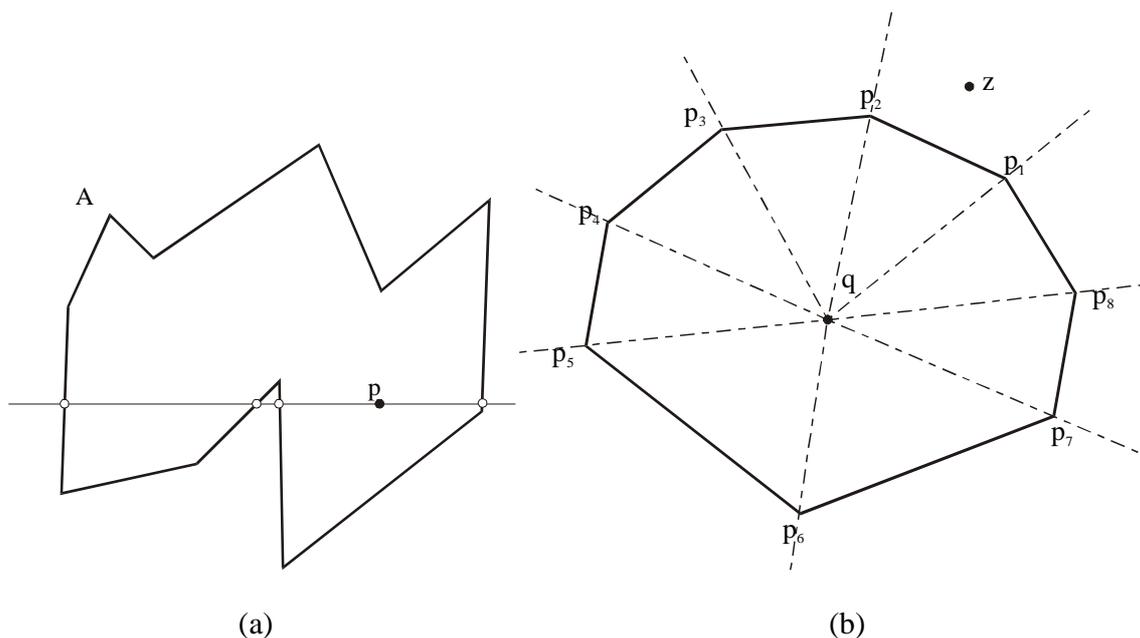
A pesar de lo abstracto que pueden parecer los problemas de búsqueda geométrica que se fundamentan en los espacios cartesianos  $n$ -dimensionales, lo cierto es que cada componente en un espacio tal puede ser la abstracción de cada variable en una amplia gama de problemas de la vida real, que van desde la estadística hasta los sistemas de información georreferenciados, pasando por los reportes obtenidos eficientemente de bases de datos de gran tamaño, aunque enfoques más ‘literales’ son también ampliamente utilizados, en especial por campos como la simulación de procesos físicos.

En la búsqueda geométrica se pueden distinguir dos tipos de problemas principales: los problemas de localización de puntos y los problemas de búsqueda de rango, los cuales constituyen, de cierta manera un par principal-dual. En los primeros el archivo es generalmente una partición del espacio en regiones definidas mediante relaciones geométricas y la muestra es un nuevo punto; así el problema consiste en determinar en cuál región de la partición está localizada la muestra. En los segundos el archivo es un conjunto de puntos en un espacio dado, en tanto que la muestra es una región definida de alguna forma sobre dicho espacio; el problema es obtener los puntos del archivo que se encuentran contenidos en dicha región.

### 3.1.1.1. Localización de Puntos

Uno de los problemas más representativos de la geometría computacional es el de la localización de puntos en el espacio sobre subdivisiones espaciales dadas. El objeto de la localización un punto sobre una subdivisión espacial es decidir en que elemento de la subdivisión está contenido el punto, razón por la cual este tipo de problemas es también conocido como de inclusión de puntos.

Los problemas de localización de puntos son tan sencillos o complejos como compleja sea la subdivisión (partición) espacial, sobre la cual se desea realizar la localización, así como la dimensión del espacio mismo. Las particiones están, en la mayoría de los casos, especificadas por restricciones lineales que definen polígonos o poliedros (generalmente convexos), aunque determinados problemas pueden considerar subdivisiones espaciales conformadas por polígonos complejos, con concavidades y auto-intersecciones, además de particiones construidas a partir de restricciones paramétricas o funcionales.



**Figura 3.1.** Problemas de localización de puntos en una partición espacial definida por un único polígono. El método usado en (a) utiliza una bandera de paridad según el número de cortes de la línea de rastreo con las aristas del polígono: el punto  $p$  está al interior si la bandera es impar. El método mostrado en (b), válido solo para polígonos convexos, hace una búsqueda de la ‘cuña’ a la que pertenece  $z$ ; una vez se ha establecido que  $z$  se encuentra en la cuña  $p_1q p_2$ , se establece de qué lado de la arista  $p_1p_2$  se encuentra  $z$

El problema de localización de puntos más simple es aquel referido a una partición del espacio bidimensional en dos regiones, una al interior y otra al exterior de un polígono. El

asunto es determinar si la muestra (el punto insertado) está dentro o fuera del polígono mencionado. Entre las técnicas usadas para la solución de este problema se encuentra la línea de rastreo horizontal o *scan-line*, que verifica una bandera de paridad según las intersecciones de la línea de rastreo, así, si la bandera es par el punto se encuentra por fuera del polígono, si es impar el punto está al interior de éste, (Fig. X.1 (a)). Otras técnicas utilizan ordenamientos de los vértices del polígono (Fig. X.1 (b)) o pertenencia a cada semiplano formado por la extensión del segmento correspondiente a cada arista del polígono (en el caso de que este sea convexo).

Para la localización de puntos en subdivisiones espaciales más complejas existen diferentes técnicas, como el método *slab*, el método de las cadenas monótonas, los refinamientos de triangulaciones, el método del trapecioide, etc.<sup>1</sup>, aunque dichos métodos están diseñados para la solución de problemas en el plano y no siempre pueden ser generalizados a dimensiones de orden superior, por lo que muchos problemas requieren del planteamiento de soluciones específicas fundamentadas en sus características particulares. Tal es el caso de problemas que involucran subdivisiones espaciales paramétricas o funcionales.

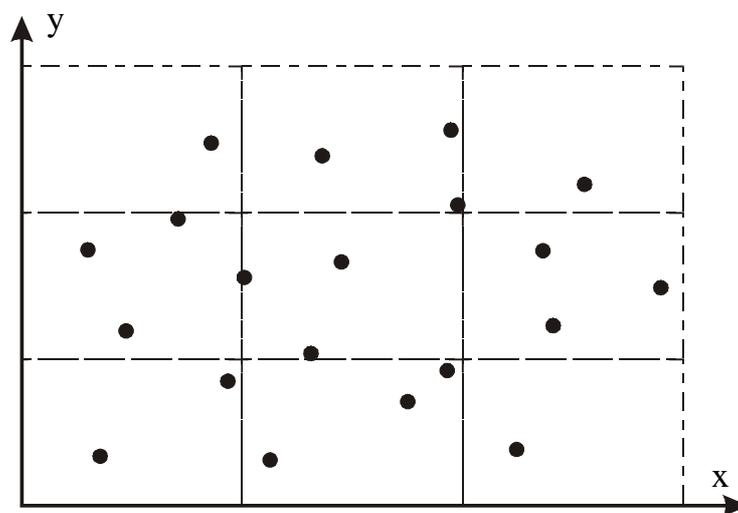
### 3.1.1.2. Búsqueda de Rango

Como se dijo previamente, la búsqueda de rango puede ser interpretada como el dual del problema de localización de puntos en el espacio. En los problemas de búsqueda de rango el archivo está constituido por una colección de  $k$ -úplras de la forma  $X = (x_1, x_2, \dots, x_k)$ , que podrían ser asumidas como lugares en un espacio cartesiano  $k$ -dimensional, en tanto que la muestra es una región *libremente movable* definida de alguna forma sobre dicho espacio. Según lo anterior, los problemas de búsqueda de rango se pueden clasificar en dos categorías principales: la primera está conformada por los problemas en los que lo relevante de la búsqueda es el conteo del número u obtención de puntos que pertenecen a la región-muestra; la segunda contiene los problemas donde es necesaria la recuperación o reporte de cada uno de los lugares contenidos en la región.

---

<sup>1</sup> Para más detalles sobre cada método ver Preparata – Shamos. Computational Geometry, an Introduction.

Un enfoque unificador para los problemas de búsqueda de rango en general establece que todo registro del archivo está asociado a un elemento de un semi-grupo conmutativo y sobre el cual se encuentra definida la operación '\*'. De este modo, en los problemas de reporte cada registro se asocia a su índice dentro del archivo, y \* corresponde a la unión de conjuntos, mientras que los problemas de conteo cada registro se asocia con la unidad, \* es entonces la suma aritmética. La búsqueda entonces ejecutará \* sobre cada registro del archivo que se encuentre contenido dentro de la región muestra.



**Figura 3.2.** Problemas de búsqueda de rango. Dado el rectángulo central la búsqueda de rango puede tener por objeto la obtención de las coordenadas de cada punto al interior de dicho rectángulo, o tan solo conocer el número de puntos contenidos por éste.

La búsqueda de rango está comúnmente asociada a la utilización de dominios *hiper-rectangulares*, regiones que parten de la generalización para espacios  $n$ -dimensionales de regiones definidas mediante límites ortogonales respecto a los ejes y entre sí, puesto que estos hacen trivial la verificación de inclusión de lugares geométricos del espacio en cuestión (ya que la representación de dichos dominios equivale al producto cartesiano de intervalos lineales sobre los distintos ejes de coordenadas<sup>2</sup>), si bien muchos problemas hacen uso de regiones de diferente morfología dependiendo de su naturaleza. El problema es que las diferentes soluciones que han sido exitosamente desarrolladas para el caso de regiones hiper-rectangulares no pueden ser aplicadas directamente a problemas que

---

<sup>2</sup> Producto cartesiano se puede definir, en este caso, como el hiper-paralelepípedo definido por los intervalos, cuyas caras son siempre paralelas a los ejes de coordenadas cartesianas.

involucran otro tipo de regiones. Una de las soluciones planteadas para estos casos es la de modelar tales regiones (esferas, por ejemplo) como familias de dominios hiperrectangulares; aunque esta aproximación presenta un pésimo desempeño en situaciones del *peor caso*, su aplicación a problemas reales podría comportarse razonablemente bien en la mayoría de los casos.

Los métodos clásicos aplicados a la solución de problemas de búsqueda de rango hacen un uso recurrente de algoritmos de pre-procesamiento, con el fin de disponer los datos contenidos en el archivo de tal forma que los subsiguientes procedimientos de búsqueda se desempeñen de manera óptima. Las estructuras más empleada por tales métodos son sin duda los árboles binarios, y entre los principales temas de estudio en éste contexto es la determinación de la forma más conveniente de disponer los registros del archivo a través del árbol de búsqueda, teniendo en cuenta que el trabajo de disponer los datos sobre la estructura de búsqueda –almacenamiento– implica un incremento en el orden de complejidad del procedimiento de búsqueda de rango como un todo. Entre los métodos generales más conocidos se encuentran el árbol binario multidimensional (*k-D tree*), el árbol de rango y el ‘acceso directo’, descritos con profundidad en el trabajo de Preparata y Schamos, aunque existen trabajos más recientes para la solución de problemas que tratan con regiones de morfología particular.

### 3.1.2. Simplexes<sup>3</sup> y Polítopos

La definición de simplexe, aunque podría parecer trivial e innecesaria, es primordial para algunas de las definiciones y postulados de mayor complejidad en la geometría computacional. Para llegar a una definición satisfactoria de simplexe se debe introducir el concepto de polítopo. Un conjunto poliédrico en  $R^n$  puede ser considerado como la intersección de un conjunto finito de hemi-espacios de  $R^n$ . Un conjunto poliédrico es siempre convexo pues cualquier hemi-espacio de  $R^n$  es por definición un conjunto convexo y la intersección de conjuntos convexos es también convexa.

---

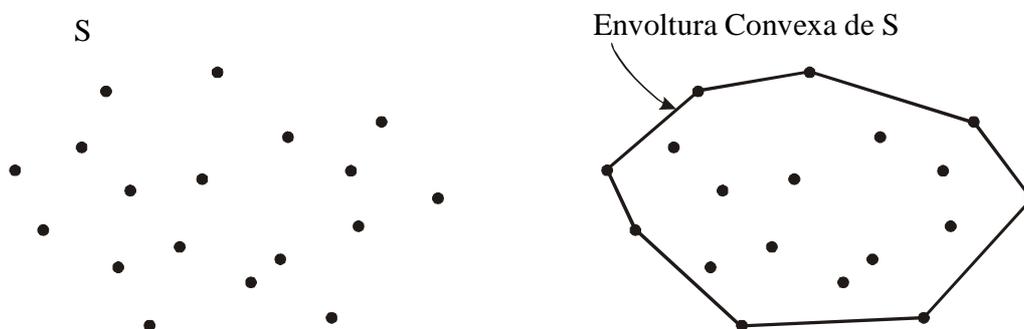
<sup>3</sup> El término inglés para definir esta estructura es *simplex*, los trabajos escritos en francés la denotan como *simplexe*. La literatura sobre este tópico en español es bastante escasa, En el presente trabajo se escogió el término *simplexe* el cual es usado en adelante.

En la literatura de geometría computacional un conjunto convexo  $n$ -dimensional es conocido generalmente como  $n$ -politopo o simplemente politopo. El mínimo conjunto de que define completamente a  $R^n$  debe tener al menos  $n + 1$  puntos siempre y cuando el conjunto de puntos sea linealmente independiente (en  $R^2$  los puntos no deben ser colineales, en  $R^3$  no deben ser coplanares, etc.). Un politopo con exactamente  $n + 1$  vértices que cumplen con la restricción de independencia lineal se denomina simplexe. En  $R^2$  un simplexe será simplemente un triángulo, en  $R^3$  un tetraedro, en dimensiones superiores se habla de simplexe.

### 3.1.3. Envolturas Convexas (Convex Hulls)

Uno de los problemas clásicos de la geometría computacional es la construcción de la envoltura convexa de un conjunto dado de lugares en el espacio, ya que su tratamiento exhaustivo y sistemático nació prácticamente con la noción de geometría computacional varias décadas atrás. Dada su gran aplicabilidad, la elaboración y optimización de procedimientos para la determinación de envolturas convexas han sido trabajadas por múltiples autores, y su desarrollo podría ser el punto de partida de un estudio sobre la reciente evolución de la investigación en el campo de los algoritmos.

Una envoltura convexa se puede definir como el mínimo conjunto convexo que contiene totalmente a una colección de lugares en el espacio, La restricción del tamaño del conjunto conduce a pensar que los límites de la envoltura convexa siempre contendrán algunos de los puntos exteriores de la colección (Fig. 3.3). Una envoltura convexa en  $R^n$  de un conjunto finito de puntos es un  $n$ -politopo.



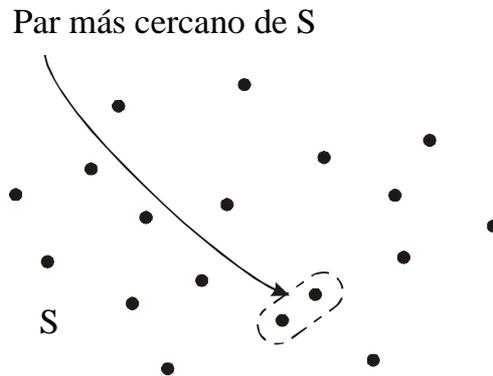
**Figura 3.3.** Envoltura convexa bidimensional.

La teoría y aplicaciones de las envolturas convexas en dos y más dimensiones han sido estudiadas extensamente y a profundidad por múltiples autores, un sinnúmero de métodos y algoritmos ha sido propuesto para la determinación de la envoltura convexa de conjuntos de puntos en el espacio, tanto arbitrarios como definidos a partir de ciertas propiedades geométricas. Uno de los aspectos a los que se ha dedicado un mayor esfuerzo es la optimización de los procedimientos para la construcción de envolturas convexas en 2 y 3 dimensiones, encontrándose en la actualidad algoritmos con una complejidad que va desde  $O(N^4)$ , como el *Graham's Scan*, hasta  $O(N \log N)$  para algunos métodos basados en técnicas de *divide-y-conquista* en el plano. La construcción de la envoltura convexa en dimensiones de orden superior aumenta su complejidad de forma exponencial y, hasta el momento, no parece haber un enfoque evite este incremento en la complejidad.

#### **3.1.4. Relaciones de Vecindad**

En el campo de los problemas de vecindad y proximidad se puede observar la misma tendencia que en los demás problemas de carácter geométrico, ya que el caso bidimensional resulta ser por mucho más claro y las soluciones planteadas para éste más elegantes y óptimas que aquellas planteadas para dimensiones de orden mayor que 2.

El problema de vecindad más conocido y estudiado corresponde a la determinación los dos puntos con la menor distancia pertenecientes a un conjunto  $S$  de puntos en el espacio, llamado *par más cercano*. (Figura 3.4). Éste es, de hecho, uno de las cuestiones fundamentales de la geometría computacional, tanto por su interés teórico (debido a su relación con la solución de problemas más complejos) como por su utilidad práctica en aplicaciones reales. Se ha demostrado que el par más cercano de un conjunto de puntos  $S$  de  $R^n$  se puede encontrar en  $O(n N^2)$ .



**Figura 3.4.** El par más cercano de un conjunto de puntos.

La generalización del problema del par más cercano es el problema de encontrar *todos los vecinos más próximo* de un conjunto  $S$  de puntos. El objeto es encontrar el punto más cercano a cada uno de los puntos pertenecientes a  $S$ . La solución a este problema es una colección de pares  $(a, b)$  tal que  $a$  es el vecino más próximo de  $b$ . La relación ‘vecino más próximo’ no es necesariamente simétrico, es decir, que  $a$  sea *el vecino más próximo* de  $b$  no implica que  $b$  sea *el vecino más próximo* de  $a$

Algunas estructuras geométricas, como el Diagrama de Voronoï y la Triangulación de Delaunay, tienen algunas relaciones con los problemas de proximidad aunque sus definiciones son independientes y no dependen de problemas específicos como el *par más cercano*.

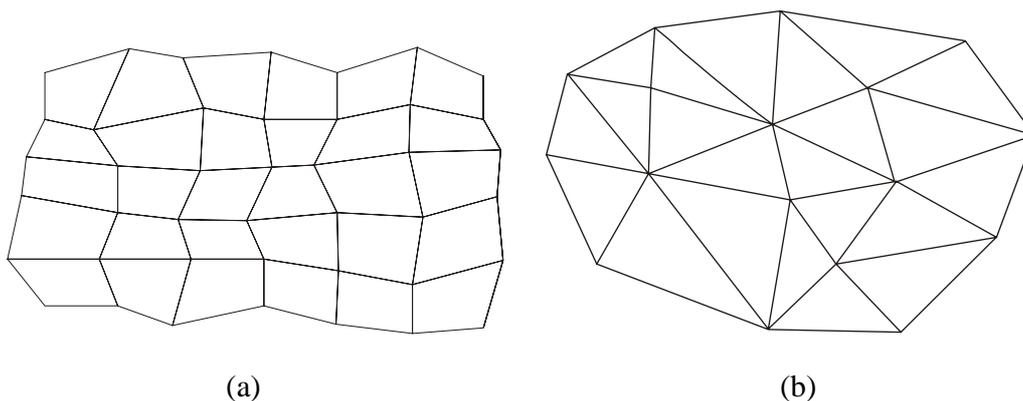
### 3.1.5. Enmallados

Entre los problemas más estudiados en la actualidad se encuentra la generación de cubrimientos espaciales de una región dada en  $\mathbf{R}^n$  mediante mallas de  $n$ -politopos, partiendo de determinadas características geométricas de dicha región, o bien de la consecución de muestreo del espacio según las necesidades de la aplicación en cuestión. En términos formales, para que un conjunto  $M$  de politopos, notados como  $K$ , constituya un enmallado válido de una región  $A$  de  $\mathbf{R}^n$ , cerrada y limitada, debe cumplir las siguientes condiciones:

- $A = \bigcup_{K \in M} K$
- Todo elemento  $K \in M$  es de interior no vacío.

- La intersección del interior de dos elementos cualesquiera de  $M$  es igual al vacío.
- La intersección de dos elementos cualesquiera de  $M$  es siempre el conjunto vacío, un punto, una arista o una cara (en dimensión tres o superiores).

Se conoce como “malla poligonal estructurada” a aquella que cumple con la restricción de que todos los nodos interiores de la malla (es decir, todos los que no pertenecen a la envoltura convexa del conjunto de nodos) tienen un número igual de elementos adyacentes; este tipo de enmallados es usado por aplicaciones de simulación de fenómenos físicos<sup>4</sup> que requieren de un cierto grado de regularidad en los elementos debido a que los cálculos de las variables principales para un elemento dado dependen de los valores de dichas variables para sus vecinos. Las mallas poligonales–poliédricas no estructuradas simplemente no presentan restricciones de vecindad (Figura 3.5).



**Figura 3.5.** Malla poligonal estructurada (a) y malla poligonal no-estructurada (b). En (a) el número de aristas incidentes a cada nodo interior de la malla es constante, cuatro, en tanto que en (b) los nodos internos pueden tener un número variable de aristas incidentes, en este caso de cuatro a siete.

Generalmente, los elementos que conforman a una malla poligonal estructurada son típicamente cuadriláteros en 2D y hexaedros en 3D. Los algoritmos empleados para la generación de mallas estructuradas involucran técnicas complejas de suavizado iterativo, que pretenden garantizar el encaje de los elementos generados con los límites o los dominios físicos. Donde se requieren límites no triviales, pueden emplearse técnicas de “bloque estructurado” que le permiten al usuario dividir el dominio en bloques topológicos. Los algoritmos de generación de mallas no estructuradas ignoran la restricción de un

---

<sup>4</sup> CFD

número fijo de elementos adyacentes a cada nodo, permitiendo que un número cualquiera de elementos converja a un nodo dado. Las mallas poligonales–poliédricas no estructuradas están conformadas generalmente por triángulos en 2D y tetraedros en 3D, aunque pueden existir mallas no estructuradas conformadas por elementos cuadrilaterales o hexaédricos.

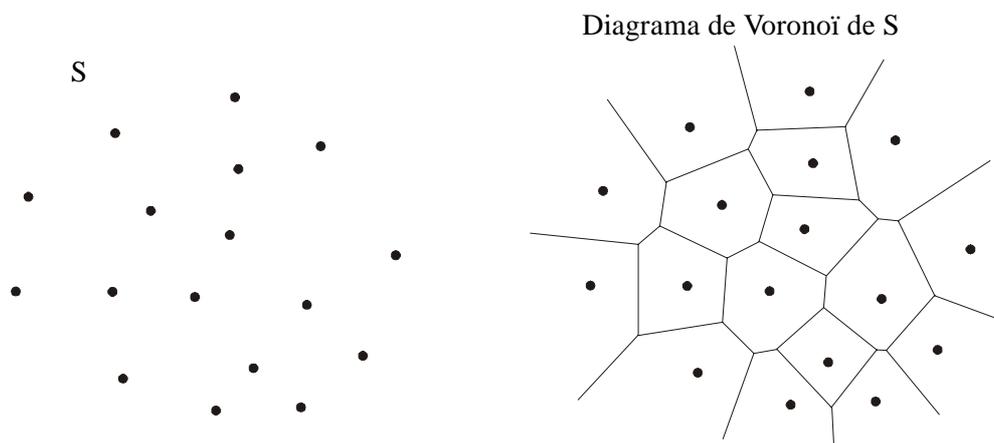
Se han planteado múltiples enfoques para generar este tipo de cubrimientos, utilizando estrategias como *divide y conquista*, *quadtrees – octrees*, *moverse adelante*, etc. Aunque hay muchas características comunes entre los algoritmos de generación de mallas poligonales–poliédricas estructuradas y no–estructuradas, la diferencia más significativa es la utilización por los métodos estructurados de algoritmos de suavizado iterativo para la consecución de mallas con un grado preestablecido de regularidad u homogeneidad.

Casos especiales de los enmallados en  $R^n$  son aquellos que tienen como característica principal es estar constituidos exclusivamente por simplexes  $n$ –dimensionales. Estos enmallados son conocidos indistintamente como triangulaciones (aunque algunos autores recientes hacen la diferenciación entre triangulaciones y *tetraedrizaciones*). Las triangulaciones están típicamente clasificadas entre los enmallados no–estructurados, aunque existen métodos para la generación de triangulaciones estructuradas, si bien éstas no siempre existen para conjuntos arbitrarios de puntos, requiriéndose muchas veces la inserción de nuevos puntos al conjunto inicial. Dentro de las posibles triangulaciones que se pueden construir sobre un conjunto cualquiera de lugares en el espacio es de especial interés la Triangulación de Delaunay, tratada en las secciones posteriores.

### **3.1.6. Diagrama de Voronoï – Triangulación de Delaunay**

En 1908 el matemático Ruso G. Voronoï planteó una estructura geométrica construida a partir del concepto de vecinos más cercanos, conocida en la actualidad como el Diagrama de Voronoï. Tal estructura ha sido punto de partida de múltiples trabajos teóricos y prácticos de geometría clásica, a principios del siglo XX, como de geometría computacional, desde la década de 1970. Para hacer una aproximación apropiada a este problema, es conveniente definir el diagrama de Voronoï en el caso bidimensional, pues la generalización a espacios de dimensiones de orden superior puede hacer bastante intrincada y confusa la descripción de las relaciones geométricas implícitas en la definición de esta estructura.

A continuación se presenta una definición del diagrama de Voronoï partiendo de la geometría bidimensional. Dado un conjunto de puntos  $S$  y dos puntos  $p_i$  y  $p_j$  pertenecientes a  $S$ , se puede denotar el *lugar de los puntos más cercanos a  $p_i$  que a  $p_j$*  como el semi-plano definido por la bisectriz ortogonal al segmento  $\overline{p_i p_j}$  en el cual se encuentra  $p_i$ ; a partir de lo anterior, el polígono de Voronoï de  $p_i$  se define como el lugar de los puntos de  $S$  más cercanos a  $p_i$  que a cualquier otro punto de  $S$ ; se deduce entonces que el polígono de Voronoï de  $p_i$  será igual a la intersección de los semi-planos formados por las bisectrices a los segmentos que unen a  $p_i$  con cada uno de los puntos de  $S$ . Aceptando que un semi-plano es siempre un conjunto convexo, la intersección de la colección de semi-plaños que contienen a  $p_i$  es también un conjunto convexo, luego el polígono de Voronoï de cualquier punto  $p_i$  que pertenece a un conjunto cualquiera de puntos  $S$  será siempre una *región poligonal convexa*<sup>5</sup> que contiene únicamente a  $p_i$ . Si el conjunto  $S$  contiene  $N$  puntos, entonces el polígono de Voronoï de cualquier punto de  $S$  estará limitado por un número menor o igual a  $N - 1$  aristas. El conjunto de los polígonos de Voronoï de todos los puntos pertenecientes a  $S$  constituye el llamado Diagrama de Voronoï, el cual genera un cubrimiento espacial completo, es decir sin regiones que no estén contenidas en alguno de los polígonos del diagrama (Figura 3.6).



**Figura 3.6** Diagrama de Voronoï Asociado a un conjunto de puntos dado. Obsérvese que las regiones exteriores, asociadas a los puntos que conforma la envoltura convexa del conjunto, no están completamente limitadas por aristas.

<sup>5</sup> Se evita usar el término polígono convexo, pues las regiones asociadas a los puntos externos del conjunto no necesariamente están completamente limitadas por aristas, luego tienen un área infinita.

Como puede verse, el Diagrama de Voronoï es único para un conjunto dado de puntos. Sin embargo existe un caso particular que dificulta las definiciones y hace más complicado el manejo geométrico en la determinación y el análisis del Diagrama de Voronoï; este caso se presenta cuando cuatro o más puntos del conjunto asociado al Diagrama de Voronoï son cocirculares. Este inconveniente es de tal *incomodidad* que algunos autores establecen el supuesto de que en el conjunto de puntos sobre el que se realizan las definiciones no se presentará dicho caso antes de introducir las propiedades del Diagrama de Voronoï<sup>6</sup>.

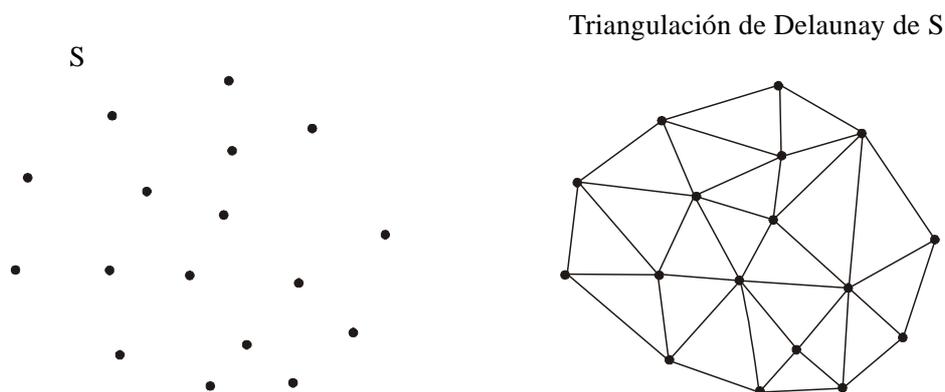
Es necesario hacer una distinción entre dos nociones encontradas en la literatura de la geometría computacional: *los diagramas de Voronoï de orden superior* y *los diagramas de Voronoï en dimensiones de orden superior*. Los primeros hacen referencia a estructuras basadas en el concepto de Diagrama de Voronoï en el plano, tal como se expuso anteriormente, en los cuales las relaciones de vecindad entre lugares geométricos han sido modificadas para trabajar con subconjuntos de puntos y no solo con puntos individuales; . Los segundos corresponden a la generalización del concepto de proximidad en dimensiones mayores a 2. Los elementos que conforman el diagrama de Voronoï en un espacio  $n$ -dimensional son polítopos, de orden  $n$  cuyos límites son *facetas* convexas (ver sección 3.1.2) definidas de manera similar a las aristas que parten de la construcción de la bisectriz a la arista que une al punto asociado al elemento con los demás puntos del conjunto; es de anotar que la generalización del Diagrama de Voronoï en más de dos dimensiones resulta poco eficiente debido al incremento exponencial de la complejidad de la estructura.

Entre las estructuras geométricas que han recibido mayor interés por parte de investigadores en los campos de la geometría computacional y los algoritmos en los últimos años es la Triangulación de Delaunay asociada a un conjunto dado de puntos en el espacio. La construcción de Triangulación de Delaunay puede ser interpretada como el problema dual de la construcción del Diagrama de Voronoï, ya que las aristas de los triángulos pertenecientes a la triangulación de un conjunto dado de puntos tienen una correspondencia de uno a uno con las aristas que conforman los elementos del Diagrama de Voronoï del

---

<sup>6</sup> Como se verá más adelante, el caso de cuatro o más puntos cocirculares afecta también la construcción de la Triangulación de Delaunay.

conjunto, y siempre son ortogonales y parten a cada arista de Voronoï en dos segmentos iguales. De esta forma la construcción del Diagrama de Voronoï a partir de la Triangulación de Delaunay (y viceversa) es inmediata. Sin embargo, la Triangulación de Delaunay de un conjunto de puntos puede ser definida por si misma, según el criterio para la generación de triángulos establecido por el autor que le dio el nombre a esta estructura (ver sección 3.2.1).



**Figura 3.7.** Triangulación de Delaunay en dos dimensiones. Se empleo el mismo conjunto de la Figura n.n.

La figura n.m muestra la triangulación de Delaunay de un conjunto de puntos en 2 dimensiones. Al igual que el diagrama de Voronoï, la Triangulación de Delaunay es única y genera un cubrimiento espacial completo, aunque en este caso restringido a la envoltura convexa del conjunto de puntos sobre los que se ha efectuado el proceso de triangulación. El problema de los puntos cocirculares también afecta a la definición de las propiedades de la Triangulación de Delaunay, ya que ésta deja de ser única para un conjunto dado si este contiene por lo menos una región en la que se encuentren cuatro o más puntos cocirculares. En la siguiente sección se hace un análisis más profundo de los aspectos fundamentales de la Triangulación de Delaunay, además de hacer una breve exposición de algunos de los métodos más conocidos para su construcción, dado que ésta fue seleccionada para la generación automática de las mallas poligonales que empleadas para el despliegue en pantalla de la superficie de los objetos tridimensionales reconstruidos por el sistema.

### 3.2. LA TRIANGULACIÓN DE DELAUNAY

El estudio de Voronoï sobre los polígonos convexos basados en las relaciones de vecindad entre lugares del espacio generó una serie de trabajos en el campo de la geometría, entre los

cuales se contaba el artículo del matemático Ruso Boris N. Delaunay de 1934 titulado *Sur la sphère vide* (sobre la esfera vacía) el cual postulaba que a partir del Diagrama de Voronoï sobre una nube de puntos en el plano se puede encontrar, por dualidad, una triangulación con características especiales. Tales características permiten diferenciar la Triangulación de Delaunay de las demás posibles triangulaciones sobre el mismo conjunto de puntos. El nombre ‘triangulación’ se debe a que originalmente el problema fue analizado en el caso bidimensional, si bien es aplicado indistintamente a cualquier dimensión, aunque los elementos que conforman la *triangulación n*-dimensional no sean triángulos sino simplexes en  $R^n$ . Algunas de las propiedades más relevantes de la Triangulación de Delaunay son:

- Dado una nube de puntos en el espacio, su Triangulación de Delaunay es única<sup>7</sup>.
- Los límites externos de la triangulación de Delaunay conforman a su vez los límites de la envoltura convexa del conjunto de puntos.
- El resultado de determinar el dual geométrico de la Triangulación de Delaunay asociada a un conjunto de puntos en el espacio es el Diagrama de Voronoï del mismo conjunto
- La esfera circunscrita a cada simplexe de la Triangulación de Delaunay asociada a un conjunto de puntos  $S$  no contiene a ningún otro punto perteneciente a  $S$ .

Esta última propiedad, conocida como el ‘Lema General de Delaunay’, es un criterio usado directa o indirectamente por los algoritmos para la construcción de la Triangulación de Delaunay, ya que permite decidir si un simplexe formado por  $n + 1$  puntos de un conjunto en  $R^n$  es un elemento válido de su Triangulación de Delaunay (sección 3.2.1). El establecimiento de un criterio sólido para la construcción de triangulaciones propuesto por Delaunay ha facilitado la implementación de algoritmos para la construcción de enmallados triangulares; por ésta razón la construcción de la Triangulación de Delaunay fue seleccionada como método de generación de mallas poligonales aplicado en la herramienta de software que permite generar un modelo geométrico del objeto digitalizado.

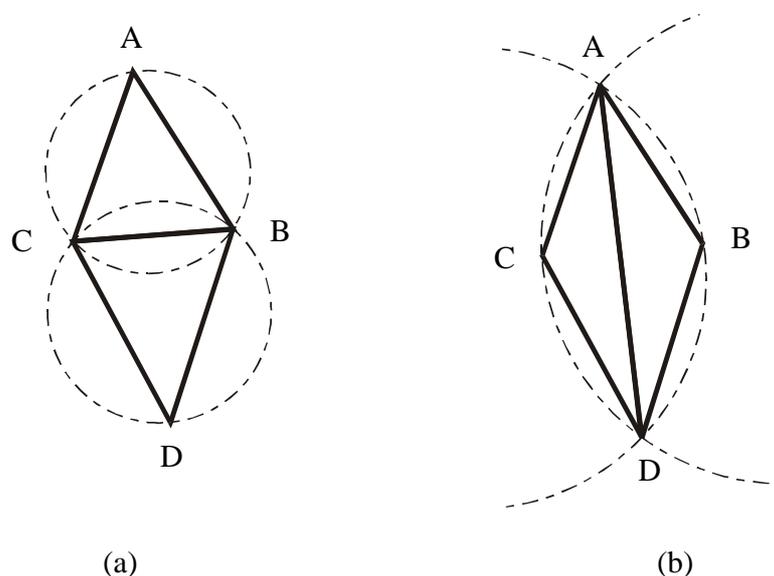
---

<sup>7</sup> Exceptuando el caso de los puntos coesféricos (sección 3.1.6).

### 3.2.1. Lema de Delaunay

El Lema General de Delaunay, llamado originalmente por su autor como *criterio de la esfera vacía*, establece el principio que permite decidir si un simplexe conformado por  $d+1$  puntos pertenecientes a un conjunto determinado de puntos del espacio  $d$ -dimensional es un elemento válido de la Triangulación de Delaunay asociada a dicho conjunto. Para comprender el Lema General de Delaunay es necesario definir el concepto de esfera circunscrita: una esfera circunscrita es una hiper-esfera asociada a un simplexe de tal forma que ésta pasa exactamente por todos los vértices del simplexe. A cualquier simplexe se puede asociar una única esfera circunscrita, aunque esta afirmación es algo difusa cuando se trata con casos degenerados<sup>8</sup>.

Dado un conjunto de puntos en el espacio  $S$ , el Lema General de Delaunay establece que un simplexe determinado es un elemento válido de la Triangulación de Delaunay de  $S$  si y solo si se cumple la condición de que ningún punto del conjunto se encuentra al interior de la esfera circunscrita asociada al simplexe en cuestión, lo que equivale a decir que ésta se encuentra *vacía*.

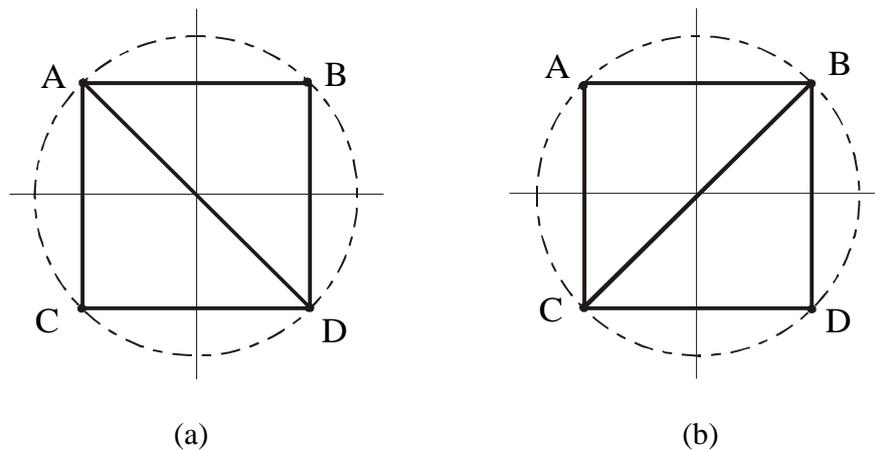


**Figura 3.8.** Criterio de la esfera vacía en dos dimensiones. En (a) los triángulos ACB y BCD cumplen con el Lema General de Delaunay, mientras que los triángulos ADB y DCA violan dicho criterio.

<sup>8</sup> Aquí el término ‘casos degenerados’ se refiere nuevamente al problema de los puntos coesféricos.

La Figura 3.8 muestra la aplicación del Lema general de Delaunay para las dos posibles triangulaciones de un conjunto sencillo de puntos bidimensionales  $S=\{A, B, C, D\}$ . La triangulación mostrada en (a) es la Triangulación de Delaunay del conjunto  $S$ , ya que el punto  $A$  se encuentra por fuera de la esfera circunscrita asociada al triángulo  $BCD$  y, a la par,  $D$  es exterior a la esfera circunscrita de  $ACB$ . Mientras tanto, la triangulación en (b) viola el Lema General de Delaunay, puesto que el punto  $C$  está contenido en la esfera circunscrita de  $ADB$  y simultáneamente  $B$  se encuentra dentro de la esfera circunscrita de  $DCA$ , luego no es la Triangulación de Delaunay  $S$ .

Cuando se presenta el problema de cuatro o más puntos cocirculares, el Lema General de Delaunay no permite decidir cuales simplexes conforman un cubrimiento local válido pues existen varias configuraciones diferentes que cumplen con el criterio de la esfera vacía (Figura 3.9). La solución a este problema es de tipo meramente práctico y está implícita en cada algoritmo utilizado para construir la triangulación.



**Figura 3.9.** Problema de los puntos cocirculares. Tanto los triángulos  $ADB$  y  $CDA$  mostrados en (a), como  $ACB$  y  $CDB$  de (b) cumplen el Lema General de Delaunay para el conjunto  $S = \{A, B, C, D\}$ . Nótese que el dual de las triangulaciones de (a) y de (b) coincide y tiene una arista de longitud 0 en la intersección de las bisectrices a los segmentos  $\overline{AB}$ ,  $\overline{AC}$ ,  $\overline{BD}$  y  $\overline{CD}$  que corresponde a la bisectriz de la arista  $\overline{AD}$  en (a) y  $\overline{CB}$  en (b).

### 3.2.2. Algoritmos para la Construcción de la Triangulación de Delaunay

La primera solución al problema de la construcción de la Triangulación de Delaunay en términos computacionales fundamentada en el Lema General de Delaunay fue planteada por Watson en su artículo de 1981 titulado *Computing the Delaunay tessellation with Application to Voronoi Polytopes*. La solución propuesta por Watson introduce el concepto

de inserción, el cual es el pilar para la construcción óptima de la Triangulación de Delaunay. La solución propuesta por Watson fue adoptada por la mayoría de los autores para su optimización y aplicación.

Posteriormente diferentes investigaciones se han dedicado a la explotación de nuevos algoritmos óptimos que permitan la construcción de la Triangulación de Delaunay de manera eficiente incluso para casos específicos y condiciones del *peor caso*, aunque en las aplicaciones prácticas estas situaciones son excepcionales y el Algoritmo Incremental, con sus variaciones, sigue siendo el más usado.

### 3.2.2.1. Fuerza Bruta

La triangulación de Delaunay para un conjunto de puntos en el espacio  $S$  se puede obtener por fuerza bruta mediante el recorrido de cada subconjunto posible de puntos que genere un simplexe válido en  $R^n$ . Si el simplexe generado por el conjunto cumple el criterio de Delaunay para cada uno de los demás puntos de  $S$  entonces éste es un elemento válido de la triangulación de  $S$  y es almacenado, de lo contrario es descartado. Este método es utilizado en varios artículos principalmente con fines didácticos y para el estudio comparativo de su desempeño frente a otros métodos.

La Triangulación de Delaunay por fuerza bruta en el plano para un conjunto de  $N$  puntos se puede construir en  $O(N^4)$ , aunque en espacios de dimensiones superiores el exponente de  $N$  puede ser aún más grande. Por tal razón, el planteamiento de un algoritmo que permita la construcción óptima de la Triangulación de Delaunay ha sido motivo de una gran variedad de trabajos con múltiples enfoques, cuya característica común es la utilización de diversas estructuras de datos que permitan la reducción de la complejidad de este problema.

### 3.2.2.2. Algoritmo Incremental

El Algoritmo Incremental para un conjunto de puntos  $S$  en dos dimensiones es el siguiente: se toman tres puntos iniciales<sup>9</sup> de  $S$  y se genera un triángulo inicial, al cual se determina su

---

<sup>9</sup> Existe la recomendación de tomar los puntos iniciales en los extremos de la nube de puntos, lo cual garantiza que la esfera circunscrita del triángulo (simplexe) inicial contenga un número significativo de los puntos de la nube

esfera circunscrita. Los demás puntos son *insertados* uno a uno en la triangulación, llamando  $p_i$  al  $i$ -ésimo punto insertado y  $T_i$  a la triangulación al momento de la  $i$ -ésima inserción. En la  $i$ -ésima inserción, si  $p_i$  se encuentra al interior de  $T_i$  se verifica el criterio de la esfera vacía para cada triángulo  $t \in T_i$ ; si la esfera circunscrita de  $t$  contiene a  $p_i$ ,  $t$  es eliminado y sus vértices son adicionados a una lista de puntos que se utiliza para generar una nueva triangulación local  $T_s$ , obtenida obtiene mediante la creación de los triángulos formados por los puntos de la lista que tienen como vértice común a  $p_i$ . De este modo

$$T_{i+1} = T_i - \{t \mid p_i \in \text{esfera\_circunscrita}(t)\} \cup T_s$$

Sí  $p_i$  se encuentra fuera de la triangulación se efectúa un proceso similar al anterior para la eliminación de los triángulos cuya esfera circunscrita contiene a  $p_i$  pero, al generar la lista de puntos para la triangulación local, además de añadir los vértices de los triángulos eliminados, se insertan en dicha lista todos los puntos de  $T_i$  que son visibles a  $p_i$  para luego generar  $T_s$ .

### 3.2.2.3. Otros métodos

Aunque el Algoritmo Incremental provee una solución satisfactoria para la construcción óptima de la Triangulación de Delaunay, el principal tópico abordado por los trabajos más recientes referidos al estudio de esta estructura es el de la elaboración de nuevos procedimientos que conduzcan a la construcción de la Triangulación de Delaunay con un orden de complejidad cada vez menor.

Los intentos por reducir la complejidad de la construcción de la Triangulación de Delaunay con base en el enfoque propuesto por Watson se fundamentan en la utilización de nuevas estructuras de datos que hagan más eficiente el proceso de inserción y reduzcan el número de simplexes de deben ser visitados para verificación del criterio de Delaunay, con el consecuente incremento en las exigencias de memoria del algoritmo. El Algoritmo Incremental Reducido es una de las versiones más sencillas de este tipo de enfoques, eliminando los casos de inserciones al exterior de la triangulación actual mediante la utilización de un simplexe inicial, añadiendo sus vértices al conjunto original, de tal forma que contenga la totalidad de los puntos del conjunto. De esta forma se obtiene la triangulación del simplexe inicial, y la triangulación del conjunto original se consigue

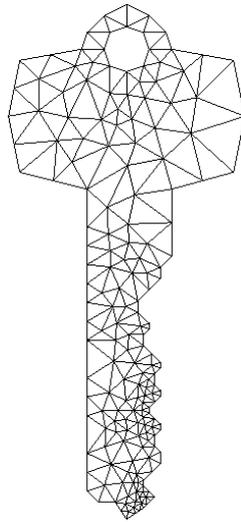
mediante la eliminación de los simplexes de la triangulación final tales que al menos uno de sus vértices pertenezca al simplexe inicial. Sin embargo se debe anotar que los límites externos de la triangulación resultante no constituyen necesariamente la envoltura convexa del conjunto original.

La mayor parte de los algoritmos para la construcción de la Triangulación de Delaunay que no utilizan el concepto de inserción están restringidos al caso bidimensional. Esto sucede con técnicas como el algoritmo de barrido, el método por inversión de aristas y aquellos basados en técnicas generales como *divide y conquista*, entre otros.

### **3.2.3. Triangulaciones Restringidas**

Como se mencionó anteriormente, uno de los resultados obtenidos después de generar la Triangulación de Delaunay sobre un conjunto determinado de puntos en el espacio  $n$ -dimensional es que los límites exteriores de la triangulación conforman simultáneamente la envoltura convexa del conjunto. Sin embargo existen múltiples aplicaciones prácticas en 2 y 3 dimensiones que requieren la determinación de la triangulación de Delaunay sobre una nube de puntos en el espacio con la condición de que los simplexes generados por el algoritmo de triangulación se encuentren dentro del dominio de una región del espacio que no necesariamente coincide con la envoltura convexa del conjunto.

En este contexto surge el concepto de triangulación restringida. En términos generales, una triangulación restringida es aquella cuyo dominio para la generación de simplexes está limitado por un conjunto de restricciones definidas mediante elementos geométricos de orden inferior al espacio al que pertenece el conjunto. Debe tenerse en cuenta que no cualquier conjunto de restricciones es admisible para la generación de una triangulación válida. Un criterio para decidir si un conjunto dado de restricciones es admisible es que todas las esferas construidas de tal forma que cada una de ellas tenga a cada arista del conjunto de restricciones como diámetro, se encuentren vacías.



**Figura 3.10.** Triangulación restringida.

El enfoque común para construir una triangulación restringida de una nube de puntos es el de obtener la Triangulación de Delaunay normal mediante algún método tradicional y luego tratar de adaptar la triangulación mediante modificaciones locales para que cumpla con la restricción de dominio establecida por el conjunto de restricciones, habiendo previamente verificado si dicho conjunto es admisible

#### **4. MODELAMIENTO GRÁFICO TRIDIMENSIONAL**

La posibilidad de interacción del usuario con los computadores de manera amigable, a partir de la implementación de una interfaz gráfica atractiva y eficiente, surgió hace relativamente pocos años. De hecho, la aparición de los primeros entornos operativos en los que la funcionalidad de las aplicaciones se fundamenta en una interacción con el usuario a partir de elementos gráficos, como botones, menús desplegables y múltiples aplicaciones con sus respectivas áreas de trabajo establecidas independiente y flexiblemente sobre la pantalla, se remonta tan solo a los últimos diez o quince años. Sin embargo, a partir de ese momento la evolución de las diferentes tecnologías que permiten la implementación de tales entornos se ha desarrollado aceleradamente, permitiendo un notable incremento en su calidad y eficiencia.

Estas ventajas han sido aprovechadas para extender aún más las posibilidades que brinda el computador para su aplicación en la sistematización y optimización de procedimientos que requieren un elevado grado de precisión. La utilización alternativa de las facilidades gráficas para la presentación de resultados, el despliegue de la información en pantalla y el diseño en sus más diversas expresiones, convirtió a la graficación por computador en uno de los campos de mayor atractivo dentro de las ciencias de la computación.

Brindar cierto grado de realismo a las imágenes generadas por computador ha sido uno de los principales objetivos de los estudios recientes en computación gráfica, debido a la pesada carga computacional, que implica la enorme complejidad del modelamiento de los objetos y fenómenos de la naturaleza. Este problema se ha podido solucionar gracias al impresionante aumento de la velocidad de procesamiento asociada a la interfaz gráfica de los computadores, paralelamente a las investigaciones para la elaboración de algoritmos de óptimos de despliegue en pantalla.

El objeto de estudio de la graficación por computador consiste precisamente en la determinación de procedimientos y estructuras de datos que modelen satisfactoriamente los objetos de la realidad, partiendo de una abstracción de su geometría y de la simulación de los fenómenos ópticos que sobre ellos ocurren.

#### **4.1. TRAZO EN COMPUTADOR DE PRIMITIVAS GEOMÉTRICAS**

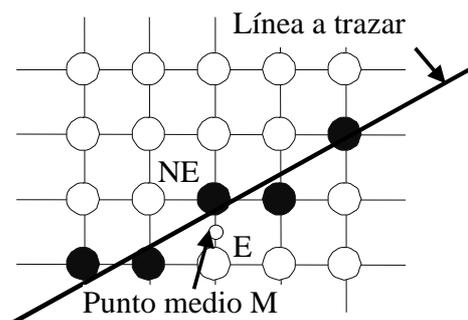
Uno de los primeros aspectos a ser determinados cuando se desea dibujar sobre la pantalla del computador, es decidir como se puede representar el objeto a desplegar, generalmente continuo y complejo, en la restringida retícula de pixeles cuadrados de la pantalla. Es entonces cuando el trabajo conjugado de creatividad y matemáticas se hace indispensable, más aún cuando a semejante restricción se añade la necesidad de rapidez y optimización en el desempeño del procedimiento, debido al elevado número de cálculos para una representación aceptable. Sobra decir que ésta última necesidad se hace aún más crítica cuando se realizan tareas de animación, máxime si esta animación se efectúa en tiempo real. La discretización del espacio es la forma más usada para tratar este problema, aunque por sí sola otorga resultados pobres cuando la resolución de la pantalla se reduce. Otra de las tareas básicas es el relleno de los objetos discretizados, que bien pueden ser polígonos conformados por líneas rectas, o bien objetos más complejos como curvas paramétricas o cónicas. Las etapas anteriormente descritas constituyen los procesos elementales del dibujo en pantalla, indispensables para la implementación de tareas más complejas de la computación gráfica, en nuestro caso el modelamiento tridimensional.

##### **4.1.1. Discretización de Primitivas Geométricas**

Una de las características fundamentales de los computadores es su naturaleza esencialmente discreta. Esto se hace evidente no sólo en la forma de realizar los cálculos aritméticos, sino en la forma como los datos son presentados al usuario, en las diferentes formas de salida de datos. Este carácter impone uno de los primeros retos cuando se emprende la tarea de la elaboración de procedimientos para la presentación de modelos gráficos en la pantalla del computador. El enfoque obvio que se asume para solucionar este problema es decidir cuales elementos de la cuadrícula se aproximan más al recorrido ideal de la primitiva gráfica mediante algún criterio, de tal forma que los pixeles activados en pantalla por el trazo de la

figura den la impresión de representarla convincentemente, generalmente modelada por alguna ecuación.

El Algoritmo de Bresenham es en la actualidad el más conocido, utilizado y eficiente para el trazado de primitivas gráficas. Su esencia consiste en imaginar siempre un punto medio entre dos pixeles por los cuales atraviesa el contorno de la figura a trazar y determinar si dicho contorno cae dentro de la mitad superior definida por el pixel superior y el punto medio (en cuyo caso se activará el pixel superior), o bien si cae dentro de la mitad inferior (activando obviamente el pixel inferior). La Figura 4.1 muestra la aplicación del criterio del punto medio a la discretización de una recta. En la aplicación realizada, la primitiva sobre la cual se realizó el diseño es la línea recta a partir de la cual se conforma el elemento básico de la aproximación de superficies tridimensionales: el triángulo.



**Figura 4.1.** Criterio del punto medio. La malla imaginaria de píxeles es atravesada por la línea ideal a trazar. El próximo píxel activo se escoge entre los píxeles E y NE según sea el más cercano a la intersección entre la malla y la línea ideal.

El algoritmo de Bresenham para el trazo de líneas rectas supone en principio que la pendiente de la recta a dibujar se encuentra entre 0 y 1, ya que las demás pendientes pueden representarse por la reflexión de una línea en este octante del plano. Luego, se selecciona el punto inicial, generalmente el que tenga una menor coordenada x y se empieza a evaluar la distancia entre el punto de intersección de la línea recta ideal con los elementos discretos de la malla, en cuyas intersecciones se encontrarían los píxeles. El siguiente píxel a ser activado para representar a la recta se escoge entre el píxel a la derecha (píxel E) y el píxel arriba a la derecha del actual (píxel NE). El criterio de decisión determina cual de los dos segmentos formados por el punto medio entre los píxeles E y NE fue atravesado por la recta ideal, para activar el píxel que conforma dicho segmento junto al punto medio. Para optimizar los

cálculos correspondientes a la ecuación de la línea recta, acelerando el algoritmo, es necesario efectuar un muestreo correspondiente a los valores enteros y utilizar una ecuación que minimice el número de operaciones aritméticas para el cálculo de la coordenada y dado un conjunto secuencial de coordenadas  $x$ . Para la coordenada  $k$ -ésima la ecuación de la recta se podría escribir como  $y_k = m x_k + b$ , entonces para la siguiente muestra entera se tendrá que  $y_{k+1} = m (x_{k+1}) + b$ . Un nuevo parámetro de decisión  $D$  se introduce con el objeto de minimizar el número de multiplicaciones y divisiones.  $D$  se puede definir a partir de la ecuación fundamental de la línea recta mencionada antes, como  $D = \frac{\Delta y}{2} - \Delta x$ . Si para el  $k$ -ésimo paso del algoritmo el parámetro  $D$  es menor que 0, quiere decir que la recta ideal pasa por el segmento correspondiente al pixel E, por lo que este debe ser activado; de lo contrario, el pixel activado será NE.

La discretización de otras primitivas geométricas planas, como las curvas de sección cónica pueden ser implementadas fácilmente siguiendo un criterio similar al expuesto para la línea recta, simplemente escogiendo el parámetro de decisión adecuado.

Una posible modificación a este algoritmo permite la aplicación de las diferentes técnicas de despliegue del modelo tridimensional en un buffer fuera de pantalla. Así, los pixeles que el algoritmo de Bresenham activa no son pintados directamente sobre la pantalla, sino que se almacenan en una lista junto con los pixeles de las demás líneas que conforman un polígono (triángulo) dado, para que posteriormente el algoritmo de despliegue consulte dicha lista con el objeto de determinar los límites espaciales del polígono que se dispone a trazar sobre el área de dibujo.

#### 4.1.2. Eliminación de Artefactos de Discretización (Antialiasing)



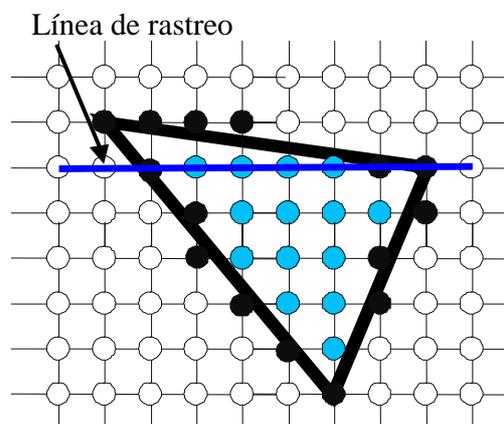
**Figura 4.2.** Problema de *aliasing*. Imagen obtenida de la aplicación desarrollada. Nótese el efecto de ‘escalera’ en el contorno de la imagen

El alisado de líneas y otras primitivas geométricas, conocido con el término inglés de *antialiasing*, consiste en hacer menos evidente los problemas generados por el muestreo discreto (*aliasing*). Su objeto es tratar de ocultar o disminuir, mediante efectos ópticos, la impresión de sierra, ocasionado por pixeles irregularmente dispuestos a lo largo del contorno de las figuras (Figura 4.2), así como asegurar la impresión de grosor constante para líneas definidas con un mismo espesor pero distinta inclinación respecto de la horizontal del dispositivo. Esto se logra valiéndose de cambios en la intensidad de los colores y gradaciones de unos colores hacia otros en las fronteras de los objetos.

La técnica más conocida para modelar el *antialiasing* es hacer una división imaginaria de cada pixel atravesado por la línea en varios *subpixeles*, lo cual equivale a un incremento virtual de la resolución de la pantalla, y analizar cuantos de ellos son realmente atravesados por la recta ideal. La razón entre el número de *subpixeles* que son atravesados por la recta ideal y el total que conforma un pixel es un criterio para asignar su intensidad relativa respecto al color ideal asignado a la línea trazada. Debe recordarse que dicha intensidad relativa no puede calcularse directamente sobre el color de la línea, sino que debe ser determinada para cada componente del modelo de color que se esté utilizando; esto último se describirá más adelante.

### 4.1.3. Relleno de Polígonos

La tarea de rellenar polígonos puede ser tan sencilla o compleja como se quiera, dependiendo de las características de los polígonos que la aplicación sea capaz de manipular, tanto como de las necesidades y objetivos de ésta. Por ejemplo, los programas dedicados al tratamiento de objetos bidimensionales deben hacer un mayor énfasis en el proceso de relleno de polígonos, teniendo en cuenta las distintas condiciones de cada polígono, como la concavidad, los polígonos con orificios internos, con aristas paralelas a los ejes (verticales u horizontales), e incluso con características anómalas como la *autointersección*, que en circunstancias específicas pueden considerarse válidas.



**Figura 4.3.** Relleno de polígonos mediante línea de rastreo

La manera más conocida para el relleno de polígonos bidimensionales consiste en almacenar una tabla de aristas activas (aquellas que corresponden al polígono que debe ser rellenado) y hacer un barrido vertical en un buffer fuera de pantalla con una línea de rastreo horizontal (Figura 4.3) para determinar las intersecciones de dicha línea con las aristas actualmente activas. Dichas intersecciones activan o no una bandera de paridad, la cual determina entre que puntos de intersección debe ser trazado un segmento de relleno horizontal, lo que equivale a activar con el color de relleno una secuencia de píxeles que va desde el punto en que la bandera de paridad se hace impar hasta aquel en el cual vuelve a un valor par.

En el caso tridimensional el relleno de polígonos suele ser mucho más abreviado, ya que los objetos tridimensionales generalmente son modelados como conjuntos disyuntos y cerrados de polígonos convexos con un número fijo de aristas (casi siempre triángulos, ocasionalmente cuadriláteros, difícilmente polígonos de orden superior) lo cual hace mucho más sencillo el

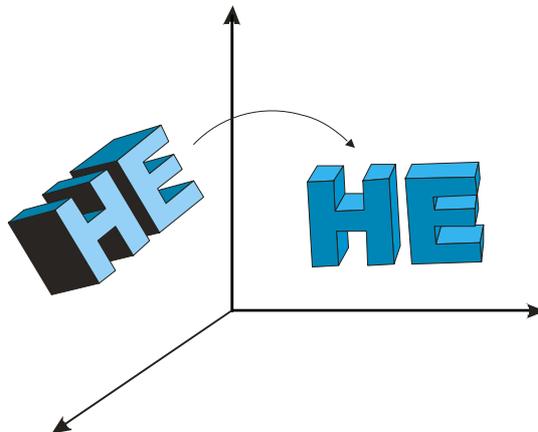
proceso de relleno para cada polígono de un determinado objeto. Es así como en la gran mayoría de los casos, las aplicaciones de modelado tridimensional evitan la implementación de complejas rutinas de relleno de polígonos, relegando esta tarea a ser un subproceso implícito dentro de la detección de superficies visibles en el trabajo de despliegue en pantalla (*render*). El relleno se convierte así en una rutina simple de trazo de líneas horizontales entre las dos coordenadas  $x$  extremas para una coordenada  $y$  dada de la proyección del polígono en un buffer fuera de pantalla a la vez que son evaluadas otras variables de presentación como la iluminación, el sombreado y la textura del polígono a rellenar. Una idea de cómo puede ser implementado un procedimiento de relleno de polígonos se presenta en la sección correspondiente al despliegue en pantalla, junto con las demás tareas de despliegue.

## **4.2. MODELAMIENTO DE LA GEOMETRIA TRIDIMENSIONAL**

Uno de los campos de las ciencias básicas a las que más se recurre por parte de la computación gráfica es sin duda el Álgebra Lineal, debido a que ésta provee de una serie de técnicas y recursos matemáticos muy útiles para la cuantificación de la posición y tamaño relativo de objetos en un espacio arbitrario  $\mathbf{R}^n$ , así como para la manipulación numérica eficiente de tales variables por medio del cálculo matricial. Un uso apropiado del Álgebra Lineal puede conducir a una implementación óptima de las diferentes tareas involucradas en la ubicación de los objetos a modelar en el espacio de coordenadas generado por computador.

### **4.2.1. Transformaciones Geométricas aplicadas al Modelamiento Tridimensional.**

Entre los principales atractivos de las aplicaciones de modelamiento tridimensional se encuentra la posibilidad de obtener una representación gráfica de los objetos modelados a partir de un sinnúmero de puntos de vista (en sus respectivas “proyecciones” sobre la pantalla), así como la capacidad de variar de forma interactiva e inmediata la posición tanto del punto de vista, las posiciones relativas de los objetos, y en general todas las características geométricas (tamaño, distancia, ángulo relativo, etc.) de los objetos pertenecientes a una escena dada. Este gran potencial se consigue mediante la implementación y aplicación secuencial de un conjunto relativamente sencillo de Transformaciones Geométricas.



**Figura 4.4.** Ejemplo de la aplicación de transformaciones geométricas en  $\mathbf{R}^3$  sobre un objeto.

En adelante, los puntos vértices de un objeto tridimensional serán tratados como  $k$ -uplas de coordenadas en  $\mathbf{R}^3$ , y las líneas y planos se representarán mediante los coeficientes de sus ecuaciones generales. Por lo tanto, las transformaciones geométricas serán implementadas como operaciones matriciales. Vale la pena resaltar que las operaciones geométricas que se aplican a un objeto tridimensional se efectúan directamente sobre los vértices pertenecientes a cada uno de los polígonos que lo conforman, y a partir de los nuevos puntos obtenidos se calculan nuevamente las ecuaciones de líneas, vectores y planos.

Las transformaciones geométricas más utilizadas por el modelamiento tridimensional son:

#### 4.2.1.1. Translación

Para cambiar la posición de un objeto en el espacio la aproximación obvia corresponde a una simple suma de componentes:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Donde  $t_x$ ,  $t_y$  y  $t_z$  son las cantidades en las que se debe desplazar cada punto en las coordenadas  $x$ ,  $y$  y  $z$  respectivamente.

#### 4.2.1.2. Rotación

La rotación en torno a alguno de los ejes del sistema de coordenadas tridimensional que se expuso en la sección 2.1.3.2. es aplicada de manera análoga en el modelamiento de objetos

tridimensionales, pues si se desea efectuar una rotación sobre un objeto alrededor de un eje del sistema de coordenadas, bastará multiplicar la matriz de transformación correspondiente por cada vértice del objeto.

#### 4.2.1.3. Escalamiento

Para alterar el tamaño del objeto basta con multiplicar cada componente por un factor de escalamiento:

$$x' = x \cdot s_x$$

$$y' = y \cdot s_y$$

$$z' = z \cdot s_z$$

La expresión matricial es fácilmente deducible:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Si se desea conservar las proporciones originales del objeto,  $s_x$ ,  $s_y$  y  $s_z$  deben ser iguales. Otras transformaciones geométricas menos conocidas, como reflexión o estiramiento se implementan como modificaciones de las transformaciones básicas anteriormente descritas

#### 4.2.2. Coordenadas Homogéneas

Una aplicación capaz de manejar simultáneamente múltiples objetos tridimensionales con cierto grado de complejidad debe realizar una gran cantidad de transformaciones muy rápidamente, más aún si se realizan tareas de animación o interactividad en tiempo real, como sucede en los simuladores o juegos de vídeo. Esta exigencia trae consigo la necesidad de hacer más eficientes los cálculos implícitos en las transformaciones geométricas. La solución más conocida es la de tener una única matriz de transformación que se aplica (multiplica) una sola vez a todos los vértices del objeto tridimensional, habiendo preparado previamente dicha matriz para que el resultado de la multiplicación por cada punto sea igual al conjunto de transformaciones que se desea aplicar al objeto tridimensional. Pero la forma de aplicar las transformaciones descritas anteriormente tiene el problema de que la translación se ejecuta mediante una suma vectorial, mientras que la rotación y el escalamiento son logrados

mediante multiplicaciones matriciales. Es entonces cuando se recurre a las Coordenadas Homogéneas, las cuales hacen que cualquier transformación geométrica pueda ser expresada como una multiplicación de matrices. La esencia del sistema de coordenadas homogéneas consiste en agregar una cuarta componente  $h$  a la representación vectorial de los puntos en el espacio tridimensional. Por simplicidad,  $h$  generalmente toma el valor de 1. De este modo, las matrices de transformación son de tamaño  $4 \times 4$ .

Así, las transformaciones básicas expresadas en coordenadas homogéneas son:

Traslación:

$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix}$$

Rotación (ej. rotación sobre el eje  $y$ ):

$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\text{sen } \theta & 0 \\ 0 & 1 & 0 & 0 \\ \text{sen } \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix}$$

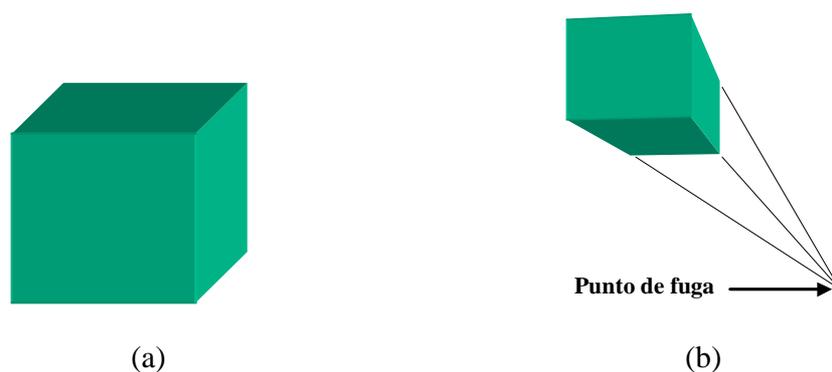
Escalamiento:

$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix}$$

Para efectuar una transformación geométrica compuesta cualquiera, basta con tener una única matriz de transformación principal, siendo originalmente la matriz identidad, a la cual se multiplican sucesivamente las matrices correspondientes a cada una de las transformaciones básicas. La matriz de transformación resultante se aplica sobre cada uno de los vértices del objeto tridimensional. Se debe tener en cuenta el orden en el que se multiplican las matrices de transformación para que el resultado sea el esperado.

### 4.2.3. Proyecciones

Una vez determinada la posición de un objeto en el espacio de coordenadas  $\mathbf{R}^3$  generado por el computador, se define como el objeto es presentado al observador, es decir, como el objeto es proyectado sobre el plano de observación, el cual generalmente corresponde a la pantalla del computador. El plano o superficie de observación es entonces la referencia sobre la cual se proyectan los objetos pertenecientes a una escena. En general, el sistema de coordenadas del plano de observación es diferente al sistema de coordenadas de la escena, comúnmente conocido como *coordenadas mundiales*. Por lo tanto, es necesario realizar una nueva transformación de coordenadas con el propósito de generar una nueva representación geométrica de los objetos que corresponda a lo que debe percibir un observador ubicado en el origen de las coordenadas del plano de observación. Así, se efectúa una nueva serie de transformaciones geométricas de tal forma que, al ser aplicadas, hagan coincidir el origen del sistema de coordenadas mundiales con el origen de coordenadas del plano de observación. De esta forma, de cada vértice de cada objeto de la escena se obtiene una nueva coordenada en el sistema del plano de observación, que corresponde a la proyección paralela u ortogonal de cada vértice (este tipo de proyección es muy usado en el campo del diseño asistido por computador). La forma más sencilla de realizar dicho tipo de proyección consiste en utilizar las componentes de los vértices que corresponden al eje perpendicular al plano de observación sólo para la determinación de superficies visibles (método que se describirá más adelante) e ignorarlas al momento de dibujar en pantalla (Figura 4.5(a)).



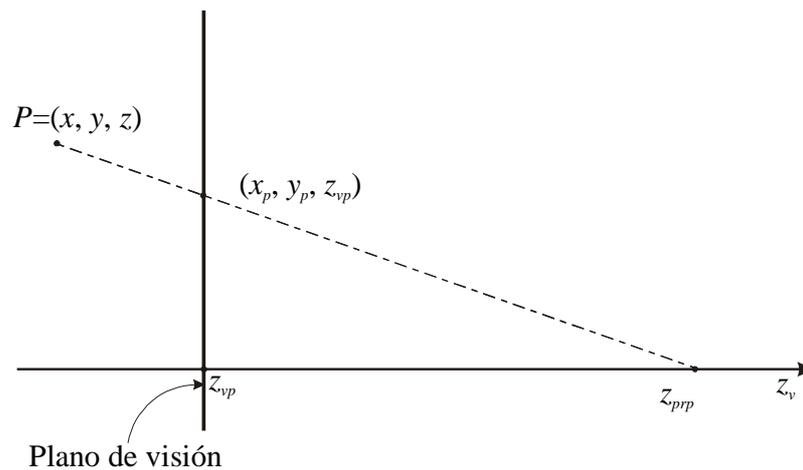
**Figura 4.5.** Proyecciones (a) ortogonal, (b) perspectiva, la cual implica el establecimiento de uno o varios puntos de fuga a los cuales convergen todas las aristas del objeto que no son paralelas al plano de vista.

Sin embargo, el anterior enfoque ignora el fenómeno óptico de la perspectiva(Figura 4.5(b)).

Por experiencia se sabe que entre más alejado se encuentre un objeto de los ojos, más pequeño

se ve. Así mismo, las partes de un objeto que se encuentran más lejanas aparecen más pequeño respecto a las partes más cercanas. La forma de implementación de la proyección en perspectiva es similar a la proyección paralela, excepto que en el momento de proyectar los puntos del sistema de coordenadas del observador al plano de observación se debe efectuar una modificación de las componentes paralelas a dicho plano. Geométricamente el efecto de perspectiva se puede modelar así:

Se definen  $x$  y  $y$  como las componentes paralelas al plano de proyección y  $z$  como la componente perpendicular;  $z_{prp}$  como la coordenada  $z$  del *punto de referencia de proyección* cuyas coordenadas  $x$  y  $y$  son iguales a 0. Se coloca el plano de observación en  $z_{vp}$  y se determina la posición de corte sobre el plano de observación  $(x_p, y_p, z_{vp})$  del rayo que va desde la posición real del vértice  $(x, y, z)$  hasta el punto de referencia de observación (Figura 4.6).



**Figura 4.6.** Obtención de las coordenadas de perspectiva.

De este modo las coordenadas del punto de corte sobre el plano de proyección se calculan como:

$$xp = x \left( \frac{z_{prp} - z_{vp}}{z - z_{prp}} \right) = x \left( \frac{d_p}{z - z_{prp}} \right)$$

$$yp = y \left( \frac{z_{prp} - z_{vp}}{z - z_{prp}} \right) = y \left( \frac{d_p}{z - z_{prp}} \right)$$

donde  $d_p = z_{prp} - z_{vp}$  es la distancia desde el punto de referencia de proyección hasta el plano de observación. Se observa que entre más cercano sea el punto de referencia de proyección al plano de observación, más marcado es el efecto de perspectiva.

Una matriz de transformación posible para la proyección de perspectiva, dado que sólo las coordenadas  $x$  y  $y$  tienen sentido sobre el plano de proyección es:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} d_p / z - z_{prp} & 0 & 0 & 0 \\ 0 & d_p / z - z_{prp} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix}$$

#### 4.2.4. Recorte de Polígonos

Dado que la superficie de la pantalla tan sólo cubre una porción limitada del plano de observación, es muy probable que algunos objetos no alcancen a ser proyectados sobre dicha superficie, por lo no son tenidos en cuenta al momento de dibujar en pantalla. Esto se consigue mediante una simple comparación de coordenadas, ignorando aquellos objetos cuyos vértices se proyecten por fuera del área de dibujo. Un problema más serio aparece cuando la proyección de algunos de los vértices del objeto dado caen dentro de la superficie de dibujo en tanto que otros vértices lo hacen fuera de ella; en este caso sólo deberán ser pintadas algunas de las regiones de ciertos polígonos del objeto. Para ello se emplean los algoritmos de recorte de polígonos, los cuales se fundamentan en la búsqueda de nuevos vértices para los polígonos que son parcialmente proyectados en la pantalla, los cuales coinciden con los puntos de intersección entre las aristas que tienen un vértice dentro y el otro fuera de la superficie de dibujo con los extremos de la pantalla. El recorte se efectúa para cada borde de la pantalla sobre las aristas de cada polígono del objeto, generando en cada paso una nueva lista de vértices que conforman el polígono recortado. Después de aplicar el recorte sobre un polígono dado para cada límite de la superficie de dibujo se genera un nuevo polígono ubicado completamente en la pantalla para su posterior despliegue. Un problema se presenta cuando el algoritmo de despliegue maneja polígonos con un número fijo de vértices, caso en el cual es necesario implementar un algoritmo de recorte que genere una nueva lista de polígonos disyuntos a partir de los nuevos vértices generados, lo cual puede ser una tarea compleja.

## **4. DESPLIEGUE EN PANTALLA (RENDER)**

Los procedimientos de despliegue en pantalla tienen por objeto presentar al usuario en la forma más realista posible una imagen obtenida a partir de los objetos tridimensionales componentes de una escena. Una implementación de despliegue de modelos tridimensionales debe plasmar con cierto grado de rigurosidad los fenómenos ópticos más comunes que se presentan en la realidad debido a las características de la luz y su interacción con los objetos y sus propiedades físicas, buscando el mayor realismo visual limitado por restricciones debidas a las características de los computadores y la necesidad de obtener rápidamente el resultado final.

### **4.3.1. Determinación de Superficies Visibles**

Uno de los procesos fundamentales dentro de una aplicación de despliegue gráfico es determinar cuales de las caras de los objetos pertenecientes a una escena serán dibujados dado que son visibles al observador. La solución presenta dos caminos diferentes, aunque no excluyentes: objeto–espacio e imagen–espacio. Una tercera alternativa combina características de los dos enfoques.

#### **4.3.1.1. Métodos Objeto–Espacio**

Los métodos objeto–espacio se fundamentan en la geometría de los objetos, así como en sus componentes (aristas o polígonos) para determinar su visibilidad. Así, pueden definirse algunas estructuras, como listas o árboles binarios de partición de superficie (BSP) para almacenar la posición de los objetos según su distancia al plano de observación, para que de este modo el algoritmo de despliegue trace en pantalla primero los objetos más distantes al observador y posteriormente los más cercanos (algoritmo del pintor). Este principio entrega buenos resultados solo en circunstancias muy específicas, como escenas con un número reducido de objetos y por lo tanto de superficies, y con la restricción de que las superficies no pueden intersectarse ni traslaparse mutuamente. Cuando la escena consta de un número elevado de polígonos, la estructura que determina su orden puede hacerse demasiado grande y el rendimiento decrece notoriamente.

#### 4.3.1.2. Métodos Imagen–Espacio

Una perspectiva distinta a la anterior da una mayor importancia a la superficie de dibujo como criterio para decidir qué dibujar. En los métodos imagen–espacio se evalúa la profundidad de las superficies que se proyectan sobre cada elemento (pixel) de la superficie de dibujo. Así, al final, el pixel en cuestión se activa con el color de la superficie más próxima al plano de proyección en ese punto. El desempeño de los métodos imagen–espacio es generalmente constante e independiente del número de polígonos, ya que entre más grande sea este número, menor superficie ocupará cada polígono (esto es cierto una vez se ha implementado el recorte). Así, la única tarea compleja en este tipo de métodos es determinar la distancia de un punto dado sobre un polígono a la superficie de dibujo.

El cálculo de la componente perpendicular a la superficie de observación se puede implementar de varias formas. Una opción es utilizar la ecuación del plano, cuyos coeficientes se pueden extraer de la ecuación de una recta normal al polígono, mediante el producto cruz de la diferencia de la representación vectorial de sus vértices, tomados en el sentido contrario de las manecillas del reloj. Una vez se han determinado los coeficientes de la ecuación del plano, la componente perpendicular, generalmente notada como  $z$ , se calcula mediante despeje desde dicha ecuación:

$$z = \frac{-Ax - By - D}{C}$$

Donde  $x$  y  $y$  corresponden a las coordenadas del pixel sobre el plano de proyección.

Otro método para determinar el valor de las coordenadas  $z$  interiores al polígono es interpolar linealmente los valores de  $z$  en los puntos de las aristas a partir de las componentes  $z$  de los vértices; así para cada punto  $i$ -ésimo de la arista que une los puntos 0 y 1:

$$z_i = kz_0 + (1 - k)z_1$$

Donde  $k$  se calcula a partir de la variación de las coordenada  $y$  entre los vértices:

$$k = (z_1 - z_0) \frac{y_i - y_0}{y_1 - y_0}$$

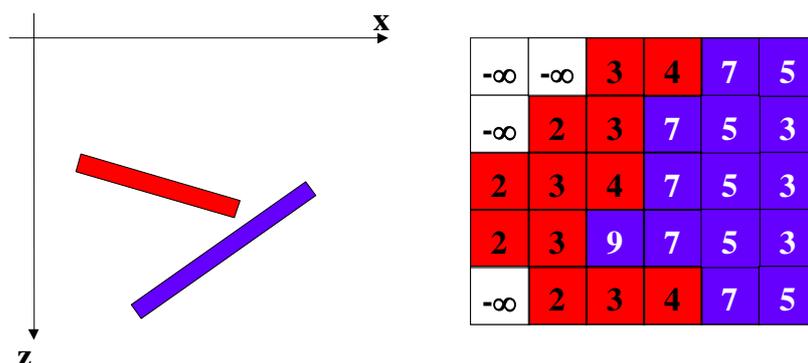
El cálculo de la profundidad para los puntos interiores se efectúa de manera similar, interpolando entre los extremos de una línea de rastreo horizontal para cada coordenada y en el rango del polígono. El nuevo valor de  $k$  está dado por:

$$k = (z_1 - z_0) \frac{x_i - x_0}{x_1 - x_0}$$

Los principales algoritmos basados en el principio imagen-espacio son el de la línea de rastreo y el de la memoria de profundidad.

**Línea de Rastreo:** Este algoritmo recorre línea por línea la totalidad de la superficie de dibujo, evaluando para cada punto sobre la línea el valor de profundidad de todas las superficies de la escena que se proyecten directamente sobre el pixel correspondiente; el pixel se activará con el color de la superficie más cercana al plano de observación.

**Z-Buffer:** El método de la memoria de profundidad, más conocido como *z-buffering*, es sin duda el más usado para la detección de superficies visibles. En este algoritmo se emplean dos matrices “paralelas”<sup>1</sup>, la matriz de profundidades, o *z-buffer*, inicializada con un valor que indique la ausencia de polígonos, y la matriz de colores, o *color-buffer*, conteniendo inicialmente el valor correspondiente al color del fondo de la escena. Las dos matrices relacionan la profundidad y el color actual de la proyección de los objetos sobre la superficie de observación, y cada elemento representa un pixel de tal superficie.



<sup>1</sup> Esto quiere decir que las dos matrices tienen el mismo tamaño, el cual a su vez es igual a la dimensión del área de dibujo en pantalla.

**Figura 4.7.** Z-Buffering. En la cuadrícula de la derecha se solapan las matrices de color y profundidad resultantes después de aplicar el método de detección de superficies visibles a la proyección del polígonos rojo y el azul (izquierda).

Para cada punto de los polígonos de la escena se evalúa la componente en la dirección  $z$  correspondiente a las coordenadas de cada pixel (Figura 4.7). Esto se logra calculando la componente  $z$  sobre los puntos de las aristas de cada polígono a partir de las coordenadas de sus vértices, y luego se calculan los valores de la componente  $z$  de la superficie de los polígonos utilizando los valores de los puntos en las aristas. Cuando se calculan los valores sobre la superficie de los polígonos se compara dicho valor con el valor actual del  $z$ -buffer para el pixel correspondiente a las coordenadas de la proyección. Si el valor calculado sobre el polígono es más cercano al plano de observación, reemplaza al valor en el  $z$ -buffer, y el color del polígono se escribe sobre la misma posición en el  $color$ -buffer. Cuando se han procesado todos los polígonos el  $color$ -buffer se transfiere totalmente a la pantalla y las dos matrices son re-inicializadas para el siguiente proceso de despliegue. Una de las razones del uso extendido de la memoria de profundidad es que en las aplicaciones dedicadas a la animación interactiva (como los juegos de vídeo), el  $color$ -buffer puede corresponder simultáneamente al  $buffer$  fuera de pantalla empleado para el intercambio que evita el parpadeo.

#### 4.3.1.3. Métodos Mixtos

Con el objeto de optimizar la determinación de las superficies visibles, en muchos casos se utilizan estrategias combinadas objeto-espacio e imagen-espacio, con la intención de eliminar cálculos de profundidad innecesarios para superficies que a priori se sabe no deben ser pintadas. La estrategia más simple es ignorar los polígonos cuyos vértices se encuentran totalmente por fuera de la superficie de dibujo. Otra estrategia es utilizar una lista de profundidad para las superficies de los objetos, con el fin de modelar la transparencia recursiva (descrita adelante) y el *antialiasing*, y luego utilizar el  $z$ -buffering para corregir los problemas de mutuo solapamiento. Un procedimiento muy utilizado es el de verificar el signo de la componente perpendicular del vector normal de la superficie en la dirección del observador; si ésta componente tiene un valor negativo, la superficie estará de espaldas al observador y por lo tanto no deberá ser evaluada por el método imagen-espacio.

### **4.3.2. Modelos de Color**

Los modelos de tratamiento de los colores están íntimamente ligados a la implementación de los dispositivos electrónicos para la presentación de colores, así como las herramientas computacionales para tratar con tablas de colores (paletas). Uno de los principios de la óptica plantea que la tonalidad de una fuente de luz depende de los componentes frecuenciales de los rayos emitidos por ésta. La mayor o menor proporción de una frecuencia dada en un haz de fotones definirá cual será el color percibido. Los colores que el ojo humano puede distinguir – desde la frecuencia más baja hasta la más alta– van del rojo al violeta, sin embargo, todas las tonalidades de este espectro pueden obtenerse mediante la mezcla, en diferentes proporciones, de un número limitado de tonos, conocidos como luces primarias.

#### **4.3.2.1. Modelo RGB (Rojo–Verde–Azul)**

Este modelo es utilizado por los monitores a color, los cuales emplean una retícula de triadas rojo–verde–azul activadas en diferente proporción para la representación de cada color. Consecuentemente, el software que controla dicho hardware se fundamenta en el tratamiento del color como una mezcla de diferentes proporciones (de 0 a 255) de componentes rojo, verde y azul. En el sistema Win32, sobre el cual se desarrolló la aplicación, un color está representado por una variable de tipo COLORREF, equivalente a un DWORD de 32 bits, de los cuales los 8 bits menos significativos representan la cantidad de rojo, los siguientes 8 la cantidad de verde, y los últimos 8 la cantidad de azul. Los restantes 8 bits más significativos generalmente se establecen como 0, aunque algunas veces pueden representar una componente  $\alpha$  (alfa) usada para transparencias en mapas de bits.

#### **4.3.2.2. Modelo CYMK (Cían–Amarillo–Púrpura–Negro)**

A pesar del éxito del modelo RGB, gracias a su simplicidad, es imposible aplicarlo a las técnicas de impresión a color, debido a que las mezclas de tinta o los tramados finos, utilizados para imprimir en colores, presentan resultados muy pobres cuando se emplean los componentes rojo, verde y azul. La intensidad del conjunto RGB difícilmente modelará con propiedad colores claros a partir de mezcla de tintas o tramado. Como solución se planteó el modelo CYMK (Cían–Amarillo–Magenta–Negro) que contiene colores mucho más luminosos (cían, amarillo y magenta), facilitando la impresión de colores claros, en tanto que

la intensidad se puede manejar agregando a la mezcla una cierta cantidad de componente negro.

#### **4.3.2.3. Modelo YIQ**

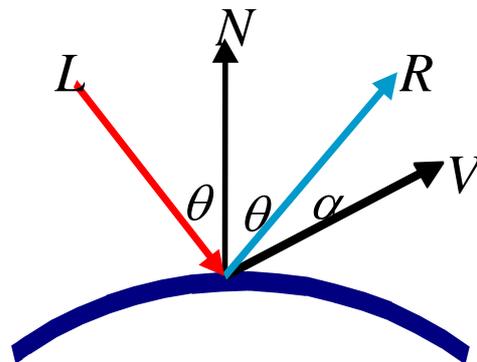
El modelo YIQ, utilizado por el estándar de televisión norteamericano, trata de ser consecuente con la no-linealidad del espectro de luz visible, es decir, la suma lineal de los componentes primarios de dos colores no resulta ser exactamente el color obtenido de la mezcla ideal de estos. En el modelo YIQ la componente Y contiene información sobre la intensidad del color (brillantez), la componente I sobre el naranja y el cian, y la componente Q sobre el verde y el magenta. Cada componente ocupa un ancho de banda diferente, para así simular el comportamiento real de la luz en cuanto a composición y suma.

#### **4.3.2.4. Modelo HSV (Matiz, Saturación, Valor)**

El modelo HSV intenta minimizar la forma de representar un color a partir de las limitaciones de percepción del ojo humano. Sabiendo que tan sólo se pueden percibir en el mejor de los casos 83.000 colores, este modelo prescinde de las componentes de colores “primarios”, y se implementa un método basado en el matiz (Hue), la saturación (Saturation) y el Valor (Value). El matiz representa un ángulo en un hexágono de colores, en cuyas esquinas están el rojo (0°), el amarillo (60°), el verde (120°), el cian (180°), el azul (240°) y el magenta (300°). El valor es una medida de la brillantez del color (0 equivale a negro, 1 a blanco). La saturación es una magnitud que expresa qué tan puro es el color (o qué tan cerca se encuentra de los colores ideales). Con esta representación una gama de 16.384 colores se puede representar con una palabra de 14 bits.

### **4.3.3. Modelos de Iluminación**

El mecanismo elegido para modelar el comportamiento de los objetos frente a la luz es uno de los factores más importantes del proceso de despliegue, ya que de este depende la calidad del resultado final. El modelo de iluminación debe contemplar algunos de los comportamientos físicos más notorios de las superficies reales en interacción con la luz visible. Tradicionalmente se han identificado tres componentes distintos en el resultado final de la iluminación sobre un cuerpo dado: el ambiental, el difuso y el especular. La Figura 4.8 muestra las variables vectoriales principales que dan origen a dichos componentes.



**Figura 4.8.** Iluminación.  $L$  es el vector de incidencia de la luz,  $N$  el vector normal a la superficie,  $R$  el vector de reflexión y  $V$  el vector de observador.

#### 4.3.3.1. Componente Ambiental

El componente ambiental describe la luminosidad del entorno de una escena, es decir, la cantidad de luz que incide sobre los objetos desde todas las direcciones de manera prácticamente uniforme. De este modo, todas las superficies de todos los objetos de una escena recibirán la misma cantidad de luz ambiental, y reflejarán tanta luz como el material del que estén constituidos lo permita. Esto último es modelado mediante un coeficiente de reflectividad ambiental  $k_{a\lambda}$ , diferente para cada material. Así, la intensidad reflejada por un objeto dado en el componente de color  $I_\lambda$  esta dada por la expresión:

$$I_\lambda = I_{a\lambda} k_{a\lambda} O_{a\lambda}$$

Donde  $I_{a\lambda}$  es la intensidad de luz ambiental de la escena, y  $O_{a\lambda}$  es el componente de color ambiental del material propio del objeto. El resultado de aplicar la iluminación ambiental es un conjunto de polígonos coloreados uniformemente.

#### 4.3.3.2. Componente Difuso

El componente difuso simula el resultado de aplicar una fuente de iluminación puntual de cierta intensidad, que es reflejada de manera uniforme (difusa, como en los objetos mate) por la superficie del objeto. Pero la intensidad de la luz reflejada depende de qué tan grande sea la componente del vector normal de la superficie en la dirección de la fuente de luz puntual, es decir, de qué tan parecidos sean el vector normal a la superficie del objeto en un punto determinado y el vector que representa la ubicación de la fuente de luz en el espacio de

coordenadas de la escena. Esto hace que los polígonos de un objeto que se encuentren expuestos más directamente a la fuente de luz se vean más brillantes que los que reciben la iluminación de una manera más sesgada. El siguiente es un modelo de color que implementa la componente difusa y la ambiental:

$$I_{\lambda} = I_{a\lambda} k_{a\lambda} O_{d\lambda} + f_{att} I_{p\lambda} k_d O_{d\lambda} (N \cdot L)$$

El primer sumando representa la componente ambiental. El segundo sumando representa la componente difusa y sus factores son:

- $f_{att}$ , el coeficiente de atenuación, el cual dice que tanto responden los objetos a las fuentes de iluminación puntuales.
- $I_{p\lambda}$ , la intensidad de la fuente de luz en el componente frecuencial  $\lambda$ .
- $k_d$ , el coeficiente de reflexión difusa del material.
- $O_{d\lambda}$ , el color difuso para la componente frecuencial  $\lambda$ .
- $N \cdot L$ , el producto punto de los vectores unitarios  $N$  (normal a la superficie) y  $L$  (dirección de la fuente de luz).

A medida que el ángulo entre los vectores  $N$  y  $L$  se hace más pequeño, su producto punto se acerca a uno, el componente difuso se hace más grande, y el efecto de la iluminación puntual más notorio, pues las intensidades de los polígonos que conforman un objeto varían más drásticamente según el cambio de su vector normal.

#### 4.3.3.3. Componente Especular, Modelo de Phong

El modelo de iluminación de Phong trata de simular el comportamiento de ciertos materiales, cuya superficie regular la hace que la luz sea reflejada de manera uniforme (especular), produciendo un brillo notorio en las partes de su superficie que reflejan la luz en una dirección cercana al vector en la dirección del observador. Este efecto es más apreciable en los objetos metálicos

La solución de Phong añade un sumando al modelo de iluminación difuso, que incrementa la intensidad cuando el coseno del ángulo entre el vector de reflexión y el vector del observador aumenta. Entonces el componente especular se escribe como  $f_{att} I_{p\lambda} k_s O_{s\lambda} \cos^n \alpha$  donde  $\alpha$  es el

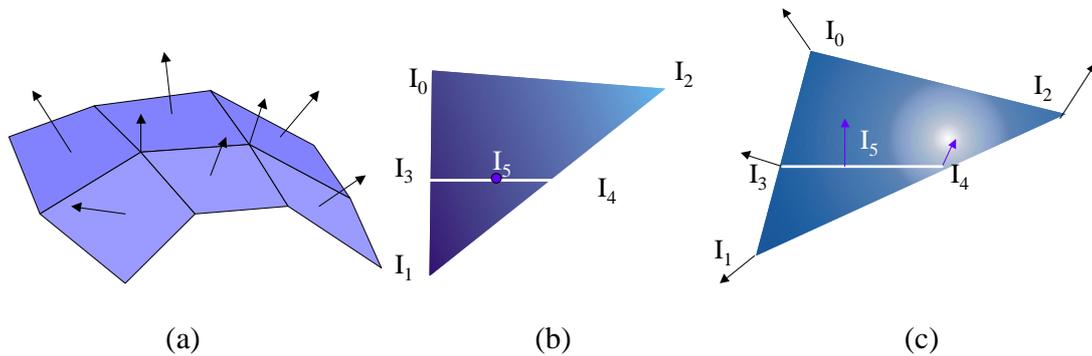
ángulo entre los vectores  $R$  (de reflexión) y  $V$  (de observación),  $k_s$  la constante de reflexión especular del material y  $O_{s\lambda}$  el color especular del objeto. Dado que el coseno del ángulo se puede expresar como el producto punto entre los vectores, la expresión para el modelo de iluminación de Phong se escribe finalmente como:

$$I_\lambda = I_{a\lambda} k_{a\lambda} O_{d\lambda} + f_{att} I_{p\lambda} \left[ k_d O_{d\lambda} (N \cdot L) + k_s O_{s\lambda} (R \cdot V)^n \right]$$

Para determinar el vector de reflexión, se calcula la reflexión del vector  $L$  en el plano formado por  $L$  y el vector normal  $N$  a la superficie, alrededor de  $N$ . De este modo se obtiene  $R = 2N(N \cdot L) - L$ . El exponente  $n$  modifica el comportamiento especular, haciendo que la superficie sea brillante para un menor rango de ángulos a medida que  $n$  aumenta. Todos los vectores mencionados anteriormente se encuentran normalizados (la magnitud de cada vector es igual a 1).

#### 4.3.4. Sombreado

Se puede asegurar que ninguna superficie real posee una tonalidad constante, debido a que las fuentes de luz ubicadas en una región limitada del espacio inciden con ángulos e intensidades diferentes sobre los puntos pertenecientes a una superficie dada. Esto se debe a la variación de su distancia respecto a la fuente puntual de luz; por lo tanto, en el procedimiento de despliegue se debe implementar algún mecanismo para modificar la tonalidad de cada polígono de la escena a través de los puntos que lo conforman. Una de las ventajas de implementar un método de sombreado en el despliegue es que con él se pueden disimular hasta cierto punto los artefactos de discretización de los objetos tridimensionales mediante polígonos, entregando un efecto de suavidad y continuidad a las superficies aproximadas por mallas poligonales. Para modelar la variación de intensidad existen varias estrategias, siendo las más utilizadas las siguientes: la primera, desarrollada por Gouraud, interpola las componentes frecuenciales de color; la segunda, resultado del trabajo de Phong, interpola el vector normal a la superficie del objeto.



**Figura 4.9.** Iluminación.  $L$  es el vector de incidencia de la luz,  $N$  el vector normal a la superficie,  $R$  el vector de reflexión y  $V$  el vector de observador.

#### 4.3.4.1. Sombreado de Gouraud

Esta estrategia asume que se puede obtener un resultado aceptable interpolando linealmente las componentes de color sobre la superficie del objeto, a partir del obtenido al aplicar el modelo de iluminación para cada uno de sus vértices (Figura 4.9(b)). La implementación consta de los siguientes pasos:

1. Calcular el vector normal de cada vértice como el promedio (o suma vectorial normalizada) de los vectores normales de los polígonos que contienen a dicho vértice. (Figura 4.9(a))
2. Aplicar el modelo de iluminación a cada vértice, asignándole el color resultante.
3. Interpolarse linealmente las intensidades de los componentes de color a lo largo de las aristas de cada polígono, asignando a cada punto el color resultante de la interpolación lineal de los dos vértices que generan la arista a la que pertenece.
4. Interpolarse las componentes de color para cada punto a lo largo de las líneas de rastreo, utilizando los colores obtenidos para los extremos de dichas líneas, ubicados sobre las aristas del polígono.

El sombreado de Gouraud es eficiente comparado con otros métodos, razón por la que es extensamente utilizado por los juegos de vídeo y otras aplicaciones en las que el desempeño es más importante que el resultado final, sin embargo presenta varios problemas, como el efecto de bandas y la degeneración del comportamiento especular del objeto

#### 4.3.4.2. Sombreado de Phong

El sombreado de Phong busca disimular la aproximación de una superficie curva mediante un muestreo de polígonos en  $\mathbf{R}^3$ . Para conseguirlo se modifica el vector normal a lo largo de la superficie de cada polígono mediante interpolación y se aplica el modelo de iluminación a cada punto (Figura 4.9(c)). Esto causa que el desempeño sea más lento que en el sombreado de Gouraud, si bien la calidad del resultado es superior. Los pasos para implementar el sombreado de Phong son los siguientes:

1. Calcular el vector normal de cada vértice en la misma forma que en el sombreado de Gouraud.
2. Interpolar el vector normal a la superficie sobre los puntos pertenecientes a las aristas de cada polígono, lo cual se consigue interpolando linealmente las componentes  $x$ ,  $y$  y  $z$  a partir de las normales de los vértices.
3. Interpolar el vector normal para cada punto a lo largo de las líneas de barrido, utilizando las normales de los extremos de dichas líneas, ubicados sobre las aristas.
4. Simultáneamente al paso anterior, para cada punto en las líneas de barrido aplicar el modelo de iluminación inmediatamente después de calcular su vector normal.
5. Es fácil implementar los algoritmos de sombreado paralelamente a los criterios imagen-espacio para la detección de superficies visibles.

#### 4.3.5. Detalle de Superficie (Textura)

Las superficies de la mayor parte de los objetos reales no corresponde a planos perfectos ni curvas suaves, por el contrario presentan múltiples características como rugosidades aleatorias, patrones más o menos regulares, estampados, tejidos, mosaicos, etc., de tal modo que la variedad posible de superficies es prácticamente infinita. En el camino hacia el realismo visual, una aproximación al modelamiento de estos detalles de superficie es necesaria. Sin embargo la carga computacional de una implementación estricta de las características superficiales de los objetos es demasiado pesada para la mayoría de computadores en la actualidad, más aún si se requiere de cierta rapidez de despliegue. Surgen entonces las técnicas de mapeo, cuyo principio consiste en plasmar sobre la superficie de un objeto (modelado con algún grado de simplicidad geométrica) una representación externa de

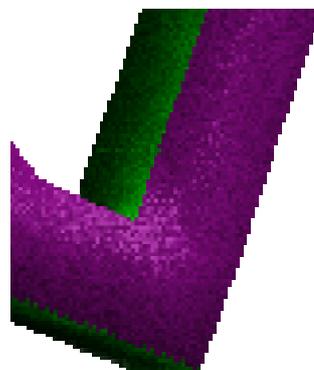
sus características superficiales. Así se consigue atender de manera aceptable la necesidad de presentación de los detalles de superficie de los objetos sin afectar sensiblemente la rapidez de los cálculos. Para el mapeo del detalle de superficie existen dos enfoques principales: la correspondencia de patrones y la correspondencia de protuberancias.

#### 4.3.5.1. Correspondencia de Patrones

En este método se establece una relación uno a uno entre los elementos de una imagen digital o matriz de tonalidades (que representa el detalle de superficie) con espacio de coordenadas  $(u, v)$  y elementos cuadrangulares definidos sobre la superficie del polígono con su propio espacio de coordenadas  $(s, t)$ . Para efectuar el mapeo de la textura sobre la superficie se utiliza una matriz de transformación que determina, en el algoritmo de despliegue el color de cada elemento cuadrangular sobre un polígono, según sea su equivalente en el espacio de coordenadas de la imagen, para luego efectuar la proyección sobre la superficie de observación. En algunos casos es necesario implementar un método de *antialiasing* cuando varios elementos de textura comparten el mismo pixel.

#### 4.3.5.2. Correspondencia de Protuberancias

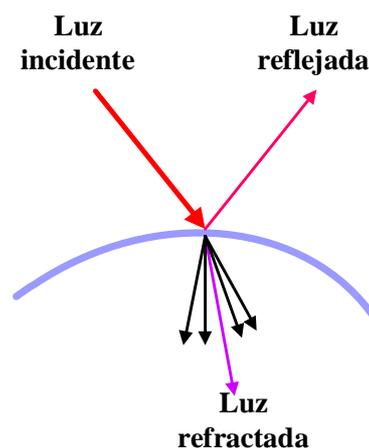
La correspondencia de protuberancias se puede describir rápidamente como una convolución de la normal de cada polígono de un objeto con una función de variación de dicha normal, que bien puede estar definida por una ecuación en  $\mathbf{R}^3$  o una tabla de variaciones de las componentes  $x, y, z$  a través de la superficie del polígono. Con este método se obtienen buenos resultados cuando se modelan objetos de un solo color con rugosidades o patrones de alto o bajo relieve. En la aplicación desarrollada se implementó un mapeo de protuberancias sencillo que simula una rugosidad aleatoria de rango variable (Figura 4.10).



**Figura 4.10.** Rugosidad modelada mediante correspondencia de protuberancias.

#### 4.3.6. Transparencia

Uno de los fenómenos ópticos más llamativos y que puede agregar al resultado del despliegue una mayor sensación de realismo es la transparencia. Como es sabido, una gran variedad de objetos físicos, como el agua, los cristales, algunos materiales sintéticos, el vidrio, etc., presentan características de translucidez, es decir, la propiedad de permitir el paso a través de su volumen de una cierta proporción de la luz que incide sobre ellos. Esta proporción varía de acuerdo con las propiedades del material del objeto transparente, además del grosor que la luz debe atravesar en su camino hacia el observador, de la proporción que es dispersada en el interior del objeto y la que es reflejada hacia el exterior sin que consiga atravesar la superficie (Figura 4.11).



**Figura 4.11.** Aspectos del fenómeno de transparencia. Cierta proporción de la luz incidente sobre el material transparente es reflejada, otra parte se dispersa al interior del objeto (flechas negras) en tanto que la restante atraviesa al objeto transparente con un cierto ángulo de refracción.

El enfoque para la implementación computacional de la transparencia asigna a cada material un coeficiente  $k_t \in [0,1]$  a partir del cual se define que tan transparente o que tan opaco es un objeto. Las aplicaciones que requieren de una gran rapidez de los cálculos asignan generalmente a  $k_t$  un valor constante, mientras que aquellas que se especializan en el realismo visual modelan el comportamiento no-lineal de la luz que atraviesa un objeto de grosor variable. El modelo de Kay asume que existe un coeficiente mínimo de transparencia  $k_{t_{min}}$  para cualquier punto sobre la superficie del objeto transparente y que el comportamiento translúcido es más notorio a medida que su vector normal se aproxima a la línea de visión del observador.

Así el coeficiente de transparencia para cada punto sobre la superficie del objeto se evalúa como:

$$k_t = k_{t_{\min}} + (k_{t_{\max}} - k_{t_{\min}})(1 - (1 - z_N)^m)$$

Donde  $k_{t_{\max}}$  es el valor máximo de la transparencia del objeto,  $m$  es el exponente asociado a la densidad del objeto transparente y  $z_N$  es la componente del vector normal a la superficie para un punto dado en la dirección del observador. Existen varios métodos para modelar la transparencia dependiendo de la importancia que tenga para la aplicación la calidad del resultado frente a la celeridad de los cálculos que conducen al despliegue.

#### 4.3.6.1. Transparencia de malla

La transparencia de malla es un modelo simplificado, que simula al objeto transparente como una malla de puntos distribuidos de forma más o menos regular, de tal forma que el objeto se dibuja como una máscara para la cual una porción de puntos correspondiente a  $k_t$  es desplegada con el color del objeto, mientras que el resto, correspondiente a  $1 - k_t$ , es ignorado. De este modo el objeto transparente se presenta como una cortina que deja entrever los objetos que se encuentran detrás de él. Este modelo es usado en aplicaciones que implementan el *z-buffering* para la detección de superficies visibles, debido a que con él se evitan problemas de transparencias recursivas (varios objetos transparentes colocados uno detrás del otro) a la par que se aceleran los cálculos, si bien se desmejora la calidad del resultado del despliegue.

#### 4.3.6.2. Transparencia interpolada

Entre las alternativas más empleadas para la implementación de la transparencia se encuentra la de simular la composición de los colores del objeto transparente y los de los objetos que se encuentran detrás de él mediante interpolación lineal entre sus componentes de color. Así, para cada componente frecuencial  $\lambda$ , la intensidad se calcula como

$$I_\lambda = (1 - k_t)I_{\lambda_1} + k_t I_{\lambda_2}$$

Donde  $k_t$  es el coeficiente de transparencia del objeto traslúcido,  $I_{\lambda_1}$  es la intensidad debida al objeto transparente e  $I_{\lambda_2}$  la intensidad debida a un objeto posterior. Cuando  $k_t$  es 0 el polígono

es totalmente opaco y no permitirá observar lo que se encuentre detrás, pero cuando  $k_t$  es 1, el polígono será totalmente transparente y, por lo tanto, invisible.

#### 4.3.6.3. Transparencia filtrada

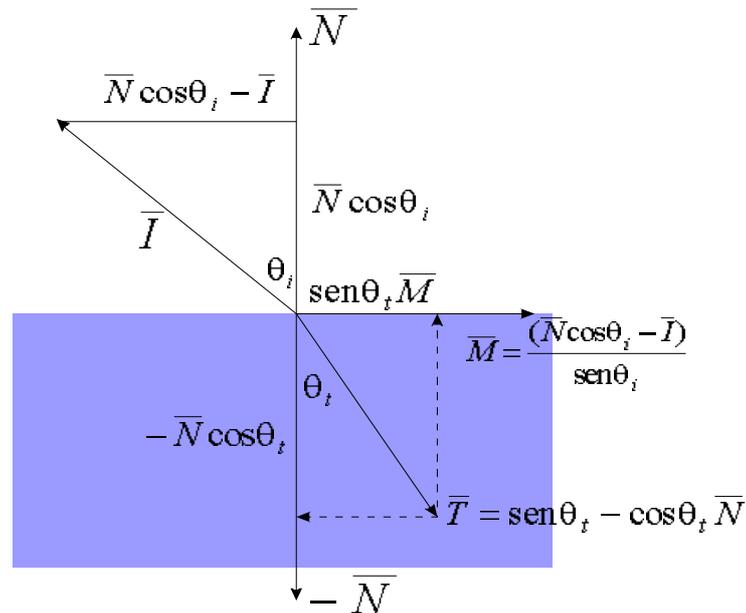
Un enfoque más cercano a la realidad es el de asumir el objeto transparente como un filtro de longitudes de onda, el cual permite el paso de los componentes frecuenciales del color en diferente proporción dependiendo de su tonalidad. Entonces, la intensidad de cada componente de color está dada por:

$$I_\lambda = I_{\lambda_1} + k_t O_{t_\lambda} I_{\lambda_2}$$

En donde  $O_{t_\lambda}$  es el color de transparencia del objeto traslúcido. El problema de este modelo es que la simple suma de componentes de color podría en algunos casos dar un resultado superior al rango en el que se definen las intensidades.

#### 4.3.6.4. Transparencia Refractiva vs. Transparencia No Refractiva

Como se mencionó anteriormente, uno de los aspectos fundamentales del fenómeno óptico de la transparencia es el cambio en la dirección de los rayos que atraviesan al objeto debido a la variación en la velocidad de la luz en el medio transparente respecto a dicha velocidad en el medio externo. Este cambio genera una diferencia entre el ángulo incidente y el ángulo de refracción, lo cual ocasiona que los objetos que se encuentran detrás del transparente aparecen distorsionados a través de él.



**Figura 4.12.** Cálculo del vector de refracción

La esencia del modelamiento de la refracción es el cálculo del ángulo de refracción para una superficie transparente dada (Figura 4.12). Una alternativa es utilizar la ley de Snell, que relaciona los ángulos de incidencia y de refracción con los índices propios de cada medio de difusión de la luz:

$$\frac{\text{sen } \theta_i}{\text{sen } \theta_t} = \frac{\eta_{t\lambda}}{\eta_{i\lambda}} = \eta_{r\lambda}$$

El vector  $T$  se calcula a partir de las componentes perpendicular y paralela a la superficie de refracción,  $M$  y  $N$  respectivamente. De esta forma  $\bar{T} = \text{sen } \theta_t \bar{M} - \text{cos } \theta_t \bar{N}$  donde  $\theta_t$  es un valor desconocido; para determinar el vector de refracción se debe expresar en términos de valores conocidos. De esta forma, al utilizar las propiedades del producto punto y las identidades trigonométricas se puede llegar a una expresión numérica para el vector de refracción:

$$\bar{T} = \left( \eta_{r\lambda} (\bar{N} \cdot \bar{I}) - \sqrt{1 - \eta_{r\lambda}^2 (1 - (\bar{N} \cdot \bar{I})^2)} \right) \bar{N} - \eta_{r\lambda} \bar{I}$$

Esta ecuación es bastante compleja y no se presta para garantizar una optimalidad en los cálculos que conducen a un despliegue eficiente en pantalla. Sin embargo, expresiones de este tipo son muy utilizadas por los métodos que se dirigen hacia la búsqueda del realismo visual, como la traza de rayos (*ray tracing*), obteniéndose efectos más realistas.

## **5. IMPLEMENTACIÓN DEL SISTEMA**

La implementación del sistema de reconstrucción gráfica tridimensional de objetos físicos comprende dos etapas para la reconstrucción tridimensional de objetos físicos. La primera de ellas corresponde a la elaboración del hardware para el registro de posiciones, que a su vez consiste en cuatro aspectos básicos: la herramienta para la ubicación de coordenadas, el dispositivo para la detección del desplazamiento angular, el registro de las rotaciones y la comunicación con el computador. La segunda consiste en el desarrollo de la herramienta de software para generación y presentación de modelo tridimensional a partir de la información obtenida y registrada por el hardware. En las secciones 5.1 y 5.2 se hace una revisión del proceso que condujo a la superación de estas dos etapas.

### **5.1. REGISTRO DE POSICIONES TRIDIMENSIONALES**

Esta sección constituye la primera de las dos grandes partes para la implementación del sistema y describe la parte mecánica y electrónico del proyecto. El objetivo es el registro de los ángulos que junto con la configuración específica de la herramienta de ubicación de coordenadas, permiten mediante la aplicación de la solución al problema cinemático directo calcular coordenadas tridimensionales ubicadas dentro del espacio de trabajo. Este incluye, desde la manipulación de las señales emitidas por los sensores ubicados en la herramienta hasta la comunicación serial con el PC.

### 5.1.1. Configuración De La Herramienta De Ubicación De Coordenadas

Para la construcción de la herramienta de ubicación de coordenadas es fundamental tener en cuenta que la ubicación de puntos debe realizarse sobre la superficie de cualquier objeto, lo cual implica la variabilidad de forma, tamaño y textura en cada uno de ellos. Así, la maniobrabilidad de esta herramienta permite llegar a los puntos que se encuentran en su espacio de trabajo realizando diferentes trayectorias o recorridos, logrando ubicar un punto a pesar de que en varios de los desplazamientos de sus enlaces, estos puedan encontrar obstáculos.

#### 5.1.1.1. Grados de Libertad y número de enlaces.

La herramienta tiene cinco grados de libertad y la conforman cuatro enlaces ( $L_0, L_1, L_2, L_3, L_4$ ). Las articulaciones ( $A_0, A_1, A_2, A_3, A_4$ ) se encuentran en la unión de los enlaces y junto con estos determinan un plano único sobre el cual realizan sus rotaciones. Salvo en la unión de los enlaces  $L_0$  y  $L_1$  en donde se pueden realizar rotaciones sobre más de un plano, en cada unión de enlaces las rotaciones se efectúan sobre solo uno de ellos.

Los enlaces  $L_0, L_1, L_2, L_3, L_4$  tienen respectivamente 21, 15.2, 15 y 12 cm. de longitud. (Figura 5.1).

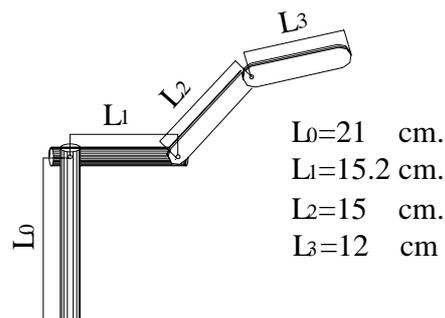
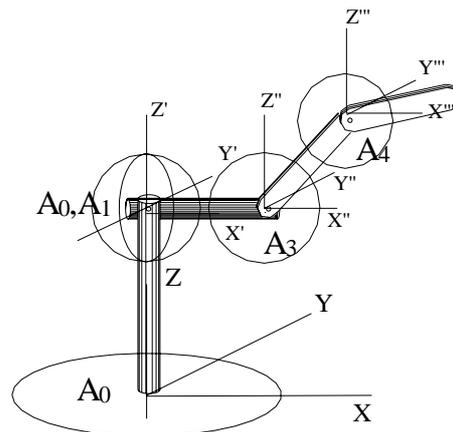


FIGURA 5.1 Enlaces y Articulaciones

#### 5.1.1.2. Planos de Rotación de las Articulaciones

La primera de las articulaciones ( $A_0$ ) se encuentra ubicada en la base del enlace  $L_0$  y permite que los giros se realicen sobre el plano  $XY$ . En la siguiente unión entre los enlaces

$L_0$  y  $L_1$  están ubicadas dos articulaciones ( $A_1, A_2$ ) que permiten realizar giros en dos planos, el plano  $XZ$  y el plano  $YZ$  dando una mejor maniobrabilidad a la herramienta. Las dos siguientes articulaciones ubicadas en las uniones de los enlaces  $L_1-L_2$  y  $L_2-L_3$  efectúan sus giros sobre el plano  $XZ$ . (Figura 5.2).



**FIGURA 5.2** Planos de rotación de las articulaciones sobre los ejes auxiliares.

Se puede observar como cada articulación gira en su propio eje de coordenadas excepto  $A_0$ , que lo hace sobre el sistema de coordenadas principal. La rotación que una articulación hace afecta la posición de los siguientes enlaces y por lo tanto la del efector final.

### 5.1.2 Dispositivo para la detección del desplazamiento angular.

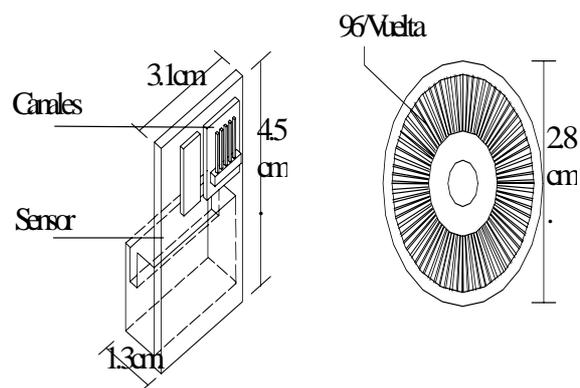
La utilización de mecanismos sensores externos permite interactuar con el entorno de una manera flexible. La utilización de tecnología sensorial para dotar a las máquinas con un mayor grado de inteligencia al tratar con su entorno es realmente un tema de investigación y desarrollo activo en el campo de la robótica y los sistemas de control. El sistema empleado para el registro de rotaciones consiste en un sensor de rotación también conocido como *encoder* el cual es descrito a continuación.

#### 5.1.2.1 Sensor óptico de rotación (Encoder).

El mecanismo utilizado para realizar los registros del ángulo que un determinado enlace realiza con respecto a una posición inicial se denomina sensor de revolución ó Encoder. Cada uno de estos puede detectar el giro de un enlace hasta en medio grado ó menos, con

lo cual la complejidad del circuito y el costo del mismo se incrementan de manera gradual. Un sensor de revolución ó rotación posibilita detectar el sentido del giro y el paso por la posición inicial (*paso por cero*). El sensor que se utiliza, consta básicamente de un circuito captador y una plantilla de revolución (Figura 5.3). El mínimo ángulo de giro que se puede detectar determina la resolución del encoder. Dependiendo de estos los encoders se pueden clasificar de alta media y baja resolución.

**5.1.2.1.1. Descripción.** El captador tiene un sensor de luz que funciona de manera similar a un interruptor. Cuando entre el emisor y el receptor del sensor no hay ninguna barrera u obstáculo, la cantidad de luz que llega al receptor es la máxima por lo tanto el sensor emite un voltaje que es el mínimo.

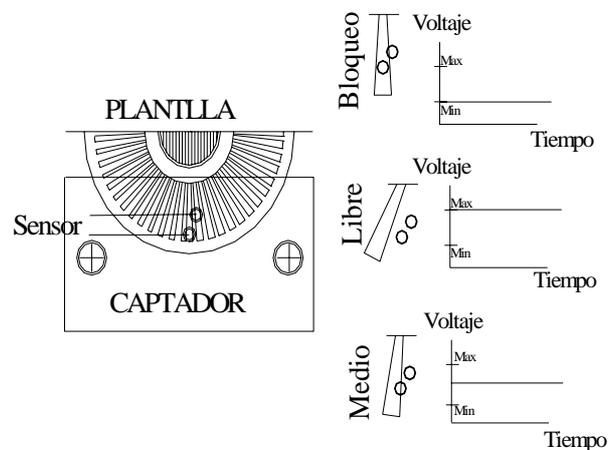


**FIGURA 5.3** Captador y Plantilla de Revolución.

Cuando una barrera impide el paso de luz del emisor al receptor el voltaje emitido por el sensor es el máximo. Este voltaje que se emite y que se encuentra en el rango de 1 voltio, está determinado por la cantidad de luz que llega al receptor. Así, los voltajes emitidos no conservan dos estados discretos como sucede en un interruptor (Encendido/Apagado), sino que permite cualquier salida de voltaje posible que esté en el rango anteriormente nombrado.

La plantilla de revolución es la pieza encargada de interferir entre el emisor y el receptor del sensor. Es una pieza circular en cuyo centro se permite introducir un eje sobre el cual se puede hacer girar la plantilla. Sobre su superficie se encuentran las ranuras que permiten

bloquear (ó comunicar) el flujo de luz entre el emisor y el receptor (Figura 5.4). Tiene 96 de estas ranuras que permiten calcular el giro que se realiza. Cada una de las ranuras equivale a 3.7 grados aproximadamente. Es decir, que el mínimo registro que podemos realizar con este tipo de sensores es de 3.75 grados. Giros inferiores a éste no pueden ser detectados. De acuerdo a lo anterior se puede considerar este encoder como de baja resolución



**FIGURA 5.4** Voltajes máximo y mínimo emitidos por el sensor.

**5.1.2.1.2. Canales de Salida.** Se observa en la figura 5.1.a que el captador tiene cinco canales que proporcionan la información referente al encoder.

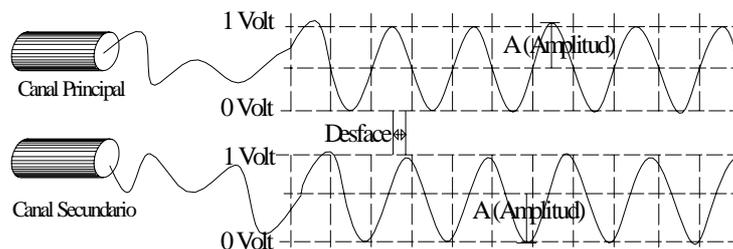
- Canal Uno: El primero de los canales en la práctica se identifica por ser un cable de color rojo; es el **Vcc**, voltaje de referencia. Este debe conectarse a 5 volts.
- Canal Dos: Corresponde a la tierra o ground. Se identifica por ser el cable de color café. Debe conectarse a tierra.

Los siguientes tres canales son de suma importancia, emiten voltajes entre 1 volt y 2 volts.

- Canal tres: Cuando se realizan giros a una velocidad constante, la señal se convierte en una onda seno con amplitud de 0.5 volts ya que los voltajes emitidos varían de acuerdo a la luz incidente en el receptor. Recibe el nombre de canal principal.

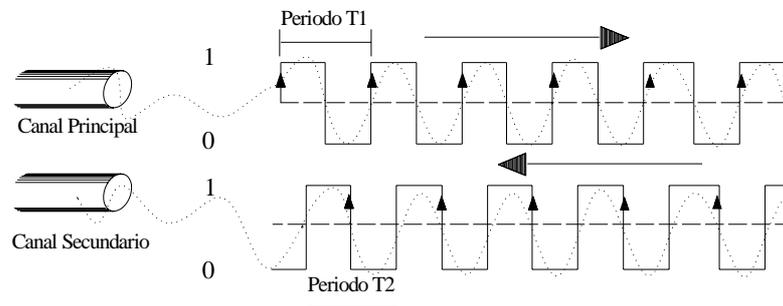
- Canal Cuatro: Este canal tiene las mismas características que el anterior canal y recibe el nombre de canal secundario. La diferencia radica en la fase de éste con respecto al canal principal. El desfase consiste en un corrimiento de la onda en el eje del tiempo, este corrimiento se da en radianes y nos permite determinar en que sentido se esta realizando un giro.
- Canal Cinco: Este canal nos permite determinar cuando se ha efectuado un giro completo, es decir; 360 grados. Es el paso por el origen. La importancia de estos canales se describe a continuación.

**5.1.2.1.3. Funcionamiento.** Se nombró anteriormente que la única diferencia entre las señales emitidas por el canal principal y el secundario es la fase. Ésta, se puede observar cuando al mover la plantilla de revolución a una velocidad constante, los canales muestran ondas senos de igual amplitud pero desfasadas un determinado ángulo. (Figura 5.5)



**FIGURA 5.5** Desfase de las ondas emitidas por los canales principal y secundario.

Como se explicó, los sensores a través de sus canales de información emiten voltajes en un determinado rango; pero para el registro de los ángulos de rotación no es de interés voltajes intermedios al producido durante el registro de un pulso, es decir 3.7 grados. El interés radica en identificar cuando se produce un cambio de voltaje máximo, a un voltaje mínimo ó viceversa. Para lograr este objetivo, estas señales se transforman en señales que emiten estados discretos.



**FIGURA 5.6.** Transformación de las ondas seno en ondas cuadradas utilizando un comparador.

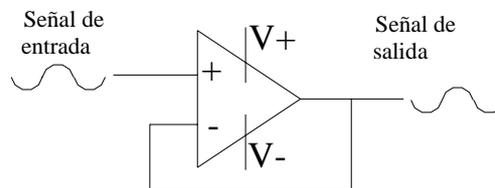
La transformación de la señal se logra asumiendo que para un valor de la onda superior a otro que se conoce como *comparador*, la onda tomara siempre el voltaje máximo y para un valor de la onda menor al comparador el valor de ésta es el voltaje mínimo logrando así *estados discretos*.

Cuando ocurre un cambio de 0 a 1 la señal del canal principal o secundario se dice que ocurre un pulso de subida. Al cambio de estado de 1 a 0 se le llama pulso de bajada. Se puede observar al recorrer la señal del canal primario de izquierda a derecha que al ocurrir un pulso de subida; el valor de la onda desfasada del canal secundario se encuentra en el valor *ceros*. Al realizar el mismo recorrido en sentido contrario (derecha a izquierda) el valor del canal secundario al momento de producirse un pulso de subida (que en el anterior recorrido se comportaban como pulso de bajada) es diferente. Esta es la forma en la cual se sabe el sentido en el cual se está realizando el giro (Figura 5.6).

### 5.1.2.2 Transformación de las señales.

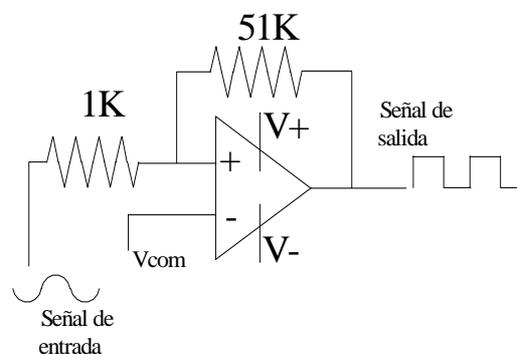
Las señales provenientes de los sensores de rotación antes de ser procesadas y transmitidas al computador, pasan por una etapa de aislamiento y una de acoplamiento que son descritas a continuación y que emplean básicamente amplificadores operacionales.

**5.1.2.2.1. Etapa de aislamiento.** La función de esta etapa es aislar la señal emitida por el sensor y la que trabaja la etapa de acoplamiento. Este aislamiento utiliza un amplificador que actúa como seguidor (Figura 5.7).



**FIGURA 5.7** Aislamiento de las Señales

**5.1.2.2.2. Etapa de Acoplamiento.** Las señales emitidas por los sensores deben pasar por una fase de acoplamiento para que el microcontrolador pueda trabajarlas. Esta fase permite que el grupo de voltajes que el sensor emite, y que básicamente están comprendidos en un rango de un voltio, sean transformadas en solo dos valores posibles logrando lo que anteriormente se denominó como estados discretos. Esta transformación se realiza utilizando un circuito comparador. (Figura 5.8).

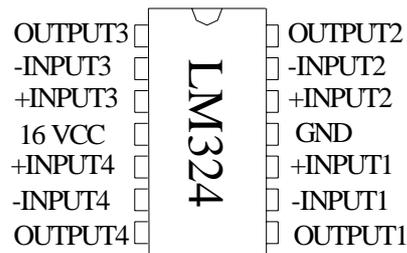


**FIGURA 5.8** Circuito Comparador

La señal proveniente del sensor de rotación (*Canal principal ó secundario*) actúa como entrada positiva del amplificador operacional pasando por una resistencia de 1K.  $V_{com}$  es el voltaje comparador, que dependiendo del rango de voltajes que emite el sensor toma el valor medio. Así, cuando el sensor emite un valor mayor que el voltaje comparador ( $V_{com}$ ); el amplificador operacional se satura y envía a la salida  $V+$ . Cuando ocurre el caso

contrario el amplificador actúa de forma similar y envía  $V_-$ , obteniendo la señal adecuada para que sea trabajada por el microcontrolador.

Los amplificadores operacionales utilizados tanto en la fase de asialamiento como en la de acoplamiento son LM324 (Figura 5.9).



**FIGURA 5.9** Amplificador Operacional LM324

### 5.1.3. Registro de Posiciones Angulares.

Una vez finalizada la etapa de acoplamiento, el microcontrolador puede trabajar y procesar las señales enviadas por los sensores de rotación ubicados en la herramienta de ubicación de coordenadas, utilizando para ello los puertos programables como entrada salida. El objetivo es determinar los ángulos correspondientes a las rotaciones de los enlaces cuando se detecta un movimiento en cualquiera de ellos.

El componente principal es el microcontrolador (*PIC-16C74A*). Para el registro se utilizan como entrada dos de los puertos programables como *entrada/salida*. De esta manera se puede leer periódicamente los datos suministrados por los sensores. Dependiendo de estas señales y teniendo en cuenta a que sensor corresponde, se realiza el incremento o decremento de cinco registros contadores dependiendo de la dirección en la cual se realiza el giro. Estos contadores corresponden a las articulaciones nombradas en la sección 5.1.1, los cuales son finalmente enviados al computador cuando se desea establecer las coordenadas tridimensionales de un punto.

### 5.1.3.1. El Microcontrolador. (PIC16C74)

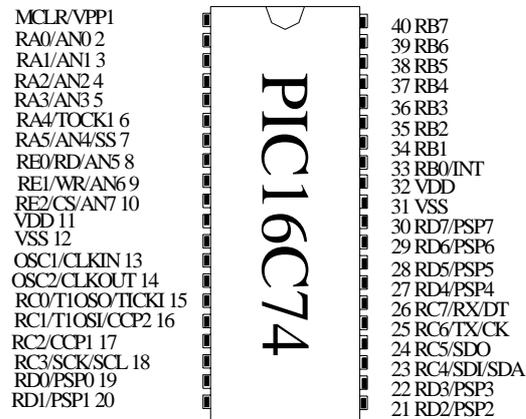
Maneja aproximadamente 45 registros de funciones especiales de hardware (Figura 5.10). Cada uno de estos registros tiene un tamaño de un byte (8 bits). Cada bit de cada registro realiza una tarea o función única. Entre otras características, el microcontrolador trabaja con 35 instrucciones simples, con las cuales se pueden construir librerías y macros para propósitos específicos y generales. Además:

- Todas las instrucciones simples tienen una duración de un ciclo de reloj a excepción de las comparaciones que tienen una duración de dos ciclos de reloj.
- Tiene tres modos de direccionamiento: Directo, indirecto y relativo.
- Posee 33 pines de entrada salida con sus respectivos controles de dirección.
- Un conversor análogo digital con ocho canales de entrada.
- Interface de comunicación serial (SCI), que maneja comunicación asincrónica *full-duplex* y *half-duplex*.

Por defecto los registros de control de entrada/salida (*TRISX*), configuran de manera inicial el puerto como entrada; es decir el contenido del registro es en binario, 0XB11111111 o en decimal 255.

**5.1.3.1.1. Puertos.** El microcontrolador tiene cinco puertos (*PORTA*, *PORTB*, ... , *PORTE*). Se mencionó anteriormente la posibilidad que tiene el microcontrolador de programar sus puertos como entrada ó salida. Esa función la desarrolla el registro llamado *TRISX*, donde la *X* corresponde a la letra del puerto que se desea programar (*A*, *B*, *C*, *D*, *E*). Si se coloca uno en un bit *x* de un registro *TRISY*, se configura el conector *x* del *PUERTOY* como entrada. De igual forma la asignación de un cero configura ese conector como salida. Los puertos utilizados en el desarrollo del proyecto son *PORTA* y *PORTB*.

Es claro que los registros *TRISX* son los encargados de configurar como entrada ó salida los puertos y que en los registros *PORTX* contienen los datos con de entrada o salida desde el puerto.



**FIGURA 5.10** Microcontrolador PIC16C74

En el puerto *PORTB*, todas las entradas se toman como digitales. A diferencia de otros puertos (*PORTA*), *PORTB* está conformado por ocho (8) pines los cuales en su orden se identifican de la siguiente manera *RB0*, *RB1*, *RB2*, *RB3*, *RB4*, *RB5*, *RB6* y *RB7*. El primero de estos puede desarrollar una de dos funciones: Como entrada normal del puerto ó como interrupción (*INTE*). Se explica más adelante como funciona este pin, cuando es configurado como interrupción.

El puerto *PORTA*, está conformado por sólo cinco conectores los cuales se identifican de manera análoga a los de *PORTB*; es decir, *RA0*, *RA1*, *RA2*, *RA3* y *RA4*. Cada uno de estos pines a excepción del *RA4* que esta en colector abierto, están configurados inicialmente para tratar señales analógicas y no digitales como en el caso del *PORTB*.

**5.1.3.1.2. Interrupciones.** Una interrupción en un microcontrolador funciona de manera análoga a la siguiente situación: Imagine una persona desarrollando determinadas labores en su oficina; suena el teléfono, la persona contesta, atiende la llamada y luego continua desarrollando sus actividades.

Las interrupciones tienen prioridad sobre el programa que se este ejecutando. Existen varias interrupciones y, registros que permiten habilitar o deshabilitar cada una de ellas. Entre las más importantes se encuentran:

- Interrupción por pin externo.
- Interrupción por Timer0.

- Interrupción por comunicación serial.
- Interrupción por el conversor análogo digital etc.

La interrupción por pin externo es la única que sucede por una situación diferente a los eventos realizados por el microcontrolador de manera interna; de allí su nombre. El pin habilitado para este fin es el mismo *RBO* ó *INTE*. La diferencia es la función que desarrolla y para la cual esta configurado.

Cuando se habilita la interrupción se hace referencia al registro *INTCON*, bit *INTE*. Al activarse la interrupción cuando se detecta un cambio de estado en este pin, se suspende el programa principal, se guarda el estado de este, y la ejecución se transfiere a una rutina de atención de la interrupción. Después de ejecutada ésta, se restablece el estado del programa principal y se continua con la ejecución de este. En caso contrario puede actuar como un pin de entrada/salida normal.

**5.1.3.1.3. Registros Asociados.** Otros registros a tener en cuenta en la programación del microcontrolador permiten la activación o deshabilitación de algunas funciones especiales que por omisión están activadas. Estos son:

- Registro *INTCON*: este registro está encargado básicamente de manejar y configurar algunas de las interrupciones que tiene el microcontrolador. Entre estas, esta el bit para la activación de la interrupción por pin externo y del *GIE* (*Global Interruption Enable*). Si no está activado el *GIE*, ninguna interrupción es admitida por el microcontrolador.
- Registro *OPTION*: en este registro se configura el flanco (de subida o bajada) mediante el cual se activa las interrupciones.
- Registro *ADCON*: por defecto el puerto A está configurado para recibir entradas análogas, por tal motivo es necesario configurarlo para que trabaje con señales digitales. Para hacerlo se utiliza el registro este registro.

**5.1.3.1.4. Código Assembler Manejado por el Microcontrolador.** El PIC16C74, tiene una arquitectura RISC con 35 instrucciones de bajo nivel. Todas las instrucciones son ejecutadas en un solo ciclo de instrucción, excepto cuando el programa realiza saltos

condicionales ó llamados a sub-rutinas. En este caso; la ejecución de esta instrucción toma dos ciclos. Un ciclo de instrucción consiste en cuatro periodos del oscilador.

Para el desarrollo de diversas aplicaciones en este tipo de microcontroladores existen diversos compiladores dentro de los cuales están:

- MPASM Assembler, el cual está integrado con el programa MPLAB que es suministrado por la empresa Microchip Technology Inc.
- MPC, El cual se puede instalar como un módulo de MPLAB para el desarrollo de las aplicaciones manejando código en lenguaje de programación C. La distribución de este la realiza la empresa ByteCraft.

A continuación se lista algunas de las instrucciones de bajo nivel que son reconocidas por la mayoría de estos compiladores y una breve descripción.

- *BCF*: Bit Clear *f*.
- *BSF*: Bit Set *f*
- *BTFSC*: Bit Test, Skip if clear
- *BTFSS*: Bit Test, Skip if set
- *CALL*: Realiza una llamada a una instrucción ubicada en una dirección *X*
- *SUBLW*: Realiza la resta
- *SUBWF*: Resta entre dos registros, uno de ellos *W* y el otro *F*.
- *XORLW*: Realiza la operación lógica OR exclusivo con un valor específico.
- *XORWF*: Realiza una operación lógica OR exclusivo con el registro *F*.

Para el caso en el cual se realizan restas, se usa el método de complemento a dos para la representación de números negativos. Para algunas de las instrucciones anteriormente nombradas que generalmente son las que realizan operaciones entre registros; existe la posibilidad de almacenar los resultados en el registro que el programador desee, el registro *W* que se conoce como registro de trabajo ó el registro con el cual se efectúa la operación.

**5.1.3.1.5. El Compilador Mpc De Bytecraft.** Todas las estructuras básicas del lenguaje C, están implementadas en este compilador. De esta forma se puede utilizar las estructuras de control básicas: *for*, *while*, *do while*, *switch*, etc.; así mismo los tipos de datos básicos: *char*, *int*, *bits* etc. Existen funciones especializadas que permiten el manejo de características muy importantes en el microcontrolador como lo son el manejo de las interrupciones y la transmisión del microcontrolador con el PC utilizando comunicación serial. Estas dos funciones son:

- `Setup_Async_Mode()`: esta función permite configurar la velocidad de transmisión en baudios, utilizando la forma de comunicación asincrónica más adecuada. (full duplex, half duplex, etc.). El argumento entero que recibe como parámetro en esta función, determina la velocidad de transmisión. Este se puede calcular de la forma  $Baude\ Rate = Fosc/64 * (x + 1)$ .
- `Transmit()` y `Receive()`: cuando se desea comunicar el microcontrolador con algún otro dispositivo, por ejemplo el computador; se puede diseñar macros para la configuración de cada uno de los registros que tienen que ver con esta comunicación. MPC posee funciones encargadas de configurar estas características; una de estas es como ya se vio es `Setup_Async_Mode()`. Otras funciones que intervienen en este proceso son las funciones `Transmit()` y `Receive()`.

Para el manejo de las interrupciones se utiliza la siguiente estructura:

```
#pragma option v;
#include <16C74.h>
void handle_PORTB();
void handle_EXTERN();
void handle_TMR0();
```

Corresponde a las funciones que se encargan de manejar cada una de las interrupciones, ya sea por conector externo, por temporizador (*timer*) ó por cambio en los cuatro primeros bits del puerto B.

La siguiente función se encarga de controlar en su conjunto; todas y cada una de las interrupciones.

```

void __INT(void)
{
    SaveContext
    if (INTCON.RBIF)
    {
        // Cambio sobre el puerto B<7:4>
        INTCON.RBIF = 0; // Borra RBIF
        handle_PORTB(); // Llama la función que maneja esa interrupción
    }
    if (INTCON.INTF)
    {
        // Interrupción por pin externo
        INTCON.INTF = 0; // Borra INTF
        handle_EXTERN(); // // Llama la función que maneja esa interrupción
    }
    if (INTCON.T0IF)
    {
        // interrupción por TMR0
        INTCON.T0IF = 0; // Borra T0IF
        handle_TMR0(); // // // Llama la función que maneja esa interrupción
    } // if
    RestoreContext
}

```

### Programa principal.

```

void main()
{
}

```

### Prototipo de las funciones.

```

void handle_PORTB()
{
}

void handle_EXTERN()
{
}

void handle_TMR0()
{
}

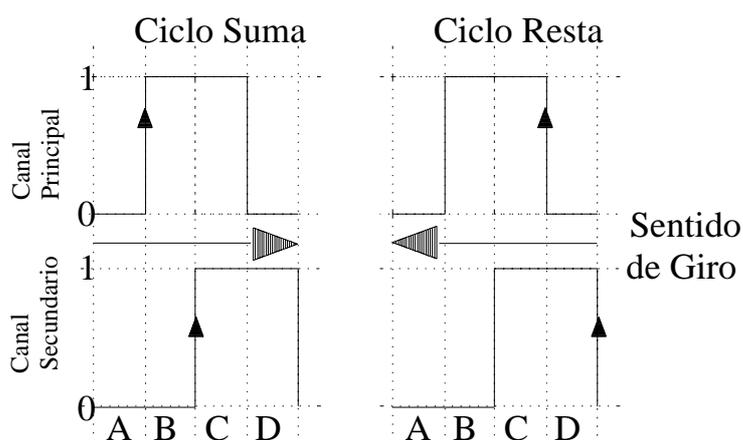
```

En general se pueden incluir funciones para cada una de las interrupciones que se manejen y que corresponden a las necesidades de una aplicación específica.

### 5.1.3.2. Algoritmo para el registro de rotaciones en el PIC.

Luego de haber tratado las señales del canal principal y secundario de los sensores de rotación haciendo que los voltajes se adecuen para que el microcontrolador pueda trabajarlas, el software del PIC se encarga de explorar periódicamente las entradas de los puertos PORTA y PORTB por donde se reciben estas señales para determinar de acuerdo a los valores en dicha señal si se debe o no realizar un decremento o un incremento en los contadores asociados a cada sensor.

**5.1.3.2.1. Variaciones de los contadores:** Los puertos solo pueden detectar la entrada de ceros o unos lógicos, de esta manera el microcontrolador puede detectar cuatro estados posibles analizando las entradas de los canales principal y secundario de un sensor determinado. Los estados son *00*, *10*, *11* y *01* donde el primer dígito corresponde a la entrada del canal principal y el segundo al canal secundario del encoder. Si se recorre en orden los estados hasta completar un ciclo, el software del PIC efectúa una variación en el contador correspondiente a ese sensor, lo cual se decide dependiendo del valor en el que se encuentra el canal secundario. A continuación se analiza los estados posibles de un sensor mediante el uso de las ondas cuadradas del canal principal y secundario. (Figura 5.11)



**FIGURA 5.11** Configuración de estados de las señales del sensor

Se puede observar que cuando se realiza un giro pasando por los estados  $A, B, C, D$ , el estado del canal secundario al ocurrir el pulso de subida en el canal principal es diferente al encontrado si se hace el recorrido en sentido contrario ( $D, C, B, A$ ). De esta forma el microcontrolador detecta si debe realizarse una suma o una resta.

**5.1.3.2.2. Validación de sumas y restas.** El movimiento arbitrario de los enlaces de la herramienta articulada puede hacer que la variación de los contadores no represente el valor real de la rotación correspondiente a una articulación. Nótese que las sumas se realizan siempre que ocurre un cambio de estado  $AB$ , que las restas ocurren cuando el cambio es  $DC$  y que una pequeña variación en la posición de los enlaces puede hacer que el mismo cambio de estado se repita sin haberse realizado un ciclo completo ( $ABCD$  ó  $DCBA$  respectivamente) para que la suma o resta sea válida. Por esto el programa del microcontrolador, encargado de la variación de los contadores, debe garantizar que las variaciones de estos se ejecuten tras efectuarse un ciclo completo luego de haberse hecho la última suma o resta.

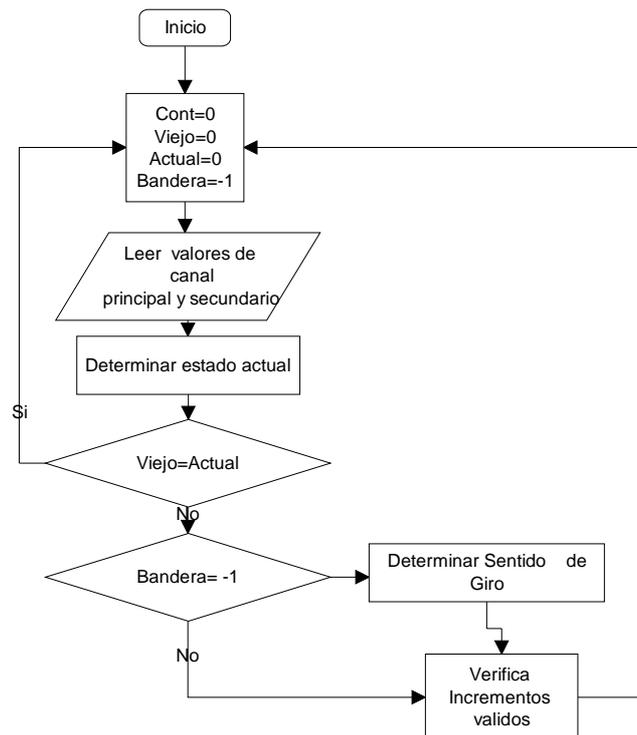
**5.1.3.2.3. Cadenas de estados.** En general el sensor de rotación puede generar diversas cadenas de estado. Dadas las siguientes cadenas de estados correspondientes a un sensor, en donde el valor inicial del contador asociado es cero ( $0$ ) se tiene que:

1.  $ABCDABCD$ .
2.  $BADCBADC$ .
3.  $ABCDABABABADC$ .
4.  $ABCD CDCBAD CBCBAD CADC$

Los valores finales de los contadores en cada una de las cadenas son respectivamente 2, -2, 1 y -2. En la primera cadena dos veces se cumple el cambio de estado  $AB$ , luego de completarse un ciclo. En la segunda cadena se presenta el caso similar pero realizando decrementos. En el último caso, se marca la importancia de verificar el cumplimiento de los ciclos para la verificación de las variaciones en el contador.

Nótese que el sensor no puede saltarse estados, es decir nunca ocurre el cambio de estados  $AC, CA, BD, DB, CA, AC, \delta DB$ , ya que los estados ocurren de manera consecutiva.

**5.1.3.2.4. Diagrama de flujo.** El siguiente diagrama muestra el algoritmo para registrar las sumas y restas validas deacuerdo a los estados que el sensor de rotación pueda presentar. Las variables *viejo* y *actual* registran respectivamente el estado anterior y actual del sensor de rotación. L variable bandera se utiliza para determinar las sumas y restas validas. (Figuras 5.12, 5.13, 5.14, 5.15).



**Figura 5.12** Programa del microcontrolador para el conteo de rotaciones

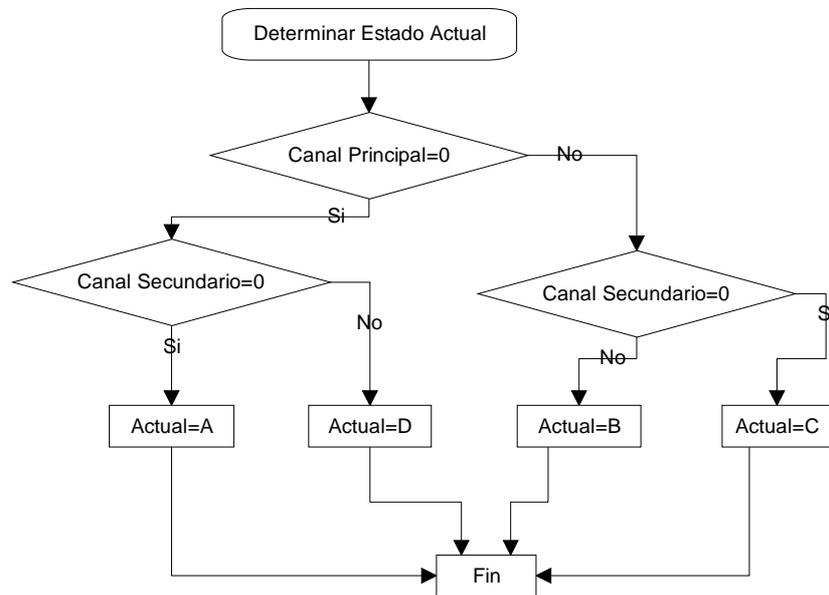


Figura 5.13 Función para la determinación del estado Actual

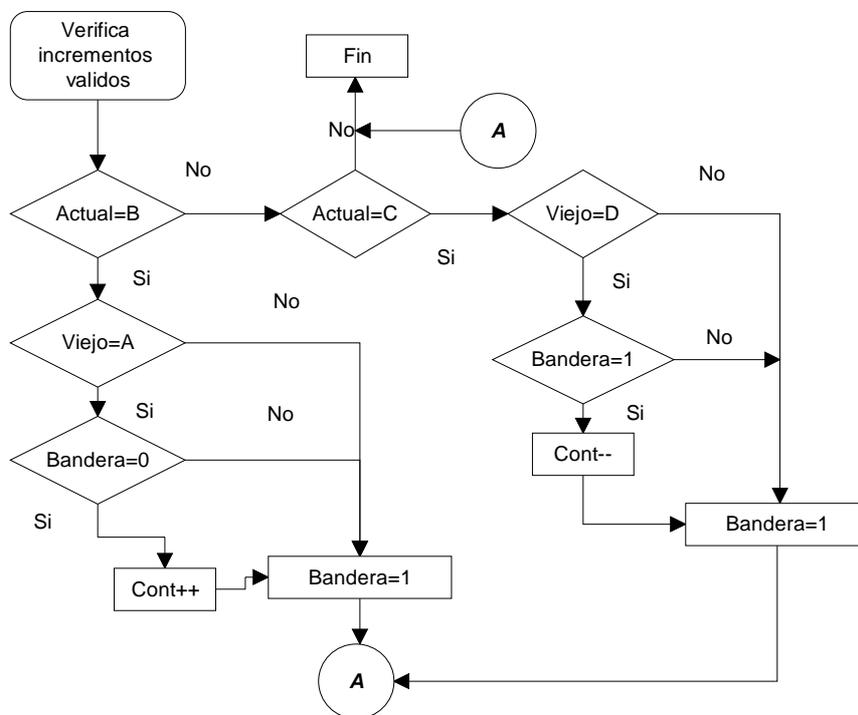
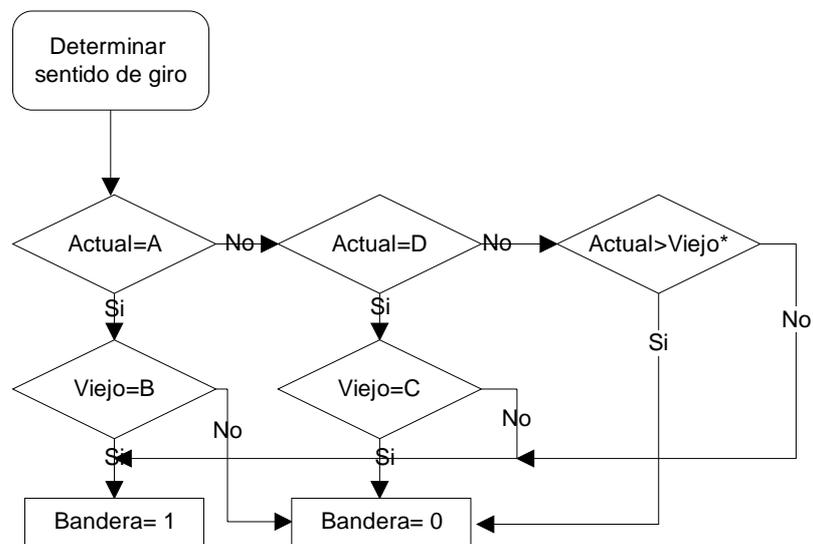


Figura 5.14 Modulo encargado de verificar el cumplimiento del ciclo de suma o resta



**Figura 5.15** Modulo para inicializar la bandera que determina incrementos o decrementos válidos. \*(Se cumple que  $D > C > B > A$ )

#### 5.1.4. Comunicación con el Computador

Este último bloque es el encargado de transmitir los datos desde el microcontrolador hacia la CPU del computador, dado que los voltajes con los que trabaja el microcontrolador y los del computador son diferentes. Se realiza un acoplamiento para poder realizar este tipo de comunicación utilizando para tal fin un integrado que se conoce como *MAX232*, el cual es una interfaz al nivel de la capa física apropiado para trabajar con comunicaciones semidúplex y dúplex. (Figura 5.16).

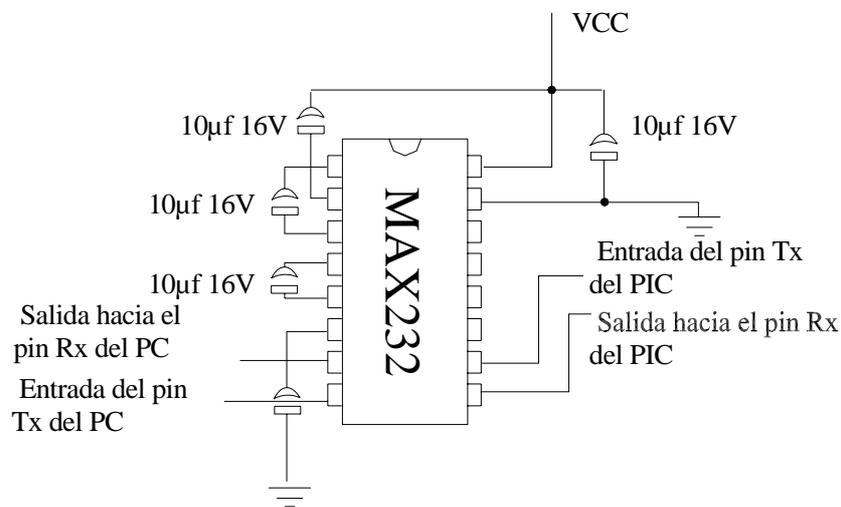


FIGURA 5.16 Interfaz de comunicación del MAX232

#### 5.1.4.1. La Interfaz Serie

Se hablo anteriormente de la posibilidad que tiene el computador de sacar información al exterior para lo cual dispone de puertos seriales y paralelos. Estos últimos se utilizan normalmente para conectar las impresoras, y los seriales para conectar módems, mouse etc. Ya que existe una gran cantidad de fabricantes de dispositivos de hardware, los conectores y demás elementos que intervengan en la comunicación de estos deben establecer un estándar que defina tanto las dimensiones físicas de los posibles conectores como la utilidad que va a tener que va a tener cada uno de ellos.

Desde un punto de vista técnico, se le llama interfaz al dispositivo que permite la conexión entre dos equipos para que exista comunicación entre ellos. En general, una interfaz puede ser un elemento de software ó hardware ya que también puede recibir el nombre de interfaz, aquel software que permite comunicar, por ejemplo, dos aplicaciones distintas.

Desde 1969 se han hecho públicas distintas normas de interfaz, cada una de ellas intentando evolucionar para permitir mayores velocidades de transmisión o mayores funcionalidades; sin embargo, la historia ha venido a darle el papel protagonista a la norma RS-232C, la cual, aun siendo de las más antiguas ha sido y es la más utilizada.

Se puede sin embargo encontrar otras interfaces como RS-449, RS-422 y RS-423 que son actualizaciones de la RS-232, diseñadas para permitir mayores velocidades de transmisión

de datos y mayor funcionalidad. También están la V24, V28 y X21 de la ITU (Unión Internacional de Telecomunicaciones),

#### **5.1.4.2. Interface RS232**

Para la comunicación entre equipos terminales ó computadoras y el módem, se ha desarrollado una interfaz conocida con el nombre de EIA RS-232-C. Esta interfaz es un protocolo de capa física del modelo OSI, que debe especificar con detalle la interfaz mecánica, funcional, de procedimientos y eléctrica. Esta última es de particular atención ya que los voltajes utilizados para la representación de unos y ceros binarios es diferente a los empleados por el circuito externo al computador. El estándar lo planeó la *EIA* (Electronic Industries Association).

En los estándares, la computadora recibe el nombre de data terminal equipment (*DTE*), y el módem, data circuit-terminating equipment (*DCE*).

**5.1.4.2.1. Especificación Eléctrica.** La especificación eléctrica para RS-232-C es que un voltaje más negativo que -3 volts es un uno binario y que un voltaje más positivo que +4 voltses un cero binario. Se permiten velocidades de transmisión de datos de hasta 20 kbps con una longitud de hasta 15 metros, ya que la longitud es inversamente proporcional a la velocidad y realmente lo que se limita es la longitud del pulso.

**5.1.4.2.2. Especificación Mecánica.** La especificación mecánica describe un conector de 25 pines de 47.04 m.m. de ancho, con todas las demás dimensiones especificadas con igual precisión. La hilera superior tiene los pines numerados de 1 a 13 (de izquierda a derecha); la hilera de abajo tiene las agujas numeradas de 14 a 25 (tambien de izquierda a derecha).

**5.1.4.2.3. Especificación Funcional.** Determina cuáles circuitos se conectan a cada una de los 25 pines, y qué significan. Nueve de los pines casi siempre se encuentran, las demás se omiten con frecuencia. A continuación se describe el funcionamiento de este protocolo cuando se desea transmitir datos. Cuando la terminal ó computadora se enciende, establece (es decir, pone en 1 lógico) la línea teminal de datos preparada (pin 20). Cuando el módem se enciende, establece la línea conjunto de datos preparados (pin 6). Cuando el módem detecta una portadora en la línea telefónica establece la línea detección de portadora (pin

8). La petición de envío (pin 4) indica que la terminal quiere enviar datos. Libre para enviar (pin 5) indica que el módem está preparado para aceptar datos. Los datos se transmiten por el circuito transmitir (pin 2) y se reciben por el circuito recibir

(pin 3). Además de estos, se disponen de otros circuitos para seleccionar la velocidad de transmisión de los datos, probar el módem, sincronizar los datos, detectar señales de llamada, y enviar datos en dirección contraria por un canal secundario.

**5.1.4.2.4 Circuitos Principales.** Las señales que se intercambian entre el terminal y el módem en el proceso de una comunicación son las siguientes:

- **GND:** Contacto 1. Tierra de protección (*Protective Ground*). Este contacto se utiliza para apantallar un cable protegido, de forma que se minimicen las interferencias producidas en ambientes con alto nivel de ruido. La referencia común para todos los contactos no es éste, sino el contacto 7.
- **SG:** Contacto 7. (*Signal Ground*). Es la referencia de todas las señales de la interfaz, incluidas las señales de datos, señales de reloj y señales de control. La tensión de esta señal siempre debe ser 0 volts. En teoría los contactos 1 y 7 deben ser independientes pero en la práctica están unidos formando una señal de tierra común.
- **TD (TXD):** Contacto 2. (*Transmitted Data*). Este circuito es el utilizado para transmitir las señales de datos desde el equipo terminal (*DTE*) al módem (*DCE*). Cuando no se transmite ningún dato, esta señal debe mantener la señal lógica 1. Para que el terminal pueda transmitir datos por el contacto 2, los circuitos *RTS*, *CTS*, *DSR*, y *DTR* deben tener una tensión alta
- **RTS:** Contacto 4. (*Request To Send*). La señal de este circuito es enviada desde *DTE* al *DCE* para preparar al módem para la transmisión. Una vez hecho esto y antes de empezar a transmitir datos, el terminal debe recibir la señal *CTS* por el contacto 5. Ambas señales, *RTS/CTS*, también pueden ser utilizadas para controlar el flujo de datos entre el módem y el terminal. Para que estas señales puedan ser reconocidas como indicadores de flujo de datos. Tanto el módem como el software de comunicaciones deben ser configurados para mantener un control de flujo *RTS/CTS*, también llamado control de flujo hardware. Cuando un módem opera de forma asíncrona, el software de

comunicaciones suele mantener la señal *RTS* constantemente en alto, indicando que el módem puede enviar datos al terminal en cualquier momento.

- **CTS: Contacto 5. (*Clear to Send*).** Se utiliza para indicarle al terminal que el módem está listo para transmitir. El módem activa esta señal después de que el terminal active su señal *RTS*. También puede ser utilizado junto con *RTS* como control de flujo de datos entre el terminal y el módem. Al igual que con la señal *RTS*, para que *CTS* pueda ser reconocido como indicador del flujo de datos, tanto el módem como el software de comunicaciones deben ser configurados para mantener un control de flujo *RTS /CTS*.
- **CD: Contacto 8. (*Carrier Detect*).** Detector de portadora. Una señal en este circuito le indica al terminal que el módem está recibiendo una señal de portadora del módem remoto. La señal de portadora tiene que estar presente durante todo el tiempo que dure la comunicación, se transmitan datos o no. Por tanto, si el terminal no detecta la señal *CD*, dará por terminada la comunicación por pérdida de portadora. En este caso el software de comunicaciones dará un mensaje similar a pérdida de portadora (*Carrier lost*) para indicar esta condición.
- **RD (RXD): Contacto 3. (*Received Data*).** Los datos que va demodulando el módem los envía al terminal por este contacto. Si el módem no tiene ningún dato que enviar al terminal, debe mantener este circuito en estado no activo (*OFF*, estado binario 1).
- **DSR: Contacto 6. (*Data Set Ready*).** La señal de este circuito indica el estado del módem. Cuando está activo (valor lógico 0). Indica que el módem está conectado a la línea telefónica y está listo para transmitir datos. También es utilizado por el módem para indicar que ha terminado un proceso de auto revisión ó que la marcación del número telefónico ha sido efectuada con éxito.
- **DTR: Contacto 20. (*Data Terminal Ready*).** Cuando está activa, se le indica al módem que el terminal está encendido y listo para una comunicación. Si no está activada la señal, el módem cortará cualquier comunicación que esté en curso. Este circuito controla el aspecto concerniente a la conexión del módem a la línea telefónica.
- **RI: Contacto 22. (*Ring Indicator* ).** Este circuito le indica al terminal que está siendo recibida una señal de llamada por el canal de comunicaciones. Este circuito es utilizado

por aquellos módems que están en modo respuesta automática, para indicarle al terminal que se está recibiendo una llamada. En respuesta a esta señal de llamada, el terminal le pasa una tensión al contacto 20 (DTR). Esta tensión le dice al módem que descuelgue y atienda la llamada.

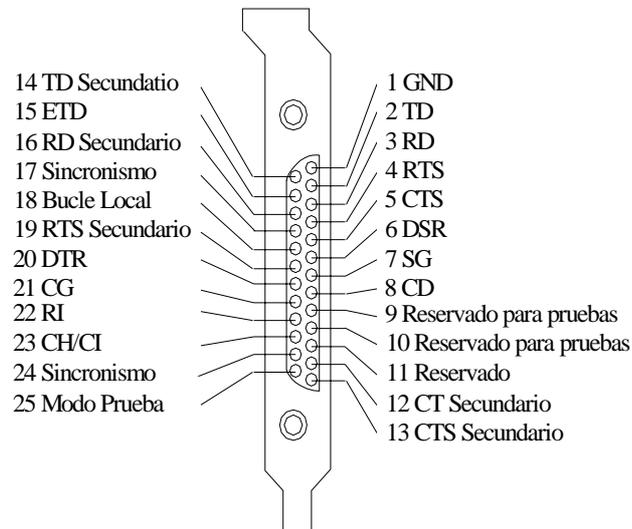
- CG: Contacto 21. (*Quality Detector*). Las señales de este circuito son transmitidas desde el módem al terminal siempre que el módem detecta una alta probabilidad de error en la recepción de los datos debido a una mala calidad de la línea. Este circuito permanece en estado activo cuando la calidad de la señal es aceptable, cambiando al estado inactivo si es inadecuada la calidad.
- CH/CI : Contacto 23.(*Data Signal Rate Selector*). Cuando el módem detecta una mala calidad de la línea y desactiva la señal CG, si este estado es mantenido durante un tiempo predeterminado, el terminal puede indicarle al módem que cambie su velocidad de operación por una más baja. Para hacer este cambio de velocidad se utiliza el contacto 23. El terminal pone el contacto 23 en estado activo para una velocidad de operación más elevada y lo pone en estado inactivo para una velocidad de operación más baja. Esta decisión de cambio de velocidad también puede ser tomada por el módem. Cuando es el terminal quien selecciona la velocidad de operación, la señal del contacto 23 va del terminal al módem, y el circuito es conocido como circuito CH. Si es el módem quien determina la velocidad de operación, la señal de contacto 23 va del módem al terminal el circuito es conocido como circuito CI.

#### **5.1.4.3. TIPOS DE PUERTOS SERIE EN UN COMPUTADOR**

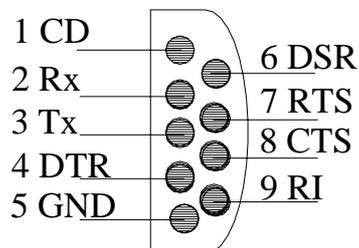
Hasta ahora, se ha descrito la comunicación serial entre un computador y un módem teniendo en cuenta el estándar RS-232, el cual considera la existencia de un conector de 25 conectores; no obstante los computadores de tipo PC disponen de dos clases de conectores de puerto serie:

1. Conector tipo D de 25 contactos (conector D25). (Figura 5.17)
2. Conector tipo D de nueve contactos (conector D9). (Figura 5.18)

Desde el punto de vista de las comunicaciones asíncronas, ambos conectores son idénticos y realizan las mismas funciones, ya que los nueve contactos del conector pequeño son precisamente los nueve contactos usados en cualquier comunicación asincrona.



**FIGURA 5.17** Conector DB25



**FIGURA 5.18** Conector DB9. Para la comunicación con el puerto serial, los pines 1,7,8 son conectados estableciendo una conexión común al igual que los pines 4 y 6.

Debido a que la comunicación no se realiza con un módem externo, aquellos conectores cuyo uso es específico de dichos dispositivos se configuran de tal manera que permitan un estado de comunicación permanente con el hardware, estableciendo un enlace común entre ellos que emule un módem listo para transmitir y recibir (Figura 5.18).

## **5.2. SOFTWARE PARA EL MODELAMIENTO Y PRESENTACIÓN GRÁFICA DE OBJETOS TRIDIMENSIONALES**

Como fue planteado desde capítulos precedentes, en la implementación de una herramienta de software para la generación automática de modelos gráficos tridimensionales a partir de objetos reales digitalizados, se requiere de la intervención de múltiples técnicas pertenecientes a diferentes disciplinas en el contexto de las ciencias de la computación.

En las siguientes secciones se hará una descripción resumida del proceso de desarrollo de la aplicación para la obtención del modelo geométrico del objeto digitalizado y su posterior reconstrucción gráfica en pantalla, haciendo uso de un enfoque sistémico para la descomposición del problema en sus aspectos fundamentales. El propósito es obtener una descripción de los procesos fundamentales para luego su posterior implantación, partiendo de las características particulares de cada uno de ellos.

### **5.2.1. Descripción del Sistema**

El proceso que va desde la interpretación geométrica de los datos registrados por el hardware hasta la presentación en pantalla del modelo involucra la implementación de múltiples técnicas para la solución al problema cinemático directo, la generación automática de mallas poligonales y la generación de una representación gráfica convincente del modelo tridimensional. Para hacer una aproximación coherente y manejable a la solución del problema es necesario realizar una clasificación del conjunto de procedimientos y rutinas que llevan a cabo éstas tareas según el campo de acción al que pertenezcan, entregando a cada módulo sus propias tareas responsabilidades. A continuación se presenta un enfoque sistémico para el análisis del problema

#### **5.2.1.1. Entradas y Salidas.**

El proceso de registro de posiciones realizado por el hardware y capturado por el software de comunicaciones en el computador entrega un conjunto secuencial de registros, cada uno de los cuales representa la posición angular de cada enlace, respecto al sistema de referencia de la articulación que lo une con el enlace inmediatamente anterior, para cada

punto rastreado por el efector final de la herramienta sobre la superficie del objeto. Este lista de registros, unida a los parámetros geométricos de la configuración física de la herramienta de ubicación de coordenadas, constituye la entrada del sistema cuando el modelo tridimensional se genera a partir del proceso de digitalización utilizando el hardware elaborado con dicho propósito. Una alternativa consiste en introducir al sistema un conjunto de puntos generado manualmente, en cuyo caso la entrada al sistema será la estructura de datos que contiene al conjunto.

El principal requerimiento del sistema, planteado en objetivos del proyecto, es la obtención de una representación gráfica en pantalla del objeto digitalizado que permita su visibilidad desde diferentes ángulos. También podría esperarse que la aplicación permitiera el almacenamiento del modelo geométrico obtenido a partir de la digitalización del objeto para su visualización posterior o posible utilización en otras aplicaciones. Estos resultados conforman las salidas del sistema. Es necesario aclarar que la rotación efectuada sobre la vista del objeto es también una entrada, aunque se diferencia de las otras ya que no interviene en la generación del modelo geométrico y sólo tiene significado para el módulo de despliegue en pantalla del objeto como se explica más adelante. Dado que la interacción entre el hardware y el software se realiza en una sola vía, el concepto de retroalimentación no tiene relevancia práctica en este sistema. Una representación elemental del sistema se muestra en la Figura 5.19.

# Software en el PC

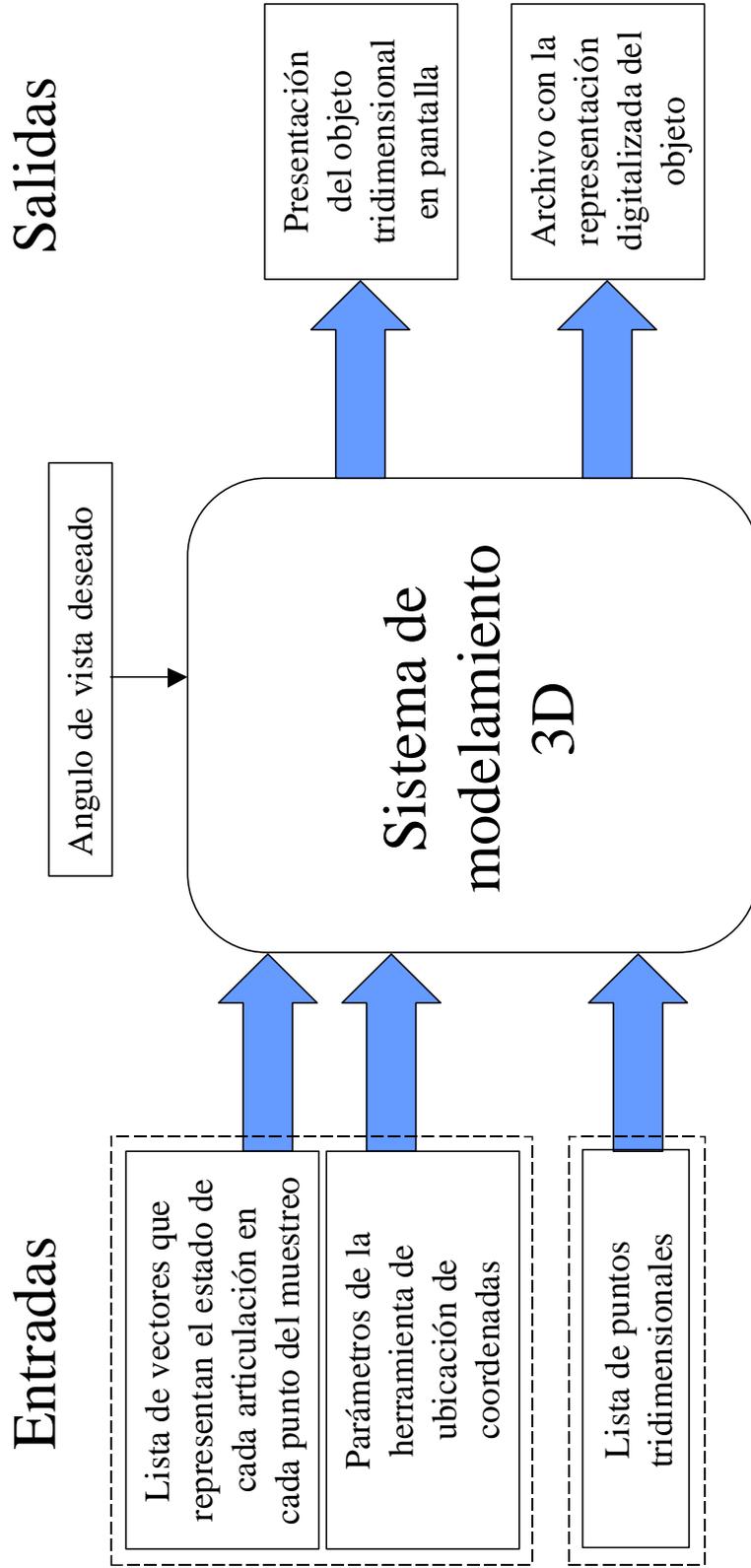


Figura 5.19 Modelo sistémico del software de modelamiento tridimensional

# Módulos del sistema de Modelamiento Tridimensional

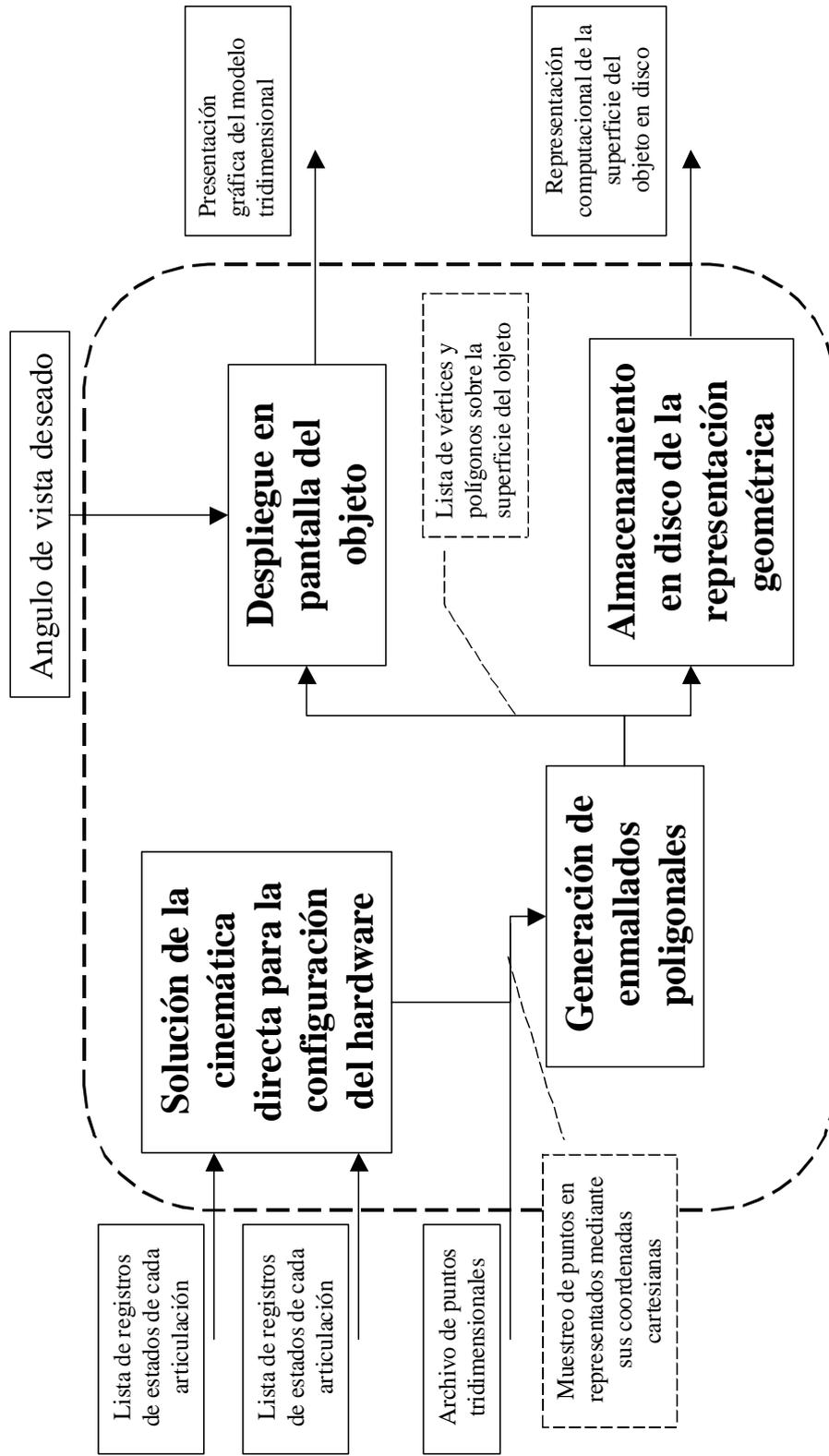


Figura 5.20 Diagrama flujo de datos entre los módulos de la aplicación.

### **5.2.1.2. Módulos principales de la aplicación**

La determinación del alcance y responsabilidades de cada módulo perteneciente al sistema propuesto tiene una estrecha relación con la naturaleza de cada problema particular expuesto en los capítulos anteriores. En un sistema de software que cumpla con los requerimientos expuestos anteriormente se pueden identificar cuatro módulos principales asociados al carácter de las tareas que deben ejecutar en el marco de la funcionalidad del sistema como un todo. El primero se encarga de la aplicación de la cinemática directa sobre la lista de registros de posición obtenida por el hardware. El segundo corresponde a la generación automática de enmallados poligonales con base en el muestreo resultante. El tercero se encarga de la utilización del modelo geométrico obtenido para la representación gráfica del objeto digitalizado. El cuarto permite el almacenamiento en disco de la representación computacional de la superficie del objeto para su posterior utilización. En la Figura 5.20 se aprecia un esquema de flujo de datos entre los diferentes módulos del sistema, que serán descritos con más detalle a continuación.

#### **5.2.1.2.1. Solución al problema cinemático directo para la configuración del hardware.**

Este módulo (figura 5.21) recibe una lista de registros obtenidos en el proceso de digitalización del objeto y la configuración geométrica de la herramienta de ubicación de coordenadas para las transformaciones geométricas, como fue descrita en la sección 5.1.1. Cada registro de la lista contiene cinco campos, en los cuales se almacena el valor del contador que representa la posición angular de cada enlace de la herramienta respecto al origen del sistema de coordenadas ubicado la articulación que lo une con el enlace anterior. Estas posiciones son convertidas a valores angulares teniendo en cuenta la resolución de los encoders (en grados por pulso) y la longitud de cada enlace. Posteriormente se aplican de forma secuencial las transformaciones de rotación y cambio de sistema para cada enlace de la herramienta, obteniéndose las coordenadas cartesianas para cada punto correspondiente a cada registro de posiciones relativas, generándose como salida una lista de puntos tridimensionales representados mediante sus coordenadas.

# Módulo de solución al problema cinemático directo

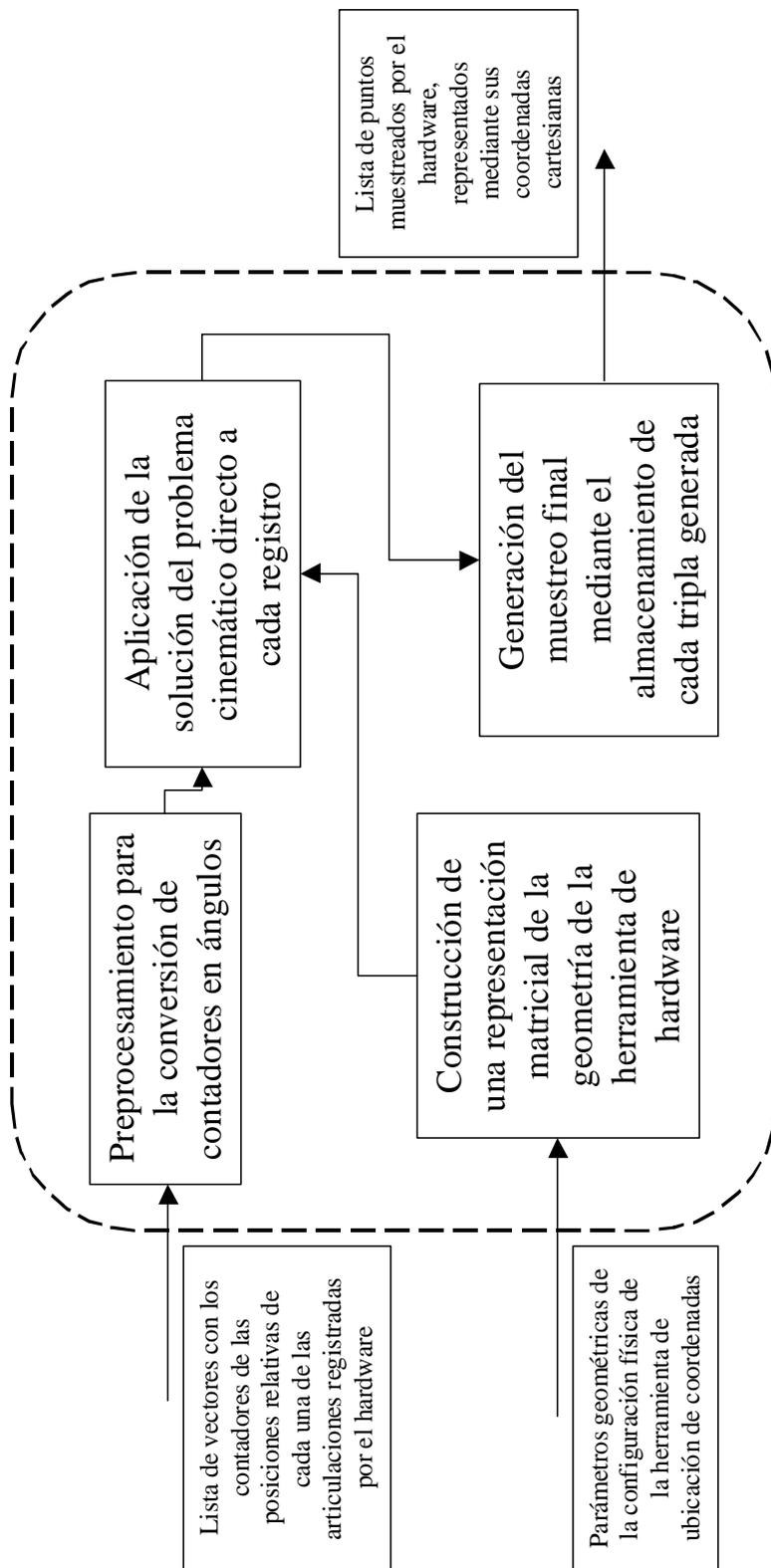


Figura 5.21. Diagrama de bloques del módulo de cinemática directa.

# Módulo de generación de enmallados poligonales

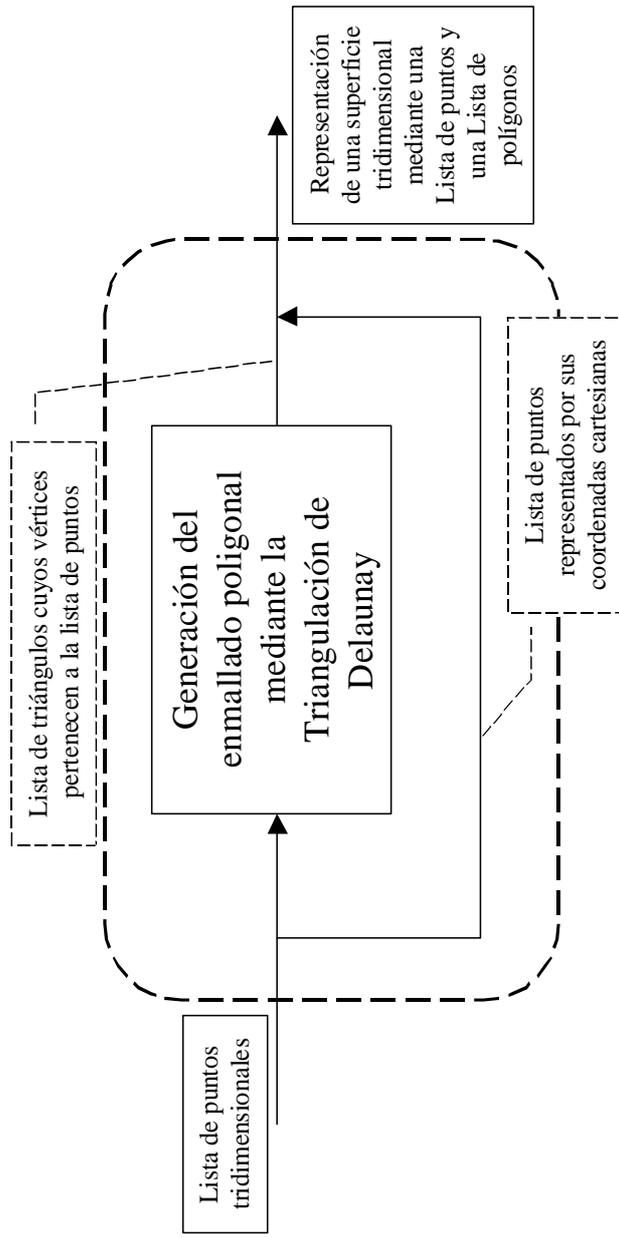


Figura 5.22. Diagrama de bloques del módulo de generación automática de mallas poligonales.

**5.2.1.2.2. Generación de enmallados poligonales.** La entrada a este módulo (Figura 5.22) es la lista de puntos en el espacio tridimensional, producto de la aplicación de la cinemática directa o de la obtención de la lista desde un archivo de puntos. Por medio de un algoritmo de fuerza bruta que hace uso del criterio de la esfera vacía se construye la Triangulación de Delaunay en 2.5 dimensiones para la proyección sobre el plano  $XY$  del muestreo contenido en la lista de puntos. El resultado de este algoritmo es una lista de triángulos (cuyos vértices son definidos mediante índices que apuntan a la lista de puntos), el cual representa un enmallado triangular que aproxima la superficie del objeto tridimensional digitalizado. La salida de este módulo es una representación geométrica de una región de la superficie del objeto tridimensional, que comprende una lista de vértices (equivalente a la lista de puntos de la entrada) asociada a la lista de triángulos generada por el algoritmo de triangulación.

**5.2.1.2.3. Despliegue en pantalla del modelo tridimensional.** Dada la representación geométrica de la superficie del objeto construida por el módulo de generación automática de enmallados, el proceso de despliegue utiliza dicha representación para presentar en pantalla una reconstrucción gráfica del objeto (según el ángulo de rotación respecto del plano de vista definido por la pantalla introducido por el usuario), utilizando algunas de las técnicas más comunes de la graficación por computador aplicada a modelos tridimensionales. En la figura 5.23 se muestra la interacción de las diferentes rutinas de procesamiento gráfico utilizadas por la aplicación, según el esbozo realizado en el capítulo cuatro. Inicialmente, sobre la lista de vértices se aplica la transformación geométrica de rotación, permitiendo que el modelo pueda girar sobre el origen de coordenadas según el ángulo introducido por el usuario mediante las teclas de control. Las nuevas ubicaciones de los vértices girados no alteran sus relaciones de pertenencia a los triángulos, pues éstas están definidas mediante índices, conservándose la morfología del objeto. Las aristas de los triángulos son entonces discretizadas mediante el algoritmo de Bresenham, guardándose en memoria una estructura que contiene las posiciones de los puntos que conforman los contornos de cada polígono en su proyección sobre el plano de observación, identificado con el área de dibujo equivalente a la matriz de color en el enfoque  $Z$ -Buffer. Estos contornos son utilizados posteriormente para generar líneas de relleno horizontal, que ejecutan un barrido sobre cada pixel interior al polígono, aplicando simultáneamente a cada

pixel los procesos de detección de superficies visibles y aplicación de iluminación y sombreado asociados al objeto. Por último se ejecuta procedimiento de transferencia de la imagen en memoria directamente a la pantalla, el cual está generalmente asociado a las rutinas gráficas que provee el sistema operativo.

**5.2.1.2.4. Almacenamiento en disco de la representación geométrica del modelo.** Este módulo, a pesar de ser independiente de los anteriores y encontrarse al mismo nivel jerárquico, se encarga de crear un archivo en disco conteniendo los aspectos fundamentales de la representación geométrica de un modelo generado por el sistema, para su posterior utilización. Ésta tarea es meramente operativa y no posee mayor interés.

# Módulo de despliegue en pantalla

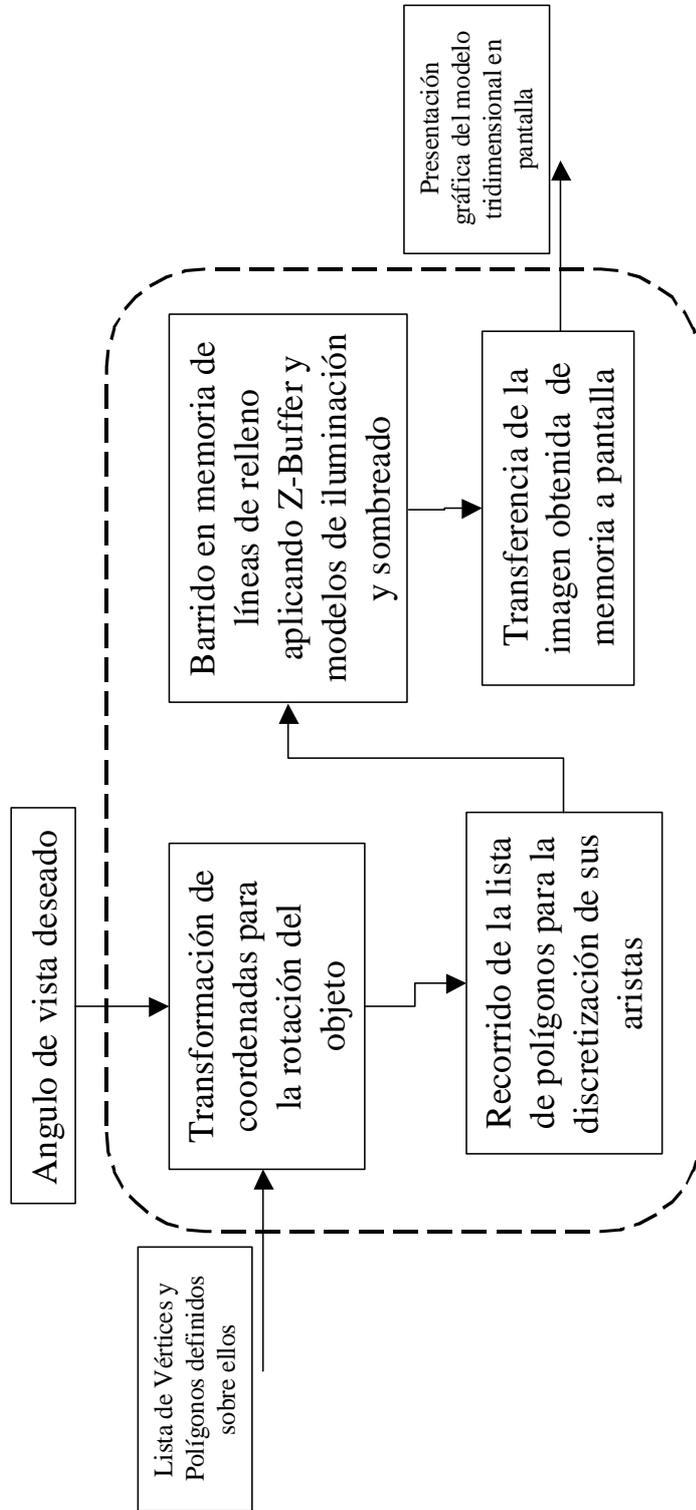


Figura 5.23. Diagrama de bloques del módulo de despliegue en pantalla.

### **5.2.2. Aspectos generales de la implementación**

La herramienta de software desarrollada se ejecuta sobre el entorno Win32. Para la implantación del sistema se escogió el lenguaje C++ y como herramienta de desarrollo se utilizó el entorno integrado de desarrollo de Borland C++ Builder v.3.0.

El lenguaje C++ permitió la abstracción de las primitivas geométricas (puntos, vectores, aristas, polígonos, esferas circunscritas) como instancias de clases especializadas, si bien la metodología de análisis y diseño orientado a objetos no fue aplicada de manera rígida. Dicha abstracción se plasmó en la construcción de una librería de objetos y funciones geométricas, diseñada con cierto grado de generalidad para su utilización simultánea por los diferentes módulos de la aplicación y para aplicaciones desarrolladas posteriormente.

En el directorio de código fuente del CD adjunto contiene todo el código fuente y archivos asociados al proyecto de la aplicación, en formato de C++ Builder v.3.0., además un programa de instalación, el cual contiene, además de la última versión de la aplicación, las librerías necesarias para su ejecución en sistemas del tipo Win32 instalados apropiadamente.

## 6. RESULTADOS Y CONCLUSIONES

Existen varios aspectos que permiten medir en cierto grado, el desempeño en el funcionamiento de una herramienta que logra reproducir la imagen tridimensional de un objeto en computador mediante la captura de puntos sobre la superficie de este. El primero de ellos y el más importante es la resolución de la herramienta, entendiéndose como la capacidad que tiene para distinguir puntos que logran dar una forma muy aproximada del objeto digitalizado con la imagen desplegada en la pantalla del computador. Nótese que la distinción entre dos puntos muy cercanos ubicados sobre la superficie de un objeto, e importantes para la reconstrucción del mismo, depende de la capacidad que el sensor de rotación tiene para identificar un determinado desplazamiento angular y de la longitud de los enlaces utilizados por la herramienta de ubicación de coordenadas.

### 6.1. DESEMPEÑO DEL HARDWARE.

A continuación se evaluarán algunos parámetros de desempeño observados en la herramienta desarrollada para la obtención y registro de puntos en el espacio.

#### 6.1.1. Maniobrabilidad de la Herramienta de Ubicación de Coordenadas

El análisis anterior permite determinar una serie de pasos para realizar el sensado de puntos, logrando un cierto grado de precisión en la ubicación de estos dado que existen diversas trayectorias a través de las cuales se puede acceder a un mismo punto y que la medición de los ángulos está restringida a la resolución de los sensores.

En lo posible el barrido de los puntos en el objeto debe posicionar inicialmente la articulación  $A_0$  y mediante los demás enlaces, capturar la mayor cantidad de puntos en esta posición. Una vez hecho esto, se reubica la articulación y se realiza el mismo

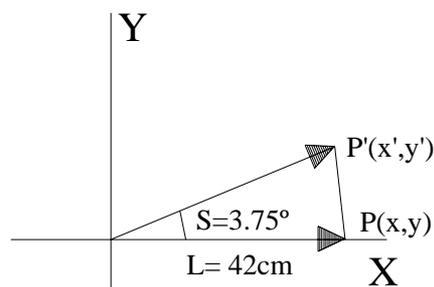
procedimiento. El objetivo es capturar los puntos más sobresalientes en la superficie del objeto, mediante el menor número de trayectos posibles. Esto logra disminuir el error en la diferenciación de los puntos al resolver el problema cinemático directo.

### 6.1.2. Lectura de Datos.

Trabajar con una frecuencia de reloj de 4MHz garantiza que el ciclo de exploración de los puertos que reciben las señales digitalizadas desde los sensores se realice de forma tal que no se pierda ningún cambio de estado. Aunque esta exploración se puede ver afectada por un incremento en la velocidad de desplazamiento con que se muevan los enlaces, el microcontrolador logra registrar todos los cambios de estado en todos los sensores para movimientos relativamente rápidos (90° por segundo aproximadamente)

### 6.1.3. Resolución.

Como se presentó en la sección 5.1.2, el sensor de rotación puede detectar como mínimo un desplazamiento angular de 3.75 grados y la máxima longitud que la herramienta puede alcanzar para la ubicación de un punto (*extremos del espacio de trabajo*) corresponde a la suma de los enlaces  $L_1$ ,  $L_2$  y  $L_3$  cuando estos se encuentran extendidos perpendicularmente al eje Z. Teniendo en cuenta que en el caso anterior se debe realizar el mayor desplazamiento para detectar un cambio en las articulaciones  $A_0$  y  $A_1$ , se puede calcular cual es esa longitud de la siguiente manera (Figura 6.1)



**FIGURA 6.1** Desplazamiento máximo para registrar 3.75 grados

Dado el enlace  $L$  que equivale a la suma de los enlaces  $L_2$ ,  $L_3$  y  $L_4$  de la herramienta articulada y cuyo punto final es la ubicación del efector final con coordenadas  $P(x,y)$ , la

posición del efector final al girarlo un ángulo  $S=3.75^\circ$  es  $P'(x',y')$ . Aplicando la fórmula de la distancia tenemos:

$$d = \sqrt{(y'-y)^2 + (x'+x)^2}$$

En donde

$$y' = 42 \operatorname{sen}(3.75^\circ)$$

$$x' = 42 \operatorname{cos}(3.75^\circ)$$

En donde  $P' = (41.91, 2.74)$ . Entonces la distancia entre esos puntos es

$$d = \sqrt{(2.74 - 0)^2 + (41.91 - 42)^2}$$

$$d = \sqrt{7.50 + 0.0064}$$

$$d = 2.7 \text{ cm}$$

Realizando el mismo procedimiento con el enlace  $L_3$  se puede concluir que el error mínimo y máximo en estos casos particulares es respectivamente de 1.01 cm y 2.27 cm. La error promedio observado en el desempeño de la herramienta se aproxima a 1.6 cm.

## 6.2. DESEMPEÑO DEL SOFTWARE

### 6.2.1. Desempeño del proceso de triangulación

La Triangulación de Delaunay, restringida a la generación de un enmallado poligonal en 2.5 dimensiones (sección 5.2) demostró ser una técnica satisfactoria para muchos casos de objetos tridimensionales cuya superficie puede ser definida como una función de dos variables continua y derivable de contorno convexo. Un estudio posterior para el modelamiento de objetos con morfologías más complejas podría partir de la exploración de enfoques diferentes.

### 6.2.2. Desempeño del proceso de despliegue.

La aplicación desarrollada contiene la implementación de dos alternativas para la ejecución del proceso de despliegue. La primera fue desarrollada a partir de los conceptos de graficación de modelos tridimensionales por computador expuestos en el capítulo 4, en

tanto que la segunda hizo uso de la librería gráfica de carácter abierto OpenGL. El desempeño del proceso de despliegue en OpenGL es bastante eficiente, ya que sus rutinas han sido optimizadas al máximo, a la par que muchas de sus funciones acceden directamente al hardware de video. Sin embargo dicha implementación implica sacrificios en la calidad del resultado, que se hacen evidentes al comparar la implementación del sombreado especular para muestreos con un número reducido de polígonos, en los cuales las rutinas desarrolladas a partir del modelo teórico presentaron resultados visiblemente más aproximados al resultado ideal. Perspectivas de trabajo futuro sobre el software de despliegue desarrollado envuelven aspectos como la solución de algunos pequeños defectos producidos por la interpolación lineal aplicada en los procesos de sombreado, además de la optimización de las operaciones matriciales y la utilización de rutinas de dibujo más eficientes.

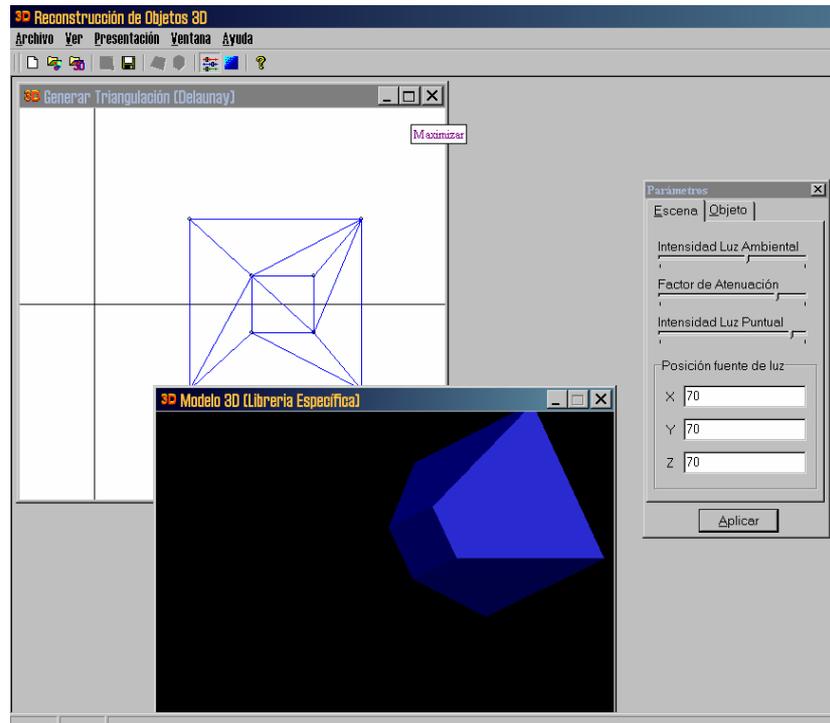
### 6.3. Ejemplos.

Como prueba del funcionamiento de la herramienta fueron digitalizados algunos objetos de morfología sencilla, permitiendo la observación del desempeño del sistema

El modelo de prueba, presentado en la Figura 6.2 fue digitalizado, obteniéndose la representación geométrica mostrada en la figura 6.3

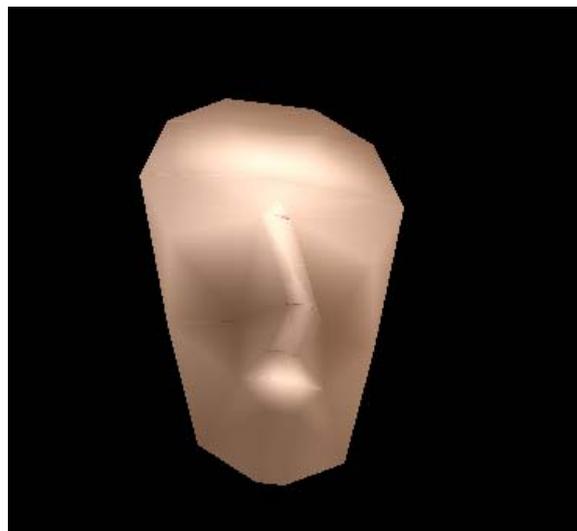


**Figura 6.2.** Proceso de digitalización de un modelo 3D.



**Figura 6.3.** Modelo obtenido mediante la digitalización de un objeto real.

La Figura 6.4 expone el resultado de los procesos de renderizado aplicados a un objeto creado modelado a partir de un archivo de puntos sin digitalización de un objeto real.



**Figura 6.4.** Modelo tridimensional obtenido mediante un archivo de puntos creado manualmente

## 8. BIBLIOGRAFÍA

Preparata, Franco y Schamos, Michael. *Computational Geometry, an Introduction*. Springer-Verlag. Nueva York, Estados Unidos. Segunda Edición, 1988.

George, Paul-Louis y Borouchaki, Houman. *Triangulation de Delaunay et Maillage, Applications aux éléments finis*. Editions Hermes. Paris, Francia. Primera Edición, 1997.

Foley, James, Van Dam Andries, Feiner, Steven y Hughes, John. *Computer Graphics, Principles and Practice*. Addison-Wesley. Wilmington, Estados Unidos. Segunda Edición, 1990.

Hearn, Donald y Baker, M. Pauline. *Gráficas por Computadora*. Prentice Hall Hispanoamericana. Nauvalcapán, México. Segunda Edición, 1995.

Owen, Steven. *A Survey of Unstructured Mesh Generation Technology*. Department of Civil and Environmental Engineering, Carnegie Mellon University. Pittsburgh, Estados Unidos. 1998.

Shewchuk, Jonathan. *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. School of Computer Science. Carnegie Mellon University. Pittsburgh, Estados Unidos. 1996.

Rumbaugh, James, Blaha, Michael, Premerlani, William, Eddy, Frederick y Lorensen William. *Modelado y Diseño Orientado a Objetos*. Prentice Hall. Madrid, España. Primera Edición, 1991.