UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

UN SISTEMA PARA ANIMACION DE DIBUJOS ESQUEMATICOS POR COMPUTADORA

TESIS QUE PARA OBTENER EL TITULO DE MATEMATICO

PRESENTA:

GERARDO LEON LASTRA

MEXICO, D.F. SEPTIEMBRE 1976.

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A HELY

A MI PAPA

Y HERMANOS

EN MEMORIA DE

MAMA

ANDRES Y JUAN

A MI MAESTRO Y AMIGO

DR. ENRIQUE CALDERON A.

I N D I C E

Prólogo

I.- Descripción de la tecnología disponible 1

1.- La pantalla y el generador de vectores 1

2.- La tableta 5

II.- Un algoritmo para aproximar curvas en el plano 8

III.- Hacia un sistema de animación 13

1.- Descripción de los dibujos 13

2.- La animación 14

IV.- El sistema 17

1.- Descripción general del sistema 17

2.- La estructura de datos 19

3.- Los procedimientos para describir el dibujo inicial y el dibujo estático 25

4.- El procedimiento para describir el dibujo final 31

5.- El procedimiento genera imágenes 32

6.- Funcionamiento global del sistema 35

Apéndice.- El algoritmo de Chaikin 41

Bibliografía 46

P R O L O G O

> Sabiéndote gran admirador de las investigaciones matemáticas, he - creído mi deber comunicarte las - particularidades de cierto método que podrás utilizar para descu - brir mediante la mecánica determi nadas verdades matemáticas... Mu- chas veces, en efecto, he descu - bierto gracias a la mecánica pro- posiciones que he podido demostrar después por geometría; el método en cuestión no constituye una de- mostración verdadera, pero es más sencillo, una vez adquirido median te aquel cierto conocimiento del tema, encontrar a renglón seguido la demostración, que si esta se - buscase sin ninguna noción estable cida de antemano.
> (Arquímedes: "Carta a Eratóstenes")

Al intentar realizar dibujos animados para la enseñanza de las matemáticas, se enfrentan básicamente dos problemas; uno se - refiere a los aspectos pedagógicos y de comunicación del tema a - tratar en un film, el otro a la fabricación de las imágenes. El presente trabajo da una solución parcial al segundo problema.

No intento aquí hablar sobre dibujos animados y educación de las matemáticas; solo me atrevo a este respecto a hacer una re flexión: si por más de dos mil años de civilización la figura geo

métrica ha permanecido fija y rígida, y ahora se nos presenta la realidad de poder moverla y deformar "continuamente", ¿acaso no es hora que empecemos a buscar un nuevo simbolismo geométrico y un nuevo desarrollo de la comunicación icónica, así como nuevas posibilidades en la educación?.

David Hilbert, que ha contribuido tanto o más que nadie a la edificación analítica de la matemática moderna, se expresa como sigue en el prefacio de su libro "Geometría y la Imaginación": . . . pienso que una representación a grandes rasgos de la geometría, basada en la intuición visual, debería contribuir a una apreciación más correcta de la matemática para un público más amplio que el de los especialistas.

Como ya he mencionado un problema para "hacer" geometría en la pantalla es la elaboración masiva de dibujos.

La graficación por computadora surge ahora como herramienta para la superación de este problema; y como suele suceder, el dar esta solución trae consigo nuevos problemas.

La graficación por computadora ha venido desarrollándose desde la década pasada y ha pesar de los esfuerzos realizados no existe un lenguaje poderoso orientado a la graficación dinámica.

Se han dado respuestas parciales al extender con subrutinas de graficación lenguajes como Fortran y Algol; y lo que mayor efectividad ha tenido es el uso de sistemas de propósito particular, como por ejemplo orientados a la cartografía, la arquitectura, el diseño, etc.

En este trabajo se describe un sistema interactivo que permite generar y animar dibujos esquemáticos; el cual puede ser utilizado por una persona que carezca de conocimientos de computación.

En el "Centro de Procesamiento y Evaluación Dr. Arturo Rosenblueth", en donde he trabajado, ha sido posible implementar una primera versión del sistema aquí descrito en una minicomputadora Imlac PDS-4, lo que me ha permitido introducirle mejoras y que por el momento solo se encuentran en este trabajo.

Quiero aprovechar para expresar mi agradecimiento al Dr. Guillermo Torres Díaz por su contribución a mi formación profesional; así mismo a mi maestro Humberto Madrid de quien obtuve una gran motivación al iniciar mis estudios profesionales.

Agradezco al Arq. Hector Carranza P. las facilidades que me brindó para la realización de este trabajo.

CAPITULO I.- DESCRIPCION DE LA TECNOLOGIA DISPONIBLE

A continuación se describen a groso modo los dispositivos de entrada y salida con los que cuenta la máquina en que se ha implementado una primera versión del trabajo descrito en los capítulos siguientes.

1.- La pantalla y el generador de vectores.

Conectado a una minicomputadora convencional se tiene un tubo de rayos catódicos el cual constituye un dispositivo de salida. La interface entre ambos dispositivos es mediante un convertidor - de señales digitales a analógicas que permite al usuario pensar que la pantalla del tubo de rayos catódicos consta de un enrejado de 2048 X 2048 puntos, los cuales son direccionables desde el procesador central, por parejas (a,b) de números enteros, en donde $0 \leq a,b \leq 2047$.

Este direccionamiento se traduce en apuntar el haz de electrones del tubo de rayos catódicos, al lugar en la pantalla que se desee.

La circuitería de la máquina dispone de lo que se conoce como "generador de vectores", el cual independientemente de la posi-

ción a que apunte el haz de electrones, toma instrucciones en el procesador, que se traducen en los incrementos simultáneos que ha de sufrir el haz a lo largo de las abscisas y ordenadas. Este movimiento puede efectuarse a voluntad del programador con o sin - bombardeo de electrones, lo que se traduce en trazar una recta y obtener un nuevo prosicionamiento, o en una nueva colocación del haz de electrones, respectivamente.

El paquete de programación provee al programador con el - juego necesario de instrucciones para hacer trabajar al generador de vectores, teniéndose para este último tres modos distintos de operación:

i) Vector largo - los incrementos pueden ser desde cero hasta 1024 unidades, y requiere de dos palabras de memoria para describir cada vector.

ii) Vector mediano - incrementos que van de cero a 64 unidades, requiere de una palabra por vector

iii) Vector corto - incrementos de cero a 3 unidades y usa de una palabra de memoria por cada dos vectores.

Se tiene entonces la posibilidad de unir dos puntos arbitrarios de la pantalla mediante una recta luminosa, habiendo dos restricciones en cuanto al número de rectas mostradas simultaneamente; la primera se refiere a la memoria necesaria para describir estas rectas, y la segunda en cuanto a la longitud total de estas rectas, y por ende al tiempo necesario que necesita el haz de electrones para hacer el recorrido total.

La manera de hacer visible una imagen que se tenga ya descrita en la memoria de la máquina es la siguiente: se carga en el acumulador la dirección de memoria en donde empieza la descripción de la imagen, y se ejecuta una instrucción que ordena al haz (es decir al tubo de rayos catódicos) que dibuje de acuerdo a la descripción que encuentre a partir de la dirección en el acumulador, el haz trabajará hasta toparse con una instrucción que le indica que se apague o desactive.

Este último procedimiento se conoce como "refrescamiento de la pantalla", y puede hacerse hasta cada cuarentavo de segundo.

Si la longitud total de lo que ha de refrescarse excede a 20 metros, entonces, puesto que la velocidad de movimiento del rayo de electrones incidiendo en la pantalla es de 400 metros por segundo, (aproximadamente), el ojo humano no tendrá la impresión de simul

taneidad y se percibirá un "flicker" o tintileo en la imagen.

Es importante mencionar que una vez dada la orden de refrescamiento, el programa en el procesador central no espera a que se termine de refrescar, sino que continúa adelante en su ejecución; de aquí que aprovechando el tiempo de refrescamiento para el cálculo de información necesaria para la descripción de una nueva imagen, surja la posibilidad de generación de imágenes que varíen poco de una a otra y obtener la sensación de movimiento continuo. Esto obviamente bajo la condición que el tiempo de cálculo no exceda a 1/20 de segundo aproximadamente. En el caso que este tiempo sea excedido el movimiento se puede recuperar por medio de técnicas de animación cinematográfica tradicionales. Aún más, disponiendo de un dispositivo que ordenara a una cámara de cine cuando debe fotografiar.

2.- La Tableta.

Pasemos ahora a describir el dispositivo de entrada más importante para la realización de este trabajo, el cual consta de una tableta y una pluma electrónica.

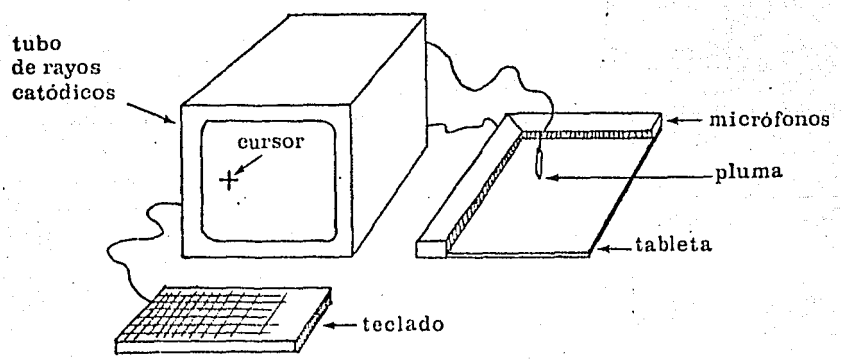

Figura 1.- Dibujo esquemático de dispositivos interactivos de entrada y salida del computador Imlac PDS-4

Estos dispositivos le dan al usuario gran capacidad de interacción gráfica, al dotarlo con la posibilidad de señalar, indirecta pero efectivamente, cualquier punto en la pantalla, de tal forma que un programa conozca las coordenadas del punto señalado.

La tableta tiene dos micrófonos a las orillas (vease fig. 1), - uno asociado al eje de las abscisas y otro al de las ordenadas, los cuales captan y miden las coordenadas de un pulso que emite la pluma electrónica; estas mediciones son congruentes con el enrejado de la pantalla, es decir, se tiene una división en regiones o puntos para la tableta, isomorfa al enrejado de la pantalla.

La forma que tiene el usuario de saber en donde señala con la - pluma, es mediante un cursor que ocupa la posición en la pantalla asociado al lugar en la tableta que señale la pluma.

Por otro lado, si se presiona la pluma contra la tableta se cerrará un circuito el cual prende una bandera en el procesador, esto - permite saber cuando se esta interesado por un punto en particular.

También es posible sensibilizar e identificar regiones en la pantalla. Esto consiste en indicar al procesador que vectores serán sensibles a un señalamiento con la pluma electrónica. Si en algún momen-

to coinciden las coordenadas del rayo de electrones, al trazar un vector sensible, con las de la pluma, se producirá una interrupción que indica la dirección en memoria que ocupa el vector sensible señalado. Lo anterior ofrece la alternativa de desplegar en la pantalla lo que se conoce como "menú" para la interacción, cuyo uso es análogo a teclear las diversas instrucciones por terminal, que acepta un sistema interactivo de tipo convencional.

CAPITULO II .- UN ALGORITMO PARA APROXIMAR CURVAS EN EL PLANO.

En este capítulo se describe un algoritmo para generar imágenes de curva, el cual juega un papel decisivo en la realización de este trabajo.

1.- Mediante las operaciones aritméticas, las funciones trascendentes y trigonométricas en punto flotante, es posible aproximar una infinidad de curvas, solo que el uso del punto flotante provoca que el tiempo de generación de poligonales que aproximen curvas sea largo y por otro lado se use mucha memoria principal (o central), puesto que no se dispone de circuitería para punto flotante. El algoritmo que aquí se describe no presenta los problemas mencionados.

Dadas tres parejas A,B,C de números enteros que especifiquen puntos de la pantalla, el algoritmo genera una "curva" que principia en A tangente al segmento AB, pasa por el punto medio de la mediana al triángulo ABC que pasa por B, y termina en C tangente al segmento BC (Ver figura 2).

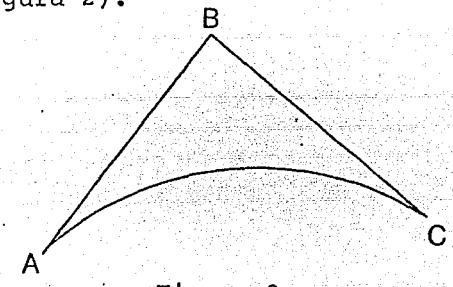

Figura 2

Para la implementación del algoritmo:

Asígnense las coordenadas de A a P_1 y las de C a P_4 y calcúlense nuevos puntos P_2 y P_3 como los puntos medios de los segmentos $\overline{AB}$ y $\overline{BC}$ respectivamente.

Empújense P_3 y P_4 a una torre de apilamiento* y recalcúlese un nuevo P_4 como el punto medio del segmento $\overline{P_2P_3}$; compárese la distancia del nuevo P_4 con P_1, en caso de ser mayor a $2\sqrt{2}$, recalcúlese P_3 como el punto medio de $\overline{P_2P_4}$ y P_2 como el punto medio de $\overline{P_1P_2}$.

Vuélvase a emplear el procedimiento hasta que P_4 se encuentre suficientemente cercano a P_1 (i.e. que su distancia sea menor o igual a $2\sqrt{2}$). Entonces genérese el vector de pantalla $\overline{P1P4}$, asígnese P_4 a P_1 y sáquense de la torre de apilamiento dos puntos llamándolos P_2 y P_4; bríncuese a la prueba de distancia. (ver figura 3).

* Lo que en Inglés se conoce como "push-down stack".

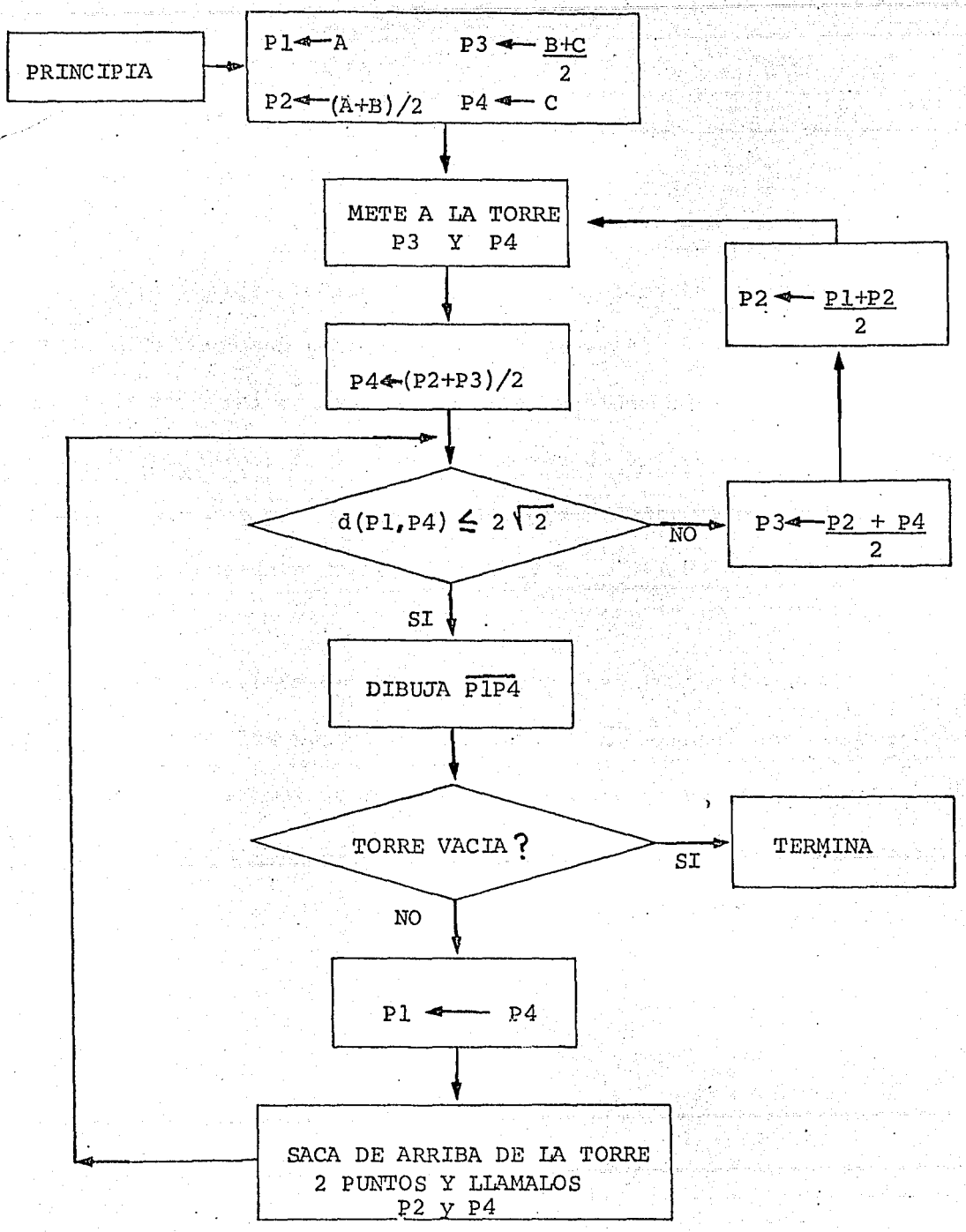

Figura 3.- Algoritmo de G. Chaikin para generar curvas.

Puesto que la división por dos en números enteros y en base dos se lleva a cabo a través de un corrimiento del punto decimal en un lugar hacia la izquierda; la implementación del algoritmo no requiere del uso de punto flotante.

En la implementación que se ha hecho para la IMLAC PDS-4, se usan solo vectores cortos, lo que representa un ahorro significativo de memoria; se programó en lenguaje de máquina y ocupa menos de 250 palabras, incluyendo una torre de apilamiento de 80 palabras y la rutina que genera los vectores. Por último hay que añadir que es sumamente rápido.

El uso repetido de este algoritmo puede generar curvas complicadas, las cuales pueden ser de apariencia lisa si son colineales los puntos paramétricos segundo y tercero de un primer arco - con los puntos paramétricos primero y segundo del arco a continuación (ver fig. 4) y además coinciden los puntos tercero y primero.

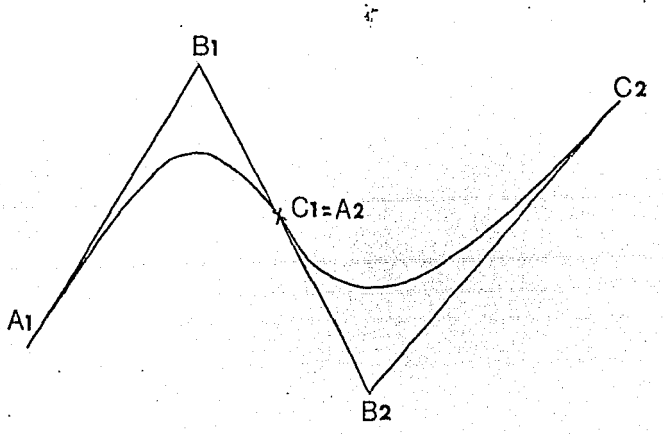

Figura 4

En el apéndice A de este trabajo, se encuentra una discusión sobre el tipo de curva al cual converge el algoritmo.

CAPITULO III .- HACIA UN SISTEMA DE ANIMACION.

1.- La descripción de los dibujos.

Dada la tecnología actual, es trivial aproximar un dibujo esquemático a base de rectas en un graficador, pero si este dibujo consta de curvas (ver fig. 5), ¿Cómo hacerlo de manera eficiente?.

Una respuesta es llamar sucesivamente el algoritmo descrito en el capítulo anterior (ver fig. 6), pasando por parámetros un punto de tangencia, la intersección de esta tangente con la siguiente y el siguiente punto de tangencia; en donde a mayor número de tangentes mayor aproximación. Cabe mencionarse que para generar una curva que se perciba como círculo, basta usar en la forma descrita un octágono regular.

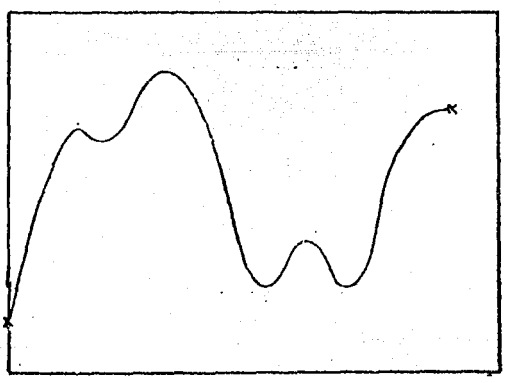

Figura 5

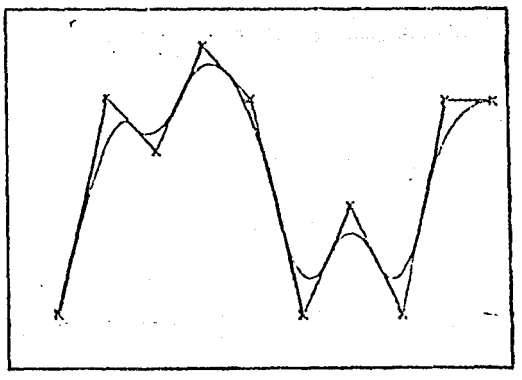

Figura 6

Dado que se dispone de la tableta y la pluma electrónica como dispositivos de entrada, no es necesario codificar las coordenadas de los puntos que describan a un dibujo, basta utilizar dichos dispositivos en forma interactiva para poder describir al objeto.

Los dibujos a base de rectas y curvas quedarán entonces descritos por dos puntos para cada segmento de recta y por tres para cada arco (suficientemente pequeño y) de curvatura de signo constante.

2.- La animación.

Tomando en consideración que el número de puntos necesarios para describir un dibujo esquemático, es pequeño en comparación a la complejidad que puede tener el mismo, surge la posibilidad de animación al variar lascoordenadas de los puntos descriptores con pequeños incrementos. Para precisar las ideas veamos la siguiente discusión:

Supóngase que la figura 7 es una componente de un dibujo más complejo, y que la figura 8 es la misma componente después que el dibujo ha sufrido la deformación esquematizada en la figura 10.

Figura 7

Figura 8

Para generar una sucesión de imágenes intermedias considérense los puntos descriptores A_1, A_2, A_3 y B_1, B_2, B_3 de estas figuras (ver fig. 9).

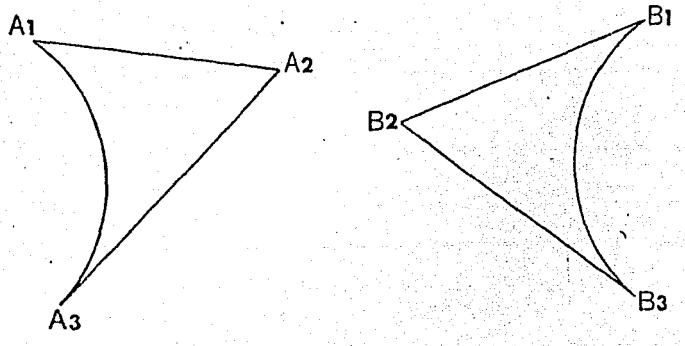

Figura 9

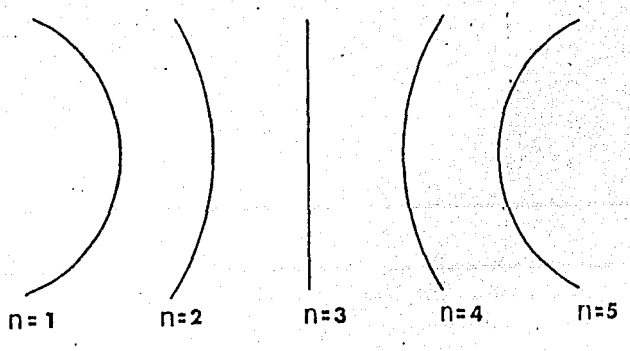

Figura 10

Si la deformación de uno en otro se deseara en n pasos, bastaría dividir cada uno de los segmentos $\overline{A_iB_i}$ en n-1 partes iguales, obteniendo así tres sucesiones de puntos cada una de las cuales principia en Ai y termina en Bi (i=1,2,3). Haciendo uso de estas sucesiones en forma obvia y si n=5 obtendríamos la sucesión de imagenes en la figura 10.

CAPITULO IV .- EL SISTEMA

Haciendo uso de los conceptos e ideas discutidos en los capítulos anteriores, se describe un sistema interactivo para la animación de figuras esquemáticas. Con objeto de optimizar el sistema, este ha de programarse en lenguaje ensamblador.

1.- Descripción general del sistema.

A groso modo, una vez que se ha inicializado el sistema, se solicita al usuario que describa la figura en su posición inicial, la cual en adelante denominaremos "dibujo inicial"; una vez terminada esta descripción, el sistema solicita lo mismo para la figura en posición final, la cual llamaremos "dibujo final". Al haber introducido ambas descripciones, las cuales han de ser a base de los mismos elementos (segmentos de rectas y curvas), introducidos en igual orden; se procede a hacer la animación, es decir a generar las imágenes intermedias. Generada la última imagen se vuelve a solicitar descripción de nuevos dibujos. (Ver fig. 11)

Existen opciones, una de ellas es la posibilidad de describir un objeto el cual aparecerá inmóbil durante la generación de imágenes animadas, al que denominaremos "dibujo estático"; otra opción es el poder

ciclar la generación de imágenes animadas, esto es, partiendo del dibujo inicial se generan las imágenes intermedias y al llegar al dibujo final, se hace retroceder el proceso hasta el dibujo inicial, repitiéndose esto hasta recibir una orden por parte de el usuario. Esta última opción ofrece la posibilidad de representar movimientos armónicos.

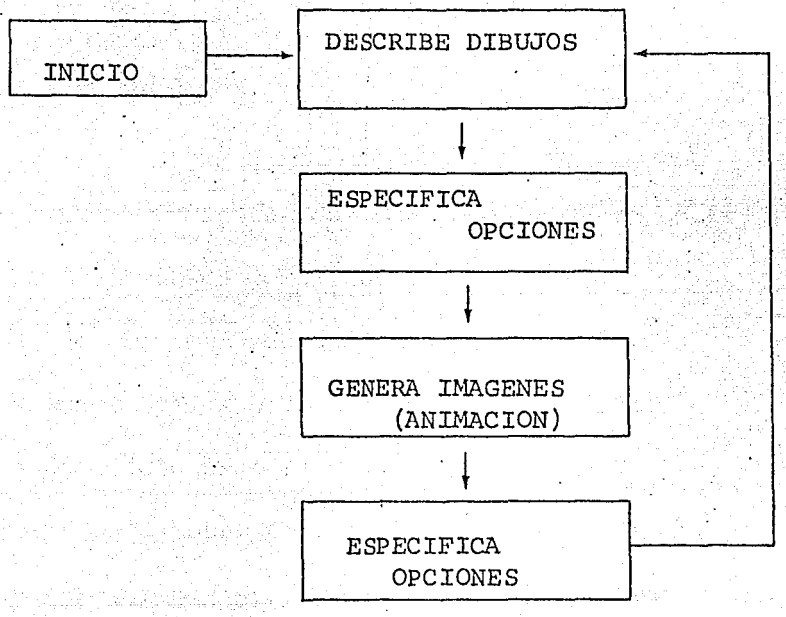

Figura 11.- Diagrama de bloque del funcionamiento del sistema.

Todos los mensajes dados por el sistema y destinados al usuario, los cuales sirven para guiar al último en la interacción, se dan a través de la pantalla con caracteres alfanuméricos; y casi todas las respuestas de el usuario se dan por medio de la tableta y pluma electrónica, a excepción de algunos comandos que se suministran por teclado.

2.- La estructura de datos.

Existen ocho listas principales que dan la estructura básica al sistema y que se han denominado como sigue:

i) de códigos descriptores para dibujo dinámico,"LCD".

ii) de puntos descriptores para dibujo inicial, "LPI".

iii) de puntos descriptores para dibujo final, "LPF".

iv) de códigos descriptores para dibujo estático, "LCE".

v) de puntos descriptores para dibujo estático, "LPE".

vi) dos listas de instrucciones al generador de vectores para dibujo dinámico, "LD1" y "LD2".

vii) una lista de instrucciones al generador de vectores para dibujo estático, "LDE".

(En donde las letras mayúsculas encerradas por comillas se usarán en adelante como abreviación).

Ninguna de las listas anteriores tiene una longitud predeter-

minada, aunque para todas existe una cota de crecimiento.

Las listas LPI y LPF contienen parejas de números enteros, que denotan las coordenadas cartesianas de los puntos paramétricos para los dibujos inicial y final respectivamente; se encuentran alternadas las abscisas y ordenadas, y la primer pareja en cada una de ellas denota el punto en la pantalla a donde debe dirigirse el haz de electrones al empezar a dibujar.

Estas listas se relacionan en forma biunívoca, de acuerdo al orden en que aparezcan las coordenadas de los puntos. La anterior relación es para indicar hacia que lugar ha de moverse cada uno de los puntos que describan al dibujo inicial.

Como ya se ha mencionado, estos puntos son para unirse con rectas, o para generar arcos de curva mediante el algoritmo descrito en el capítulo II.

La función de la lista LCD es indicar como unir los puntos en LPI y LPF. Es decir, contiene códigos que representan rectas o curvas, de acuerdo al orden con que se haya hecho uso de estos elementos básicos para aproximar un dibujo.

El orden en que aparezcan será acorde al orden de los puntos en LPI y LPF, y a los movimientos que ha de seguir el haz al refrescar la pantalla.

Hagamos uso de un ejemplo simple para precisar las ideas: - si el usuario desease representar internamente el dibujo de la figura 12a, el cual ha de deformarse al de la figura 12b, la representación interna habría quedado como en la figura 14, de acuerdo a las figuras 13 a y 13 b.

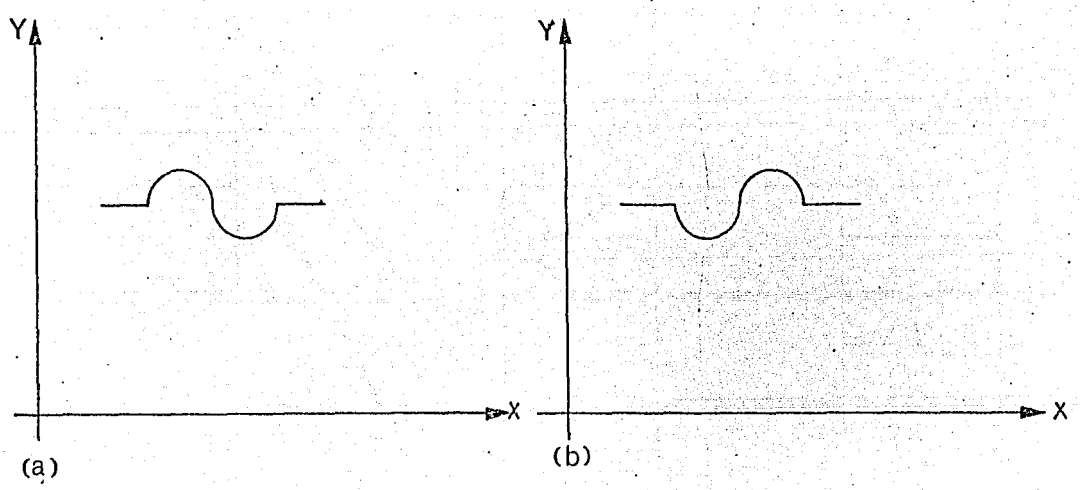

Figura 12 .- Dibujo inicial (a) y final (b). Los ejes corresponden a las orillas inferior e izquierda de la pantalla.

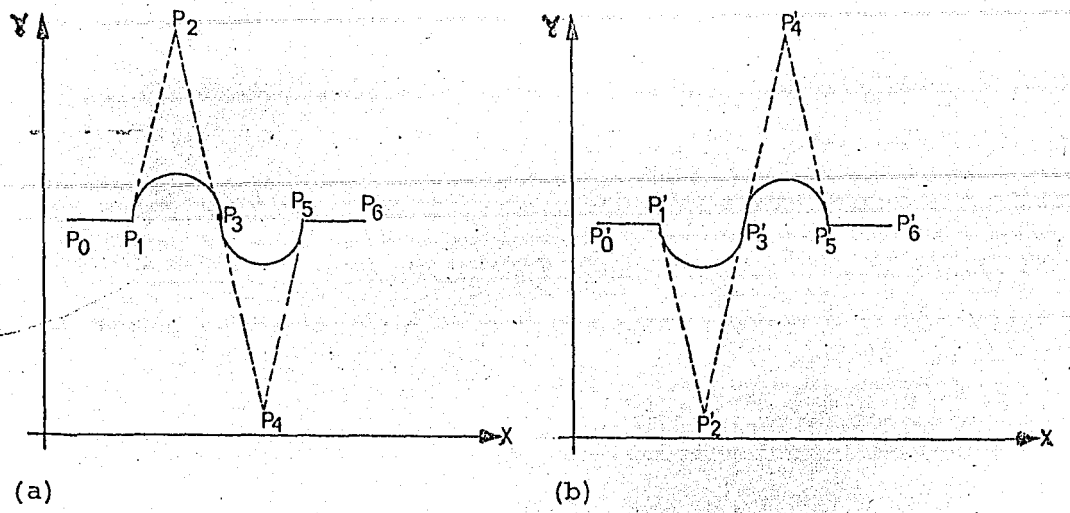

Figura 13.- Los puntos marcados corresponden a los puntos paramétricos.

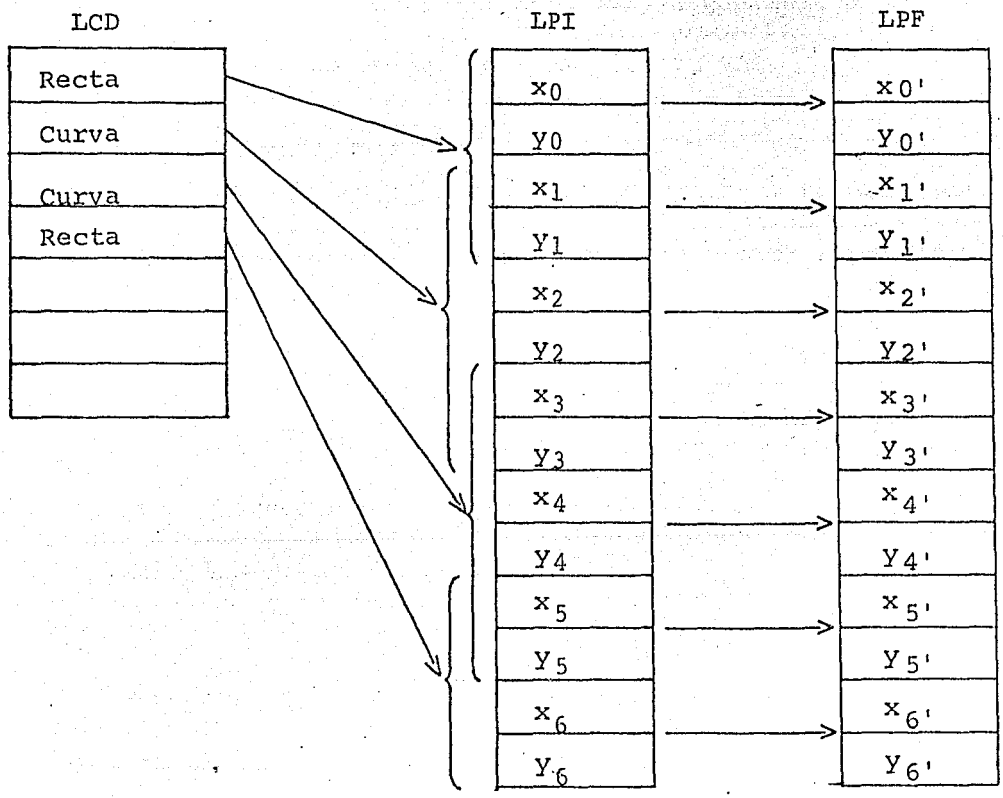

Figura 14.- Representación interna de la figura 7.

Cada casilla representa una palabra en procesador central.

Cada uno de los puntos en LPI debe dirigirse al correspondiente en LPF.

Los códigos en LCD no solo informan si se trata de segmento rectilíneo o curvo, sino también con que tipo de línea ha de generarse el elemento, pudiendo ser continua, punteada o invisible. (ver fig. 10)

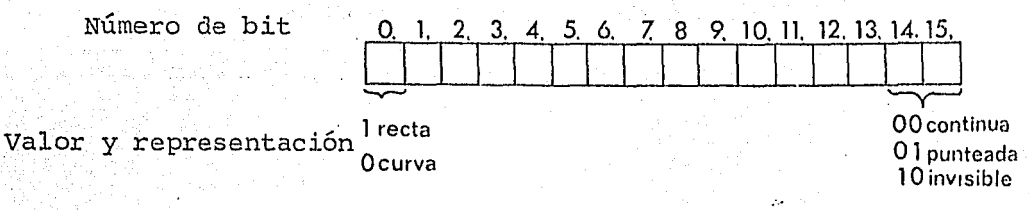

Figura 15.- Palabra de memoria y organización de los bits para formar un elemento en la lista de códigos LCD.

Las listas LCE y LPE están estructuradas y relacionadas de manera idéntica que LCD y LPI, y son necesarias para la generación del objeto estático o fijo.

LD1, LD2 y LDE son listas en donde se almacenarán las instrucciones para el generador de vectores que se encargará de refrescar en la pantalla las imágenes deseadas.

Son necesarias dos listas de instrucciones al generador de vectores para el dibujo dinámico, pues mientras una se use para refrescar la pantalla con una imagen, se estarán generando en la otra las instrucciones para la imagen siguiente; la lista para el objeto fijo se estará refrescando constantemente.

Para lograr lo anterior basta intercambiar las listas LD1 y LD2 en los procesos de generación y refrescamiento. Al final de cada una de estas listas se incerta una instrucción de transferencia que indique al generador de vectores continue desde la cabeza de LDE, al final de la cual encontrará una instrucción que le ordene desactivarse.

3.- Los procedimientos para describir el dibujo inicial y el dibujo estático.

Se dispone de una subrutina común para describir el dibujo inicial y el estático, la cual queda resumida en la figura 16.

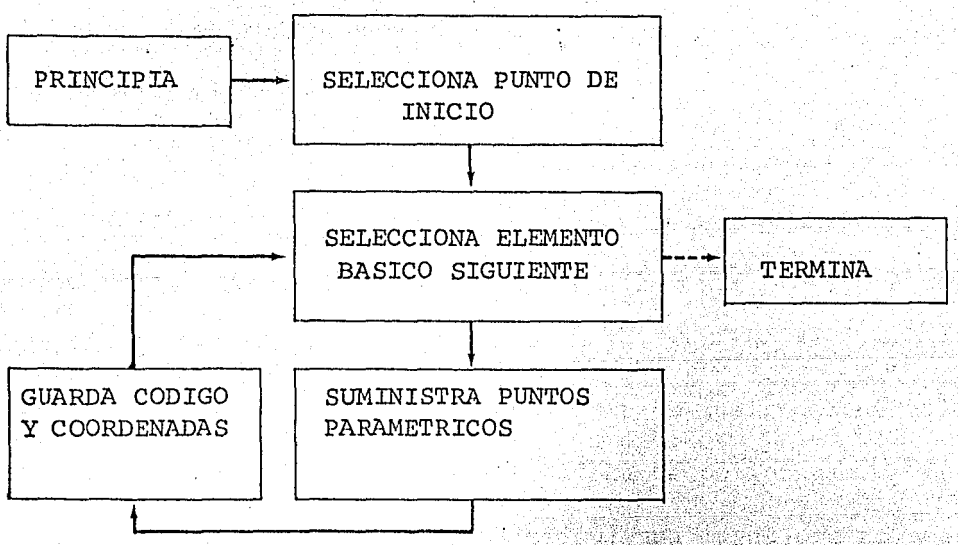

Figura 16.- Resumen del procedimiento para describir dibujo inicial o dibujo estático

Esta subrutina toma por parámetros las direcciones de inicio de las listas de códigos y puntos (LCD y LPI, o LCE y LPE) las cuales construye, además lleva un contador para el número de códigos descriptores.

Por hacer clara la descripción del procedimiento nos referiremos a las listas LCD y LPI, aunque de manera idéntica lo haríamos con LCE y LPE.

Una vez tomados los parámetros de entrada se activa la pluma electrónica y aparece en la pantalla un mensaje que solicita al usuario un punto de inicio para introducir la descripción.

El cursor aparecerá en la posición correspondiente a la que ocupe la pluma electrónica en este momento sobre la tableta, posición que se actualizará constantemente.

El usuario debe colocar la pluma electrónica en el lugar seleccionado y presionarla contra la tableta, (lo cual provocará una interrupción que en adelante llamaremos "golpe de pluma"). En este momento se genera a la cabeza de una lista de instrucciones al generador de vectores (que llamaremos "lista de poligonal", y que se refrescará hasta terminar la descripción) una marca que aparecerá en el lugar correspondiente al punto de inicio.

También se guardan a la cabeza de LPI las coordenadas del punto seleccionado.

El usuario tiene en este momento diversas alternativas, las cuales aparecen en la pantalla en forma de menú, (ver fig. 17), según la opción seleccionada el usuario debe de señalar con la pluma electrónica el cuadrado correspondiente (después de los puntos sus

pensivos).

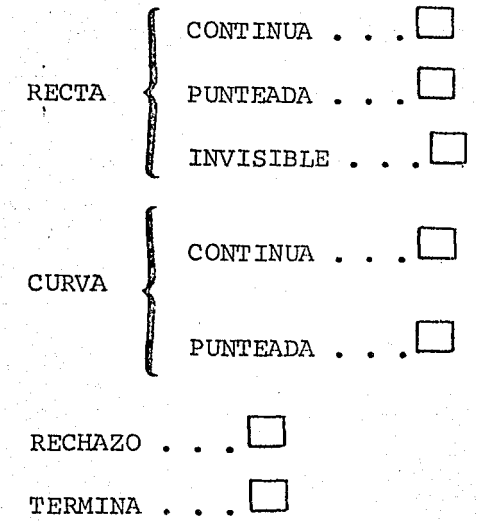

Figura 17 .- Menú de opciones en la rutina describe dibujo inicial (ó estático).

RECTA CONTINUA . . . ☐

Desaparece el menú de la pantalla siendo refrescados el cursor que corresponde a la posición de la pluma* y la poligonal; esta vez (y en adelante) hay una línea recta que va desde el punto - descriptor inmediato anterior hasta el cursor.

Al ser movida la pluma por el usuario esta recta variará en longitud e inclinación, lo que producirá la impresión de ser una - liga elástica, fija por un extremo al último punto descriptor y por el otro al cursor.

La función de esta "recta elástica" es dotar al usuario con la capacidad de hacer colineales tres o más puntos descriptores.

Al producirse un golpe de pluma, se determinará la recta a introducir; se guardan las coordenadas del punto seleccionado en el primer lugar libre en LPI, se genera un código para recta continua en el primer sitio libre en LCD, se suma una unidad al contador de

* En adelante usaremos las expresiones "pluma" y "cursor" dando a entender "pluma electrónica" y "cursor en la posición correspondiente a la de la pluma electrónica" respectivamente.

códigos descriptores, se aumenta esta recta a la lista de poligonal y regresa a desplegar el menú.

RECTA PUNTEADA . . . ☐ y RECTA INVISIBLE . . . ☐

Estas opciones se tratan en forma análoga a RECTA CONTINUA, solo que los códigos generados difieren en los últimos dos bits - (ver fig. 15).

CURVA CONTINUA . . . ☐

Similar a las anteriores opciones, desaparece el menú y en su lugar la recta de apariencia elástica, que va desde el último - punto introducido hasta el lugar actual del cursor.

Esta vez el usuario debe de hacer dos golpes de pluma para determinar el arco de curva elegido, (este queda determinado por - el último punto introducido antes de entrar a esta parte, y los dos nuevos puntos seleccionados), las coordenadas de los puntos señalados se guardan en el orden de la selección a partir del primer lugar libre en LPI.

Se genera la curva (es decir las instrucciones para el generador de vectores mediante el algoritmo del capitulo II,) en la lis

ta de poligonal, se añade un código para curva continua en LCD, se suma uno al contador de códigos, y vuelve a desplegar el menú.

CURVA PUNTEADA . . . ☐

Se procesa de manera idéntica a la anterior, solo que genera el código correspondiente.

RECHAZA . . . ☐

Hay ocasiones en que se cometen errores en la lectura de coordenadas por la pluma electrónica, y también es posible que el usuario cometa equivocaciones al ir seleccionando los elementos para la descripción; es por esto que es necesaria la opción RECHAZA.

La forma que tiene el usuario de saber si tiene la necesidad de recurrir a esta opción, es refrescando la lista de poligonal a lo largo del proceso de descripción.

Cada vez que se introduce un nuevo elemento básico, (segmentos de recta o curva), y antes de volver amostrar el menú, se añade a la lista de poligonal a partir del primer lugar libre, las instrucciones al generador de vectores para refrescar la recta o curva introducida.

La dirección de este primer lugar libre en la lista de poligonal se guarda en una torre de apilamiento.

Al haber un rechazo, se toma la dirección de hasta arriba de la torre de apilamiento, (mencionada en el párrafo precedente) para eliminar el elemento último anterior de la lista de poligonal, se da de baja el último código en LCD y sus puntos asociados en LPI, por último se decrementa en una unidad el contador de códigos. Se continúa refrescando el menú.

TERMINA . . . □

Simplemente provoca que se salga de la rutina dando por terminada la descripción del dibujo.

4.- El procedimiento para describir el dibujo final.

Este es totalmente diferente al anterior, pues ahora el usuario no tiene libertad para escoger los elementos básicos, sino que el sistema se los pedirá de acuerdo a la descripción del dibujo inicial; esto es con objeto de simplificar la generación de imágenes animadas.

La función interna del procedimiento es llenar la lista LPF, interactuando con el usuario de acuerdo a los códigos almacenados

en LCD. Es decir, de acuerdo a cada código de elemento básico (principiando por el punto de inicio) y su orden en LCD, se despliega en la pantalla un mensaje que indica al usuario el tipo de elemento a introducir.

Puesto que en esta etapa también es necesaria la opción de rechazo, junto con el mensaje que indica al usuario que elemento ha de introducir, aparecerá: RECHAZO . . . ☐ El rechazo se refiere solamente a las coordenadas del elemento último anterior, si se desea ha de hacerse un golpe de pluma en el cuadrado correspondiente, lo que provocará se retroceda un lugar en la lectura de LCD.

5.- El procedimiento genera imágenes.

Lo primero que hace al entrar a esta subrutina, es generar mediante LCE y LPE en LDE las instrucciones para refrescar el dibujo estático. Entonces se inicializa un contador K a cero, el cual llevará un conteo de iteraciones en cada una de las cuales se generará una imagen de transición entre el dibujo inicial y el final. El número de imágenes transitorias ha sido especificado anteriormente por el usuario (Ver fig. 18).

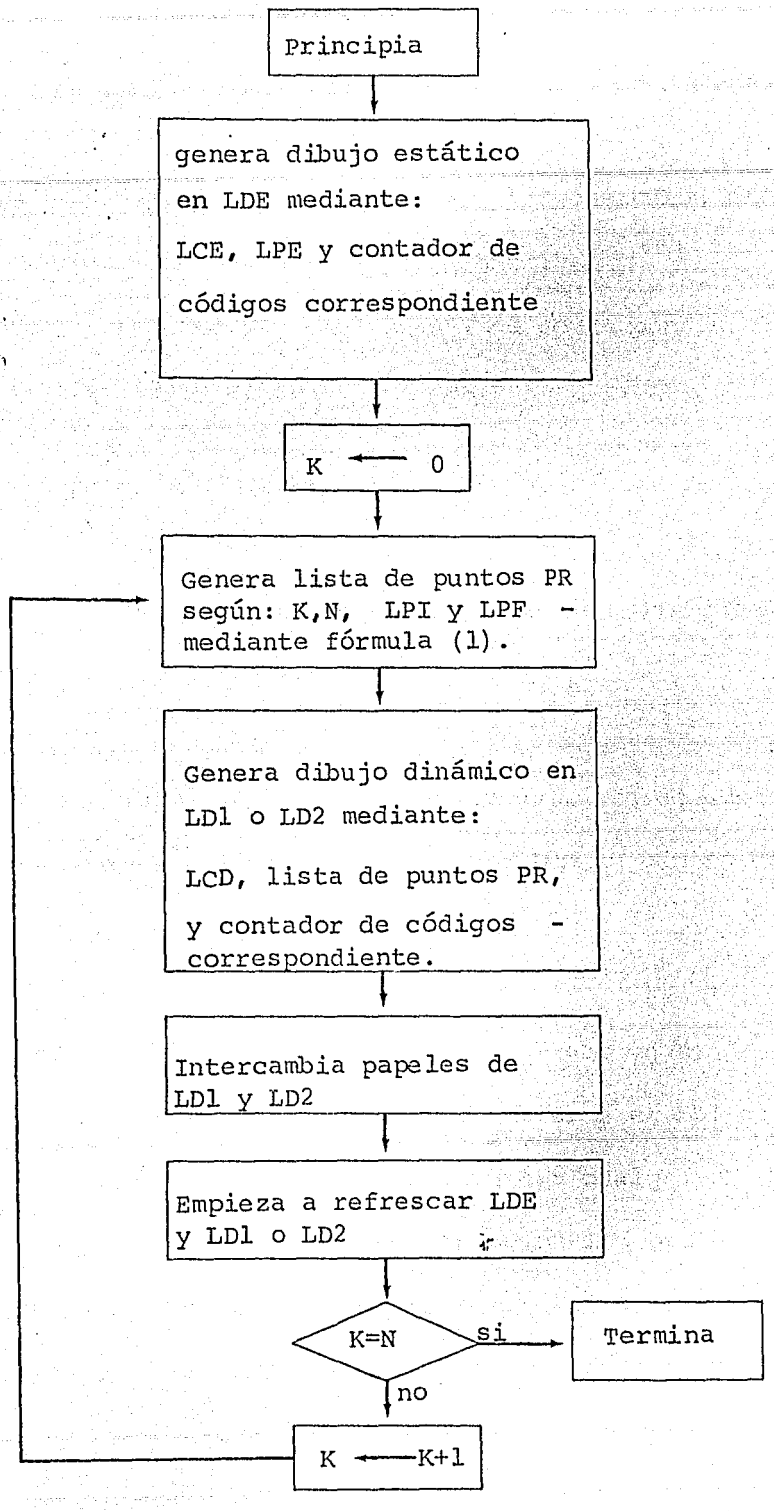

Figura 18.- Diagrama de flujo del procedimiento genera imágenes.

Cada iteración que corresponde a la generación de una imagen de transición consiste en lo siguiente:

Tomar para cada punto PI en LPI su correspondiente PF en LPF (véase fig. 14) y calcular PR* mediante la formula (1).

$$PR = PI + (K/N)\ (PF-PI) \qquad (1)$$

En donde N es el número de imágenes de transición a generar y K el contador de iteraciones.

De esta forma se obtendrá una nueva lista de puntos PR por iteración. Con esta lista y la ayuda de LCD se genera cada imagen de transición.

Como se mencionó anteriormente el lugar en donde se generarán las instrucciones al generador de vectores es LD1 alternada con LD2; en la primera iteración se empieza con cualquiera de ellas, y en adelante se intercambiarán por iteración los procesos de refrescamiento y recepción de instrucciones para el generador de vectores. LDE se refrescará constantemente con alguna de las listas anteriores.

* PR denota un punto del conjunto de puntos para describir un dibujo intermedio

6.- Funcionamiento global del sistema.

En la figura 19 se tiene el diagrama de flujo del sistema con los principales procedimientos o subrutinas. En la columna derecha de la misma figura cada cuadrado representa la pantalla y lo que en esta se vería si se hiciera la animación bosquejada por la figura 12, (sin haber dibujo estático). Cada dibujo de pantalla está colocado a la altura correspondiente a la subrutina en el diagrama de flujo que lo generaría.

El tercer dibujo de arriba hacia abajo en la columna derecha de la figura 19, es un menú que aparece una vez descritos los dibujos inicial y final. Si se hace un golpe de pluma a DIBUJO ESTATICO . . . ☐ el sistema llama a la subrutina para describir el dibujo estático, y al finalizar con esta descripción aparecerá el menú:

NUMERO DE PASOS
- RAPIDO . . . ☐
- MEDIO . . . ☐
- LENTO . . . ☐

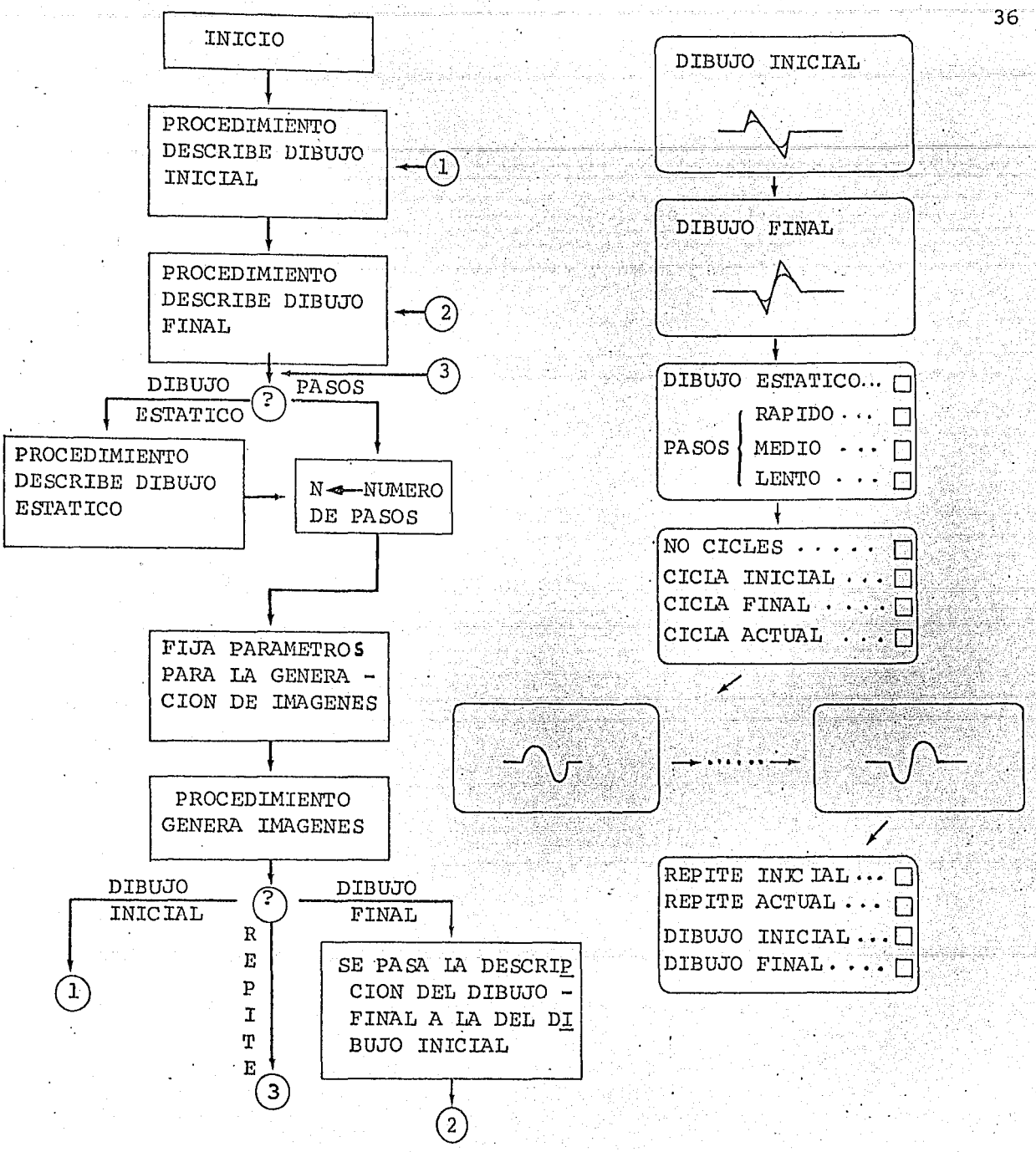

Figura 19.- La columna izquierda es el diagrama de flujo del sistema. La columna derecha es un esbozo de las imágenes y mensajes al usuario mas importantes dentro de un ciclo completo; descripción de dibujos y animación de los mismos.

Estas opciones se refieren al número de imágenes transitorias a generar entre el dibujo incial y el dibujo final: RAPIDO para - 20 imágenes, MEDIO para 40 imágenes, y LENTO para 80 imágenes.

Puesto que este último menú también aparece antes de describir el dibujo estático, si el usuario no requiere de esta imagen - fija, puede especificar directamente el número de pasos.

A continuación aparece el menú:

NO CICLES . . . ☐

CICLA INICIAL . . . ☐

CICLA FINAL . . . ☐

CICLA ACTUAL . . . ☐

Estas son opciones que tiene el usuario sobre la generación de imágenes, e inmediatamente después de hacer una selección, se - procede con la generación de los dibujos animados.

NO CICLES . . . ☐

Se genera desde el dibujo inicial hasta el final, y aparece el menú de opciones siguiente.

CICLA FINAL . . . ☐

Se provoca se cicle la generación de imágenes, es decir, - partiendo del dibujo inicial al llegar al dibujo final se repite el proceso de generación de imágenes de transición pero en orden - inverso, (o sea del final al inicial). Se vuelve a repetir todo el proceso; así hasta que el usuario oprima cualquier tecla. Esto - provocará una interrupción de unos cuantos microsegundos en la generación de imagenes, la cual se aprovecha para modificar el proceso de generación; de tal forma que la próxima vez que se genere una imagen correspondiente al dibujo final, se detenga el proceso y aparezca el siguiente menú de opciones.

CICLA INICIAL . . . ☐

Análoga a la opción anterior, solo que después que el usuario oprima una tecla, la imagen en que se detendrá el ciclo será la correspondiente al dibujo inicial. Al aparecer el menú siguiente, habrá habido un cambio de identificación entre los dibujos inicial y final, es decir, lo que el usuario originalmente había descrito como dibujo inicial, ahora el sistema lo identifica como dibujo final y el inicial al que fuera final.

CICLA ACTUAL. . . ☐

La generación de imágenes se cicla en forma análoga a las

anteriores opciones; esta vez al haber la interrupción provocada por el usuario al oprimir una tecla, la generación de imágenes se detiene en el momento mismo de la interrupción.

Como la descripción de esta última imagen es recuperable - esta pasa a ser ahora el dibujo final, y la correspondiente al dibujo inicial, será la que antes del proceso de generación de imágenes fuera el dibujo inicial si la interrupción del proceso se provocó cuando el ciclo "viajaba" del dibujo inicial al final; de otra forma será la que correspondía al dibujo final.

Acabada la generación de imagenes aparece el menu:

REPITE INICIAL . . . ☐

REPITE ACTUAL . . . ☐

DIBUJO INICIAL . . . ☐

DIBUJO FINAL . . . ☐

REPITE INICIAL . . . ☐

Si ha habido algún cambio de identificación en los dibujos inicial y final después del proceso de generación de imágenes, se restauran las identificaciones originales.

Se hace una transferencia en el programa al punto en donde

el usuario ha de describir el dibujo estático (si lo desea), o especificar el número de pasos. (ver fig. 19)

En este momento no se ha perdido la descripción de el objeto estático, sin embargo el usuario tiene la posibilidad de modificarla.

REPITE ACTUAL . . . ☐

Se conservan las identificaciones de dibujos actuales, y se transfiere el programa al punto en donde el usuario ha de describir el dibujo estático o especificar el número de pasos.

DIBUJO INICIAL . . . ☐

Se transfiere a la descripción del dibujo inicial como si el sistema se inicializara, solo que se conserva la descripción del dibujo estático actual.

DIBUJO FINAL . . . ☐

Lo que en este momento es descripción del dibujo final, pasa a ser la del dibujo inicial, y se transfiere al programa a solicitar la descripción de un nuevo dibujo final.

APENDICE A.-El algoritmo de Chaikin.

La construcción hecha por el algoritmo del capítulo II puede describirse como sigue:

Dada la terna (A,B,C) de puntos en el plano con A y C en la curva a aproximar, el siguiente punto P en la curva se encuentra mediante la fórmula:

$$P=(A+2B+C)/4 \qquad (1)$$

Este es el punto medio de el segmento determinado por (A+B)/2 y (B+C)/2.

Ahora se redefinen A, B, C en cualquiera de los casos:

Caso a) $A \leftarrow A$
$B \leftarrow (A+B)/2$
$C \leftarrow P$

Caso b) $A \leftarrow P$
$B \leftarrow (B+C)/2$
$C \leftarrow C$

Ambos casos dan lugar a un nuevo punto P en la curva, definido por la ecuación (1).

El procedimiento se continúa (usando ambos casos) hasta que el conjunto de puntos en la curva sea suficientemente denso para la aplicación gráfica en cuestión.

Otra posible descripción de la construcción es: tomar el triángulo ABC, con A y C en la curva (ver fig. I); el siguiente punto P = (A+2B+C)/4 en la curva, es el punto medio del segmento con vértices (A+B)/2 y (B+C)/2.

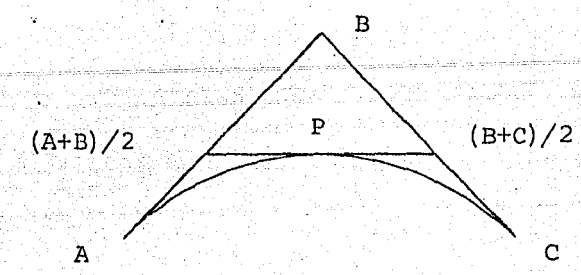

Figura I.

Estos puntos determinan dos nuevos triángulos cuyos vértices son A, (A+B)/2, P y P,(B+C)/2, C. Cada uno de los dos nuevos - triángulos da origen a un nuevo punto en la curva. Esta construcción ha de continuarse hasta obtener tantos puntos en la curva - como sean necesarios. (Ver Fig. II).

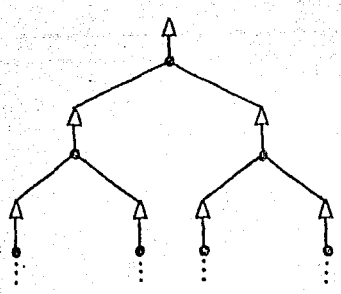

Figura II.- Jerarquía en la definición de triángulos y puntos por el algoritmo de G. Chaikin.

Proposición 1.- Dada la parábola $\alpha(t)=(t,kt^2)$ $t\in\mathbb{R}$, y dos puntos distintos A y C en la parábola; si B es el punto de intersección de las rectas tangentes a la parábola en A y C. Entonces el punto $P=(A+2B+C)/4$ está en la parábola, y la recta tangente a la parábola en P contiene a los puntos $(A+B)/2$ y $(B+C)/2$.

Demostración.- Si $A=(a,ka^2)$ y $C=(c,kc^2)$, entonces las ecuaciones de las rectas tangentes a α en A y C son:

$$2kax-y=ka^2$$

$$2kcx-y=kc^2$$

De donde al resolver el anterior sistema de ecuaciones lineales se obtiene que $B=((a+c)/2,kac)$. Por lo tanto $P=(A+2B+C)/4=((a+c)/2,k(a+c)^2/4)$ está en la parábola α.

En P la recta tangente tiene representación paramétrica $\beta(t)=P+t(1,k(a+c))$ $t\in\mathbb{R}$. Y se tiene que $\beta((a-c)/4)=(A+B)/2$

$\beta((c-a)/4)=(B+C)/2$ ∎

Proposición 2.- Los puntos generados por el algoritmo de Chaikin están en un arco de parábola.

La demostración se hará por inducción sobre los triángulos y puntos que se van definiendo.

Demostración.- El arco de parábola.

$\alpha(t)=t^2(A-2B+C)+2t(B-A)+A \quad t\in[0,1]; \quad A,B,C\in\mathbb{R}^2$ no colineales es tal que:

$\alpha(0)=A \qquad \alpha'(0)=2(B-A)$

$\alpha(1)=C \qquad \alpha'(1)=2(C-B)$

Por tanto A B y C satisfacen las hipótesis enunciadas en la proposición 1 y necesariamente

$P=(A+2B+C)/4$ está en la parábola.

Suponga que $A_nB_nC_n$ es uno de los triángulos que se definen en algún momento en el algoritmo de Chaikin y es tal que

$\alpha(t_0)=A_n \qquad \alpha'(t_0)=k_0(B_n-A_n)$

$\alpha(t_1)=C_n \qquad \alpha'(t_1)=k_1(C_n-B_n)$

con $t_0, t_1\in[0,1]$, $t_0<t_1$ y $k_0, k_1\in\mathbb{R}$

De la proposición 1 se sigue que existe $s\in[0,1]$ con $t_0<s<t_1$ tal que $\alpha(s)=P=(A_n+2B_n+C_n)/2$ i.e. P está en la parábola. Además, de la misma proposición 1, se sigue que los triángulos inducidos de vértices $A_n, (A_n+B_n)/2, P$ y $P, (B_n+C_n)/2, C_n$ son tales que:

$$\alpha'(t_0) = 2k_0\,((A_n+B_n)/2 - A_n)$$

$$\alpha'(s) = k_2\,(P-(A_n+B_n)/2) = k_2\,((B_n+C_n)/2 - P) \qquad k_2 \in \mathbb{R}$$

$$\alpha'(t_1) = 2k_1\,(C_n-(B_n+C_n)/2)$$

■

REFERENCIAS.

1. G. Chaikin, An algorithm for high-speed curve generation, Computer Graphics and Image Processing Vol. 3, 1974, -- 346-349.

2. R.F. Riesenfeld, On Chaikin's Algorithm, Computer Graphics and Image Processing Vol. 4, 1975, 304-310.

3. W.M. Newman y R.F. Sproull, Principles of Interactive Computer Graphics, McGraw-Hill, N.Y., 1973.

4. C. Warkle, Artspeak: A Graphics Language for Artists, Proceedings ACM Symposium on Graphic Languages, Abril 1976, Miami Florida.

5. R. Williams, A Survey of data structures for computer graphics systems, Computing Surveys Vol. 3, 1971, 1-21.

6. PDS-4 System Reference Manual, Imlac Corporation, 1974.

7. M. D. Price, Interactive Graphics for Computer-aided -- Design, Addison-Wesley, Reading Mass., 1971.