

Animación por computadora



Una imagen cuyas características faciales han sido transformadas mediante las técnicas de morfismo.
(Cortesía de Vertigo Technology, Inc.)

- | | | | |
|------|---|-------|---------------------------------------|
| 13.1 | Métodos de barrido para las animaciones por computadora | 13.6 | Sistemas de fotogramas clave |
| 13.2 | Diseño de secuencias de animación | 13.7 | Especificaciones de movimientos |
| 13.3 | Técnicas tradicionales de animación | 13.8 | Animación de figuras articuladas |
| 13.4 | Funciones generales de animación por computadora | 13.9 | Movimientos periódicos |
| 13.5 | Lenguajes de animación por computadora | 13.10 | Procedimientos de animación en OpenGL |
| | | 13.11 | Resumen |

Hoy en día, se utilizan los métodos infográficos de forma común para generar animaciones para muy diversas aplicaciones, incluyendo el entretenimiento (películas y dibujos animados), la industria publicitaria, los estudios científicos y de ingeniería, la formación y la educación. Aunque todos tendemos a pensar en la animación como si implicara el movimiento de objetos, el término **animación por computadora** se refiere, en general, a cualquier secuencia temporal donde se aprecien cambios visuales en una imagen. Además de cambiar las posiciones de los objetos mediante traslaciones o rotaciones, una animación generada por computadora puede mostrar variaciones temporales que afecten al tamaño de los objetos, a su color, a su transparencia o a las texturas superficiales. Las animaciones utilizadas en la industria publicitaria recurren frecuentemente a las transiciones entre una forma de objeto y otra: por ejemplo, transformar una lata de aceite para motores en un motor de automóvil. También podemos generar animaciones por computadora variando parámetros de la cámara, como la posición, la orientación o la distancia focal. Y las variaciones en los efectos de iluminación u otros parámetros y procedimientos asociados con la iluminación y la representación de escenas pueden usarse también para producir animaciones por computadora.

Una consideración importante en las animaciones generadas por computadora es la cuestión del realismo. Muchas aplicaciones requieren imágenes suficientemente realistas. Por ejemplo, una representación precisa de la forma de una tormenta o de otro fenómeno natural escrito con un modelo dinámico tiene una gran importancia para evaluar la fiabilidad del modelo. De forma similar, los simuladores para el entrenamiento de los pilotos de aeronaves y de los operadores de equipos pesados debe producir representaciones razonablemente precisas del entorno. Las aplicaciones de entretenimiento y publicitarias, por el contrario, suelen estar más interesadas en los efectos visuales. De este modo, puede que las escenas se muestren con formas exageradas y movimientos y transformaciones no realistas. Sin embargo, hay muchas aplicaciones de entretenimiento y publicitarias que sí que requieren una representación precisa en las escenas generadas por computadora. En algunos estudios científicos y de ingeniería, el realismo no constituye un objetivo; por ejemplo, determinadas magnitudes físicas suelen mostrarse con pseudo-colores o con formas abstractas que cambian con el tiempo con el fin de ayudar al investigador a comprender la naturaleza del proceso físico.

Dos métodos básicos para la construcción de una secuencia animada son la **animación en tiempo real** y la **animación imagen a imagen**. En una animación por computadora en tiempo real, cada etapa de la secuencia se visualiza a medida que se la genera. Por ello, la animación debe generarse a una frecuencia que sea compatible con las restricciones de la frecuencia de refresco. Para una animación imagen a imagen, se genera de forma separada cada imagen de la secuencia y se la almacena. Posteriormente, las imágenes pueden grabarse sobre una película o mostrarse de forma consecutiva en un monitor de vídeo, en el modo de «reproducción en tiempo real». Las secuencias animadas simples se suelen producir en tiempo real, mientras que las animaciones más complejas se construyen más lentamente, imagen a imagen. Pero algunas aplicaciones requieren animación en tiempo real, independientemente de la complejidad de la animación. Una animación para un

simulador de vuelo, se tiene que generar en tiempo real, porque las imágenes de vídeo deben construirse respondiendo de manera inmediata a los cambios de las configuraciones de control. En tales casos, se suelen desarrollar sistemas hardware y software especializados con el fin de poder generar rápidamente las complejas secuencias de animación.

13.1 MÉTODOS DE BARRIDO PARA LAS ANIMACIONES POR COMPUTADORA

La mayoría de las veces, podemos crear secuencias simples de animación en nuestros programas utilizando métodos de tiempo real, pero en general, podemos producir una secuencia animada en un sistema de visualización por barrido generando una imagen cada vez y guardando dicha imagen completa en un archivo para su visualización posterior. La animación puede verse entonces recorriendo la secuencia completa de imágenes, o bien pueden transferirse esas imágenes a una película. Sin embargo, si queremos generar una imagen en tiempo real, debemos producir las imágenes de la secuencia con la suficiente rapidez como para que se perciba un movimiento continuo. Para una escena compleja, la generación de cada imagen de la animación puede ocupar la mayor parte del ciclo de refresco. En tal caso, los objetos que se generen primero se visualizarán durante la mayor parte de ese ciclo de refresco, pero los objetos generados hacia el final del mismo desaparecerán muy poco después de mostrarlos. Asimismo, en las animaciones muy complejas, el tiempo de generación de la imagen podría ser superior al tiempo necesario para refrescar la pantalla, lo que puede hacer que se perciban movimientos erráticos y que se muestren imágenes fracturadas. Puesto que las imágenes de la pantalla se generan a partir de los valores de píxel sucesivamente modificados que hay almacenados en el búfer de refresco, podemos aprovechar algunas de las características del proceso de refresco de pantalla en los sistemas de barrido con el fin de generar rápidamente las secuencias de animación.

Doble búfer

Un método para producir una animación en tiempo real con un sistema de barrido consiste en emplear dos búferes de refresco. Inicialmente, creamos una imagen para la animación en uno de los búferes. Después, mientras se refresca la pantalla a partir del contenido de dicho búfer, construimos la imagen siguiente de la secuencia en el otro búfer. Cuando dicha imagen se complete, cambiamos los roles de los dos búferes para que las rutinas de refresco utilicen el segundo búfer mientras se crea la siguiente imagen de la secuencia en el siguiente búfer. Este proceso alternativo de conmutación de búferes continua mientras dure la secuencia. Las bibliotecas gráficas que permiten tales operaciones suelen disponer de una función para activar la rutina de doble búfer y otra función para intercambiar los papeles de los dos búferes.

Cuando se realiza una llamada para conmutar los dos búferes de refresco, el intercambio puede realizarse en diversos instantes. La implementación más sencilla consiste en conmutar los búferes al final del ciclo de refresco actual, durante el retorno vertical del haz de electrones. Si un programa puede completar la construcción de una imagen dentro del tiempo que dura un ciclo de refresco, como por ejemplo $\frac{1}{60}$ de segundo, la secuencia animada se mostrará de forma sincronizada con la tasa de refresco de pantalla.

Pero si el tiempo necesario para construir una imagen es mayor que el tiempo de refresco, la imagen actual se mostrará durante dos o más ciclos de refresco mientras se genera la siguiente imagen de la secuencia de animación. Por ejemplo, si la tasa de refresco de pantalla es de 60 imágenes por segundo y se tarda $\frac{1}{30}$ de segundo en construir cada nueva imagen de la secuencia, las imágenes se mostrarán en pantalla dos veces y la velocidad de animación será únicamente de 30 imágenes por segundo. De forma similar, si el tiempo de construcción de una nueva imagen es de $\frac{1}{25}$ de segundo, la velocidad de imagen de la animación se reduce a 20 imágenes por segundo, ya que cada una de las imágenes tendrá que ser mostrada tres veces.

Con la técnica de doble búfer pueden aparecer velocidades de animación irregulares cuando el tiempo de generación de cada imagen está muy próximo a un múltiplo entero del tiempo de refresco de pantalla. Como ejemplo de esto, si la velocidad de refresco de pantalla es de 60 imágenes por segundo, podría producirse una velocidad de animación errática si el tiempo de construcción de la imagen estuviera muy próximo a $\frac{1}{60}$ de

segundo, $\frac{2}{60}$ de segundo o $\frac{3}{60}$ de segundo, etc. Debido a las pequeñas variaciones en el tiempo de ejecución de las rutinas que generan las primitivas y sus atributos, algunas imágenes podrían requerir algo más de tiempo para generarse y otras podrían requerir un tiempo más corto. Eso podría hacer que la velocidad de animación cambiara de forma abrupta y errática. Una forma de compensar este efecto consiste en añadir un pequeño retardo temporal al programa, mientras que otra posibilidad es alterar los movimientos o la descripción de la escena con el fin de acortar el tiempo de construcción de las imágenes.

Generación de animaciones mediante operaciones de barrido

También podemos generar animaciones en tiempo real en los sistemas de barrido, para algunas aplicaciones limitadas, utilizando la transferencia en bloque de matrices rectangulares de píxeles. Esta técnica de animación se utiliza a menudo en los programas de juegos. Como hemos visto en la Sección 5.6, un método simple para mover un objeto de una ubicación a otra en el plano xy consiste en transferir el grupo de píxeles que definen la forma del objeto hasta una nueva ubicación. Las rotaciones bidimensionales en múltiplos de 90° también son simples de realizar, e incluso podemos rotar bloques rectangulares de píxeles otros ángulos distintos, siempre que utilicemos procedimientos de *antialiasing*. Para una rotación que no sea un múltiplo de 90° , necesitamos estimar el porcentaje de recubrimiento de área correspondiente a los píxeles que se solapan con el bloque rotado. Pueden ejecutarse secuencias de operaciones de barrido para obtener una animación en tiempo real de objetos bidimensionales o tridimensionales, siempre que restrinjamos la animación a movimientos dentro del plano de proyección. Entonces, no será necesario invocar algoritmos de proyección ni de detección de superficies visibles.

También podemos animar los objetos según trayectorias de movimiento bidimensionales utilizando **tablas de transformación de colores**. Con este método, predefinimos el objeto en posiciones sucesivas a lo largo de la trayectoria de movimiento y asignamos a los sucesivos bloques de píxeles una serie de entradas en la tabla de color. Los píxeles correspondientes a la primera posición del objeto se configuran con un color de primer plano y los píxeles de las demás posiciones del objeto se configuran con el color de fondo. La animación se consigue entonces cambiando los valores de la tabla de colores, de modo que el color del objeto en las posiciones sucesivas a lo largo del trayecto de animación se vaya transformando en el color de primer plano, a medida que la posición precedente se configura con el color de fondo (Figura 13.1).

13.2 DISEÑO DE SECUENCIAS DE ANIMACIÓN

La construcción de una secuencia de animación puede ser una tarea complicada, particularmente cuando requiere un guión y múltiples objetos, cada uno de los cuales puede moverse de diferente forma. Un enfoque básico consiste en diseñar tales secuencias de animación mediante las siguientes etapas de desarrollo:

- Realización del guión.
- Definición de los objetos.
- Especificación de los fotogramas clave.
- Generación de los fotogramas intermedios.

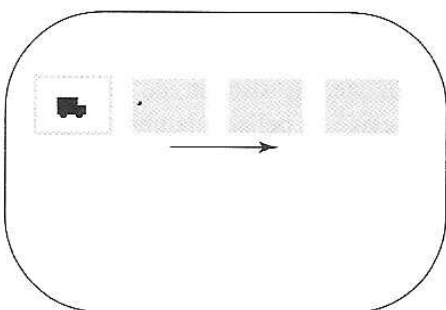


FIGURA 13.1. Animación en tiempo real en un sistema de barrido mediante una tabla de colores.

El **guión** es un resumen de la acción en el que se define la secuencia de movimiento como el conjunto de sucesos básicos que deben tener lugar. Dependiendo del tipo de animación que haya que producir, el guión puede estar compuesto por un conjunto de burdos dibujos y una breve descripción de los movimientos, o puede ser simplemente una lista de las ideas básicas que describen la acción. Originalmente, ese conjunto de burdos dibujos que describen el guión se solía fijar en un panel de gran tamaño que se utilizaba para presentar una vista global del proyecto de animación. De aquí proviene el nombre inglés «*storyboard*».

Para cada participante en la acción se proporciona una **definición del objeto**. Los objetos pueden definirse en términos de las formas básicas, como por ejemplo polígonos o *splines* superficiales. Además, suele proporcionarse una descripción de los movimientos que tengan que realizar cada personaje u objeto descrito en el guión.

Un **fotograma clave** es un dibujo detallado de la escena en un cierto momento de la secuencia de animación. Dentro de cada fotograma clave, cada objeto (o personaje) se posiciona de acuerdo con el tiempo correspondiente a dicho fotograma. Algunos fotogramas clave se eligen en las posiciones extremas de la acción, mientras que otros se espacian para que el intervalo de tiempo entre un fotograma clave y el siguiente no sea excesivo. Para los movimientos intrincados se especifican más fotogramas clave que para los movimientos simples o lentos. El desarrollo de los fotogramas clave suele, por regla general, ser responsabilidad de los animadores expertos, siendo normal que se asigne un animador distinto para cada personaje de la animación.

Los **fotogramas intermedios** (*in-betweens*) son los comprendidos entre los sucesivos fotogramas clave. El número total de imágenes o fotogramas, y por tanto el número total de fotogramas intermedios, necesarios para una animación vendrá determinado por el medio de visualización que se utilice. Las películas requieren 24 imágenes por segundo, mientras que los terminales gráficos se refrescan con una tasa de 60 o más imágenes por segundo. Normalmente, los intervalos temporales de la secuencia se configuran de tal modo que haya entre tres y cinco fotogramas intermedios entre cada par de fotogramas clave sucesivos. Dependiendo de la velocidad especificada para la secuencia, será necesario definir más o menos fotogramas clave. Como ejemplo, una secuencia de película de un minuto de duración contiene un total de 1440 fotogramas; si se requieren cinco fotogramas intermedios entre cada par de fotogramas clave, entonces será necesario desarrollar 288 fotogramas clave.

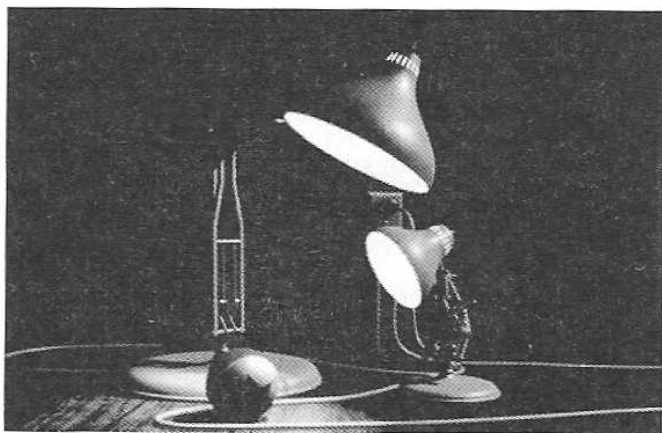


FIGURA 13.2. Una imagen del galardonado corto animado *Luxo Jr.* Esta película fue diseñada utilizando un sistema de animación basado en fotogramas clave y técnicas de dibujos animados con el fin de que las lámparas se muevan con si estuvieran vivas. Las imágenes finales fueron obtenidas con múltiples fuentes luminosas y con técnicas de texturado procedimental. (Cortesía de Pixar. © 1986 Pixar.)

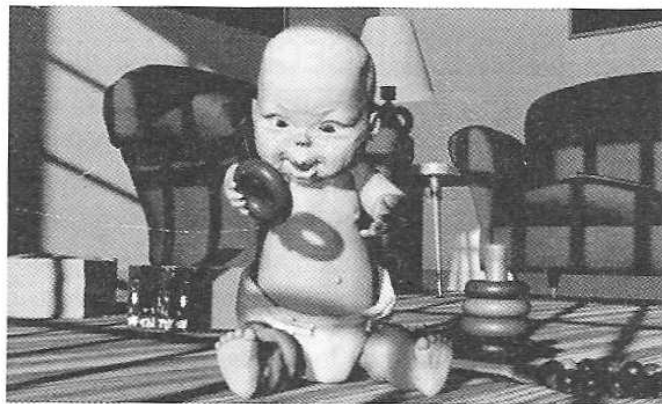


FIGURA 13.3. Un fotograma del corto *Tin Toy*, la primera película de animación por computadora que ganó un Oscar. Diseñada mediante un sistema de animación basada en fotogramas clave, la película también requirió un detallado modelado de las expresiones faciales. Las imágenes finales fueron obtenidas utilizando sombreado procedimental, técnicas de auto-sombreado, desenfoque de movimiento y mapeado de texturas. (Cortesía de Pixar. © 1988 Pixar.)

Puede que sea necesario llevar a cabo diversas otras tareas, dependiendo de la aplicación. Estas tareas adicionales incluyen la verificación del movimiento, la edición y la producción y sincronización de una banda sonora. Muchas de las funciones necesarias para producir animaciones generales se llevan ahora a cabo con ayuda de computadoras. Las Figuras 13.2 y 13.3 muestran ejemplos de imágenes generadas por computadora para secuencias de animación.

13.3 TÉCNICAS TRADICIONALES DE ANIMACIÓN

Los profesionales que trabajan en la producción de películas animadas utilizan diversos métodos para mostrar y enfatizar secuencias de movimientos. Estos métodos incluyen la deformación de los objetos, el espaciado entre fotogramas de la secuencia, la anticipación y seguimiento del movimiento y el enfoque de la acción.

Una de las técnicas más importantes para simular efectos de aceleración, particularmente para los objetos no rígidos, es la técnica de **compresión y expansión**. La Figura 13.4 muestra cómo se utiliza esta técnica para enfatizar la aceleración y deceleración de una bola que rebota contra el suelo. A medida que la bola acelera, comienza a expandirse. Cuando la bola impacta en el suelo y se detiene, primero se comprime y luego se vuelve a expandir a medida que acelera y rebota hacia arriba.

Otra técnica utilizada por los profesionales de las películas animadas es la **temporización**, que hace referencia al espaciado entre fotogramas de la secuencia. Un objeto que se mueva más lentamente se representará mediante fotogramas con un espaciado menor, mientras que un objeto que se mueva rápidamente se mostrará distribuyendo menos fotogramas a lo largo de todo el trayecto de movimiento. Este efecto se ilustra en la Figura 13.5, donde podemos ver que los cambios de posición entre un fotograma y otro se incrementan a medida que se acelera la bola en su caída.

Los movimientos de los objetos también pueden enfatizarse creando acciones preliminares que indiquen una **anticipación** de un movimiento inminente. Por ejemplo, un personaje de dibujos animados puede inclinarse

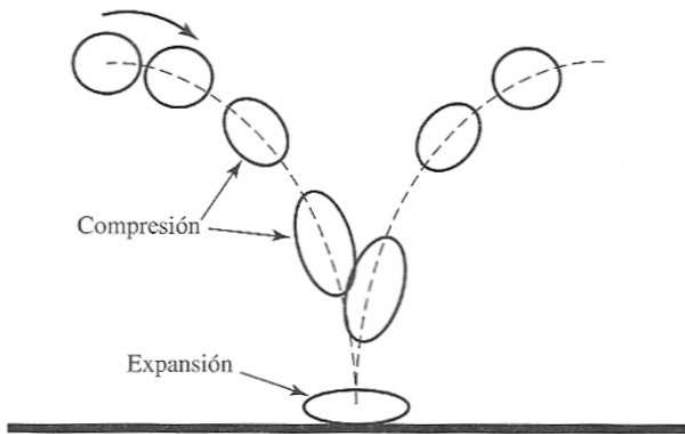


FIGURA 13.4. Ejemplo de rebote de una pelota donde se ilustra la técnica de «compresión y expansión» para enfatizar la aceleración de los objetos.

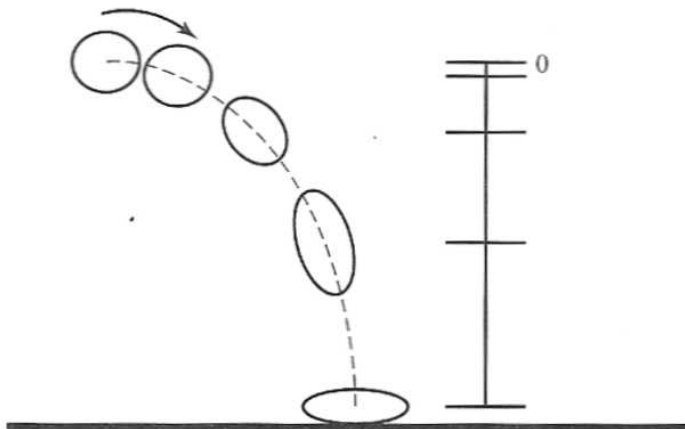


FIGURA 13.5. Los cambios de posición entre fotogramas para el rebote de una bola se incrementan a medida que lo hace la velocidad de una bola.

se hacia adelante y girar el cuerpo antes de comenzar a correr. O bien, otro personaje puede hacer un «molinillo» con los brazos antes de arrojar una bola. De forma similar, las **acciones de seguimiento** pueden utilizarse para enfatizar un movimiento anterior. Después de arrojar una bola, un personaje puede continuar moviendo el brazo hasta volver a acercarlo al cuerpo. O bien, el sombrero de la cabeza de un personaje que se ha detenido abruptamente puede salir volando. Asimismo, las acciones pueden enfatizarse mediante las técnicas de **regulación del punto de atención**, que hacen referencia a cualquier método que permita centrarse en una parte importante de la escena, como por ejemplo aquella parte donde un personaje está ocultando algo.

13.4 FUNCIONES GENERALES DE ANIMACIÓN POR COMPUTADORA

Se han desarrollado muchos paquetes software para diseño de animaciones generales o para realizar tareas de animación especializadas. Las funciones de animación típicas incluyen la gestión del movimiento de los objetos, la generación de vistas de los objetos, la proyección de movimientos de la cámara y la generación de fotogramas intermedios. Algunos paquetes de animación, como por ejemplo Wavefront, proporcionan funciones especiales tanto para el diseño global de animación como para el procesamiento de objetos individuales. Otros son paquetes de propósito especial para características concretas de una animación, como por ejemplo los sistemas para generar fotogramas intermedios o los sistemas para animación de personajes.

En los paquetes de animación generales se suele proporcionar un conjunto de rutinas para la gestión de la base de datos de objetos. Las formas de los objetos y sus parámetros asociados se almacenan y actualizan en la base de datos. Otras funciones de manejo de los objetos incluyen las necesarias para generar los movimientos de los objetos y las empleadas para representar las superficies de los objetos. Los movimientos pueden generarse de acuerdo con restricciones especificadas utilizando transformaciones bidimensionales o tridimensionales. Entonces, pueden aplicarse funciones estándar para identificar las superficies visibles y aplicar los algoritmos de representación.

Otro conjunto típico de funciones simula los movimientos de la cámara. Los movimientos estándar de una cámara son el zoom, las panorámicas y los giros. Finalmente, dada la especificación de los fotogramas clave, pueden generarse automáticamente los fotogramas intermedios.

13.5 LENGUAJES DE ANIMACIÓN POR COMPUTADORA

Podemos desarrollar rutinas para diseñar y controlar las secuencias de animación mediante un lenguaje de programación de propósito general, como C, C++, Lisp o Fortran, pero también se han desarrollado diversos lenguajes especializados de animación. Estos lenguajes incluyen normalmente un editor gráfico, un generador de fotogramas clave, un generador de fotogramas intermedios y una serie de rutinas gráficas estándar. El editor gráfico permite al animador diseñar y modificar formas de objetos, utilizando *splines* de superficie, métodos constructivos de geometría sólida u otros esquemas de representación.

Una tarea importante dentro de la especificación de la animación es la *descripción de la escena*. Esto incluye el posicionamiento de los objetos y las fuentes luminosas, la definición de los parámetros fotométricos (intensidades de las fuentes luminosas y propiedades de iluminación de las superficies) y la configuración de los parámetros de la cámara (posición, orientación y características del objetivo). Otra función estándar en este tipo de lenguajes es la *especificación de acciones*, que implica la disposición de las trayectorias de movimiento correspondientes a los objetos y a la cámara. Y también necesitamos las rutinas gráficas usuales: transformaciones de visualización y de perspectiva, transformaciones geométricas para generar movimientos de los objetos en función de las aceleraciones o especificaciones de proyectos cinemáticos, identificación de superficies visibles y operaciones de representación de las superficies.

Los **sistemas de fotogramas clave** fueron diseñados originalmente como un conjunto separado de las rutinas de animación para generar los fotogramas intermedios a partir de los fotogramas clave especificados por los usuarios. Ahora, estas rutinas suelen formar parte de paquetes de animación más generales. En el caso más

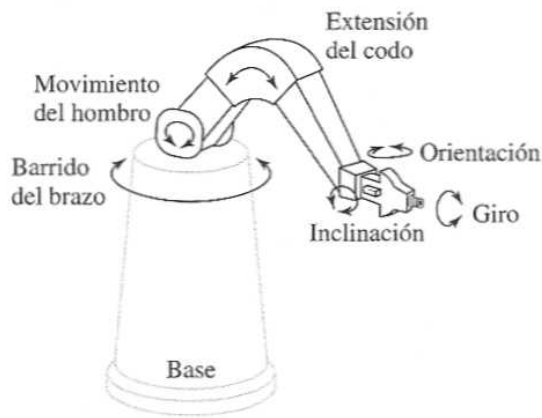


FIGURA 13.6. Grados de libertad para un robot estacionario de un único brazo.

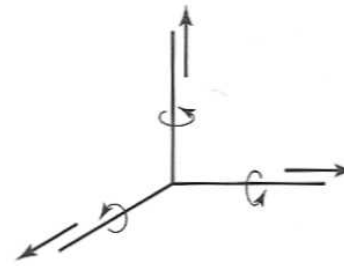


FIGURA 13.7. Grados de libertad de traslación y rotación para la base del brazo robotizado.

simple, cada objeto de la escena se define como un conjunto de cuerpos rígidos conectados en las uniones y con un número limitado de grados de libertad. Por ejemplo, el robot de un único brazo de la Figura 13.6 tiene seis grados de libertad, que se denominan barrido del brazo, movimiento del hombro, extensión del codo, inclinación, orientación y giro. Podemos ampliar el número de grados de libertad de este brazo robotizado a nueve permitiendo una traslación tridimensional de la base (Figura 13.7). Si también permitimos rotaciones de la base, el brazo robotizado puede tener un total de doce grados de libertad. Por comparación, el cuerpo humano tiene más de 200 grados de libertad.

Los **sistemas parametrizados** permiten especificar las características del movimiento de los objetos como parte de la definición de esos mismos objetos. Los parámetros ajustables controlan características de los objetos tales como grados de libertad, las limitaciones del movimiento y los cambios permitidos en la forma.

Los **sistemas de script** permiten definir las especificaciones de los objetos y las secuencias de animación mediante un *script* introducido por el usuario. Mediante el *script*, puede construirse una biblioteca de objetos y movimientos diversos.

13.6 SISTEMAS DE FOTOGRAMAS CLAVE

Puede utilizarse un sistema de fotogramas clave para generar un conjunto de fotogramas intermedios a partir de la especificación de dos (o más) fotogramas clave. Los trayectos de movimiento pueden especificarse mediante una *descripción cinemática* como un conjunto de *splines* curvas, o bien pueden *fundamentarse físicamente* los movimientos especificando las fuerzas que actúan sobre los objetos que hay que animar.

Para las escenas complejas, podemos separar los fotogramas en componentes u objetos individuales denominados *cels* (transparencias de celuloide). Este término fue acuñado en el mundo de las técnicas de dibujos animados, donde el fondo y cada uno de los personajes de una escena se dibujaba en una transparencia separada. Entonces, apilando las transparencias por orden, desde el fondo hasta el primer plano, se fotografiaba el conjunto para obtener el fotograma completo. Con este sistema, se utilizan los trayectos de animación especificados para obtener el siguiente *cel* de cada personaje, interpolando las posiciones a partir de los tiempos correspondientes a los fotogramas clave.

Con transformaciones complejas de los objetos, las formas de los objetos pueden cambiar a lo largo del tiempo. Como ejemplos podríamos citar las ropas, las características faciales, las ampliaciones de determinados detalles, las formas evolutivas y la explosión o desintegración de los objetos. Para las superficies descritas con mallas poligonales, estos cambios pueden dar como resultado modificaciones significativas en la forma de los polígonos, por lo que el número de vistas de un polígono podría ser distinto entre un fotograma y el siguiente. Estos cambios se incorporan en el desarrollo de los fotogramas intermedios añadiendo o eliminando aristas a los polígonos de acuerdo con los requisitos impuestos por los fotogramas clave correspondientes.

Morfismo

La modificación de la forma de un objeto, para que éste adopte una forma distinta, se denomina **morfismo**, palabra que proviene de «metamorfosis». Un animador puede modelar un morfismo haciendo que las formas de los polígonos efectúen una transición a lo largo de los fotogramas intermedios comprendidos entre un fotograma clave y el siguiente.

Dados dos fotogramas clave, cada uno de ellos con un número diferente de segmentos de línea que especifican la transformación de un objeto, podemos primero ajustar la especificación del objeto en uno de los fotogramas de modo que el número de aristas poligonales (o el número de vértices de los polígonos) sea el mismo para los dos fotogramas. Esta etapa de preprocesamiento se ilustra en la Figura 13.8. Un segmento de línea recta en el fotograma clave k se transforma en dos segmentos de línea en el fotograma clave $k + 1$. Puesto que el fotograma $k + 1$ tiene un vértice adicional, añadimos un vértice entre los vértices 1 y 2 en el fotograma clave k para equilibrar el número de vértices (y aristas) en los dos fotogramas clave. Utilizando interpolación lineal para generar los fotogramas intermedios, efectuamos la transición del vértice añadido en el fotograma clave k hacia el vértice $3'$ según el trayecto lineal mostrado en la Figura 13.9. En la Figura 13.10 se proporciona un ejemplo de un triángulo que se expande linealmente en un cuadrilátero. Las Figuras 13.11 y 13.12 muestran ejemplos de morfismo en anuncios de televisión.

Podemos enunciar reglas de procesamiento generales para ecualizar los fotogramas clave en términos del número de aristas o del número de vértices que haya que añadir a un fotograma clave. Vamos a considerar primero la ecualización del número de aristas, donde los parámetros L_k y L_{k+1} denotan el número de segmentos de línea en dos fotogramas consecutivos. El número máximo y mínimo de líneas que habrá que ecualizar será el siguiente:

$$L_{\max} = \max(L_k, L_{k+1}), \quad L_{\min} = \min(L_k, L_{k+1}) \quad (13.1)$$

A continuación, calculamos los siguientes dos valores:

$$\begin{aligned} N_e &= L_{\max} \bmod L_{\min} \\ N_s &= \text{int} \left(\frac{L_{\max}}{L_{\min}} \right) \end{aligned} \quad (13.2)$$

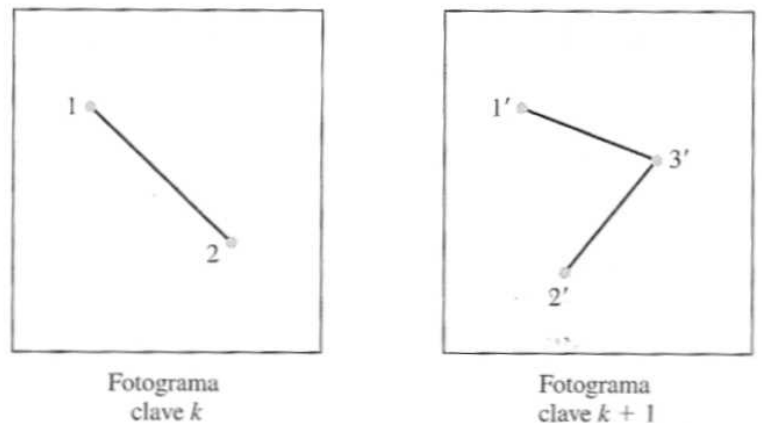


FIGURA 13.8. Una arista con dos vértices 1 y 2 en el fotograma clave k evoluciona para convertirse en dos aristas conectadas en fotograma clave $k + 1$.

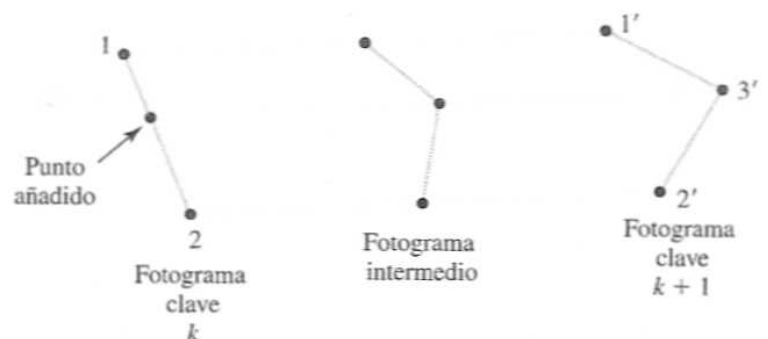


FIGURA 13.9. Interpolación lineal para transformar un segmento de línea en el fotograma clave k en dos segmentos de línea conectados en el fotograma clave $k + 1$.

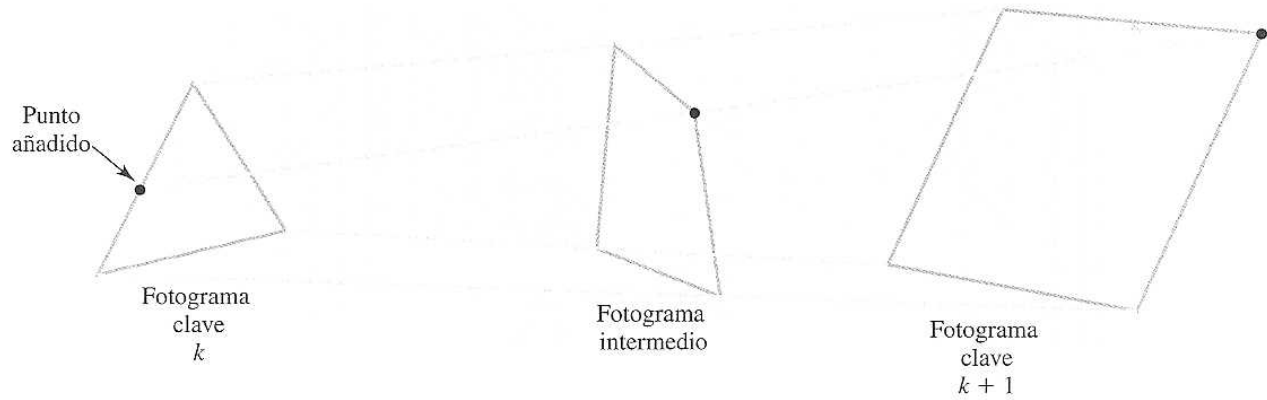


FIGURA 13.10. Interpolación lineal para transformar un triángulo en un cuadrilátero.

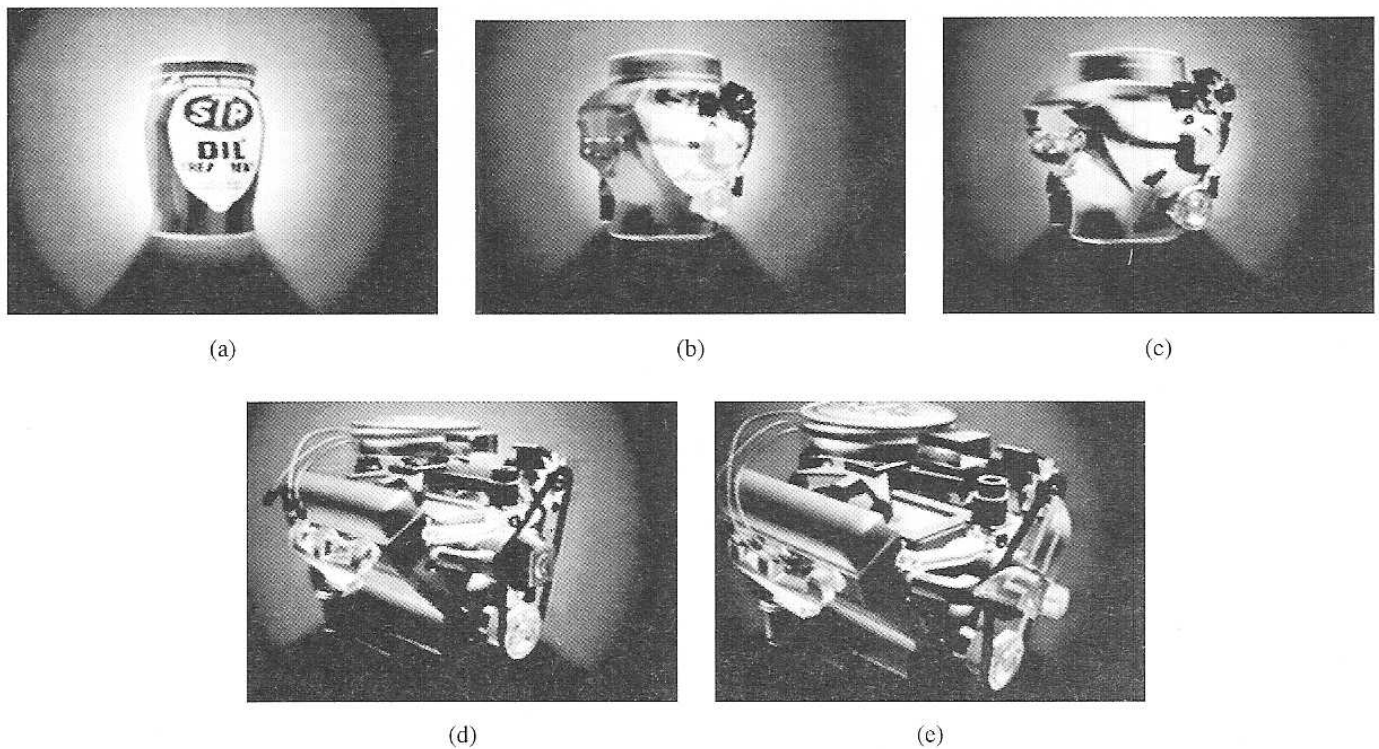


FIGURA 13.11. Transformación de una lata de aceite para automóviles STP en un motor de automóvil. (Cortesía de Silicon Graphics, Inc.)

Las etapas de preprocesamiento para la eculización de aristas pueden llevarse a cabo entonces con los dos siguientes procedimientos:

- (1) Dividir N_e aristas de $fotogramaclave_{\min}$ en $N_s + 1$ secciones.
- (2) Dividir las líneas restantes de $fotogramaclave_{\min}$ en N_s secciones.

Como ejemplo, si $L_k = 15$ y $L_{k+1} = 11$, dividiríamos cuatro líneas de $fotogramaclave_{k+1}$ en dos secciones cada una. Las líneas restantes de $fotogramaclave_{k+1}$ se dejan intactas.

Si eculizamos el número de vértices, podemos utilizar los parámetros V_k y V_{k+1} para denotar el número de vértices en dos fotogramas clave consecutivos. En este caso, determinamos los números máximo y mínimo de vértices de la forma siguiente:

$$V_{\max} = \max(V_k, V_{k+1}), \quad V_{\min} = \min(V_k, V_{k+1}) \quad (13.3)$$

A continuación calculamos los siguientes dos valores:

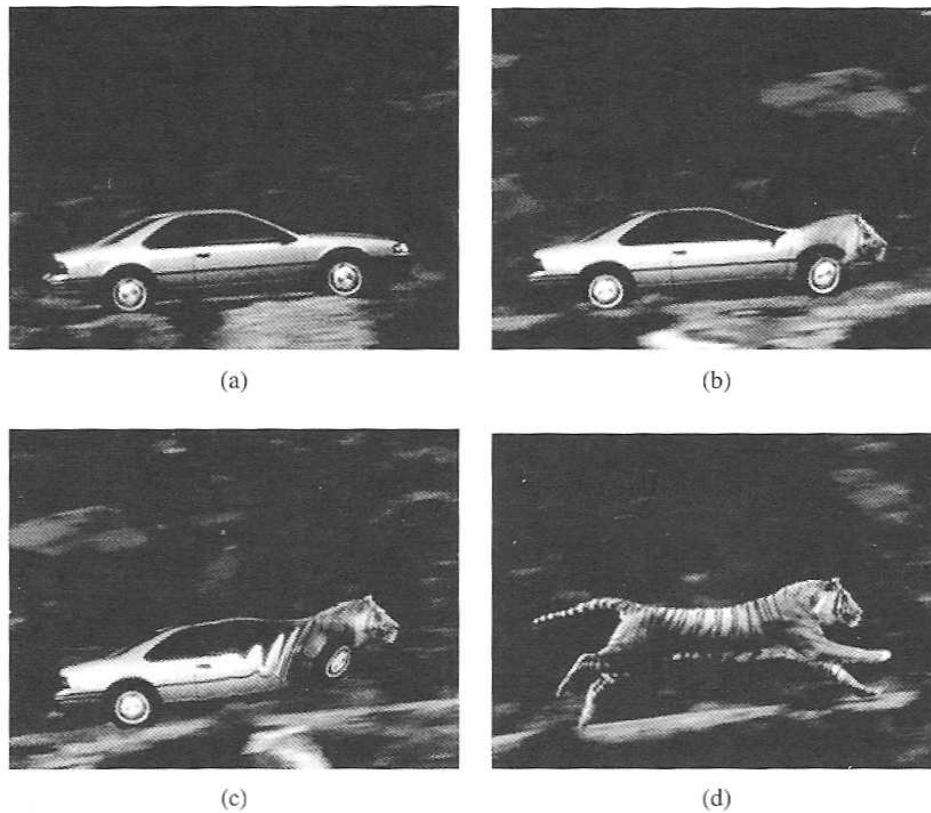


FIGURA 13.12. Transformación de un automóvil en un tigre. (Cortesía de Exxon Company USA y Pacific Data Images.)

$$N_{ls} = (V_{\max} - 1) \bmod (V_{\min} - 1) \quad (13.4)$$

$$N_p = \text{int} \left(\frac{V_{\max} - 1}{V_{\min} - 1} \right)$$

Estos dos valores se utilizan entonces para llevar a cabo la ecualización de vértices mediante los procedimientos:

- (1) Añadir N_p puntos N_{ls} secciones de línea de *fotogramaclave*_{min}.
- (2) Añadir $N_p - 1$ puntos a las aristas restantes de *fotogramaclave*_{min}.

Para el ejemplo de transformación de un triángulo en un cuadrilátero, $V_k = 3$ y $V_{k+1} = 4$. Tanto N_{ls} como N_p son 1, por lo que añadiríamos un punto a una arista de *fotogramaclave*_k. No se añadiría ningún punto a las líneas restantes de *fotogramaclave*_k.

Simulación de aceleraciones

A menudo se utilizan técnicas de ajuste de curvas para especificar los trayectos de animación entre fotogramas clave. Dadas las posiciones de los vértices en los fotogramas clave, podemos ajustar las posiciones mediante trayectos lineales o no lineales. La Figura 13.13 ilustra un ajuste no lineal de las posiciones en los fotogramas clave. Y para simular aceleraciones, podemos ajustar el espaciado temporal correspondiente a los fotogramas intermedios.

Si el movimiento debe tener lugar a velocidad constante (aceleración cero), utilizamos intervalos temporales iguales para los fotogramas intermedios. Por ejemplo, con n fotogramas intermedios y con sendos tiempos t_1 y t_2 para los fotogramas clave (Figura 13.14), el intervalo temporal entre los fotogramas clave se divide en $n + 1$ subintervalos iguales, lo que nos da un espaciado de los fotogramas intermedios igual a:

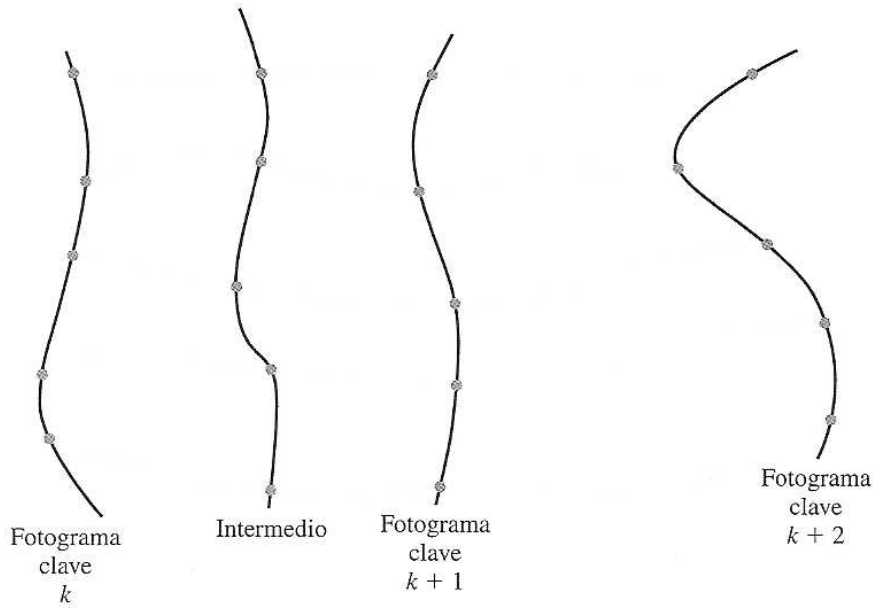


FIGURA 13.13. Ajuste de las posiciones de los vértices en los fotogramas clave mediante *splines* no lineales.

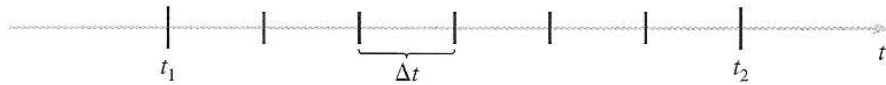


FIGURA 13.14. Posiciones de los fotogramas intermedios para movimiento a velocidad constante.

$$\Delta t = \frac{t_2 - t_1}{n + 1} \tag{13.5}$$

El tiempo correspondiente al fotograma intermedio *j*-ésimo será:

$$tB_j = t_1 + j\Delta t, \quad j = 1, 2, \dots, n \tag{13.6}$$

y este tiempo se utiliza para calcular las coordenadas de posición, los colores y otros parámetros físicos para dicho fotograma de la secuencia.

Usualmente, hacen falta cambios de velocidad (aceleración distinta de cero) en algún punto de las secuencias de animación o de dibujos animados, particularmente al principio y al final de un movimiento. Las partes de arranque y de parada de un trayecto de animación se suelen modelar con *splines* o funciones trigonométricas, pero también se han aplicado funciones temporales parabólicas y cúbicas para modelar las aceleraciones. Los paquetes de animación suelen proporcionar funciones trigonométricas para simular las aceleraciones. Para modelar una velocidad creciente (aceleración positiva), lo que puede hacerse es incrementar el espaciado temporal entre fotogramas, de modo que se produzcan cambios más grandes en la posición a medida que aumenta la velocidad del objeto. Podemos obtener un tamaño creciente para el intervalo temporal mediante la función:

$$1 - \cos \theta, \quad 0 < \theta < \pi/2$$

Para *n* fotogramas intermedios, el tiempo correspondiente al fotograma intermedio *j*-ésimo se calcularía como:

$$tB_j = t_1 + j\Delta t \left[1 - \cos \frac{j\pi}{2(n+1)} \right], \quad j = 1, 2, \dots, n \tag{13.7}$$

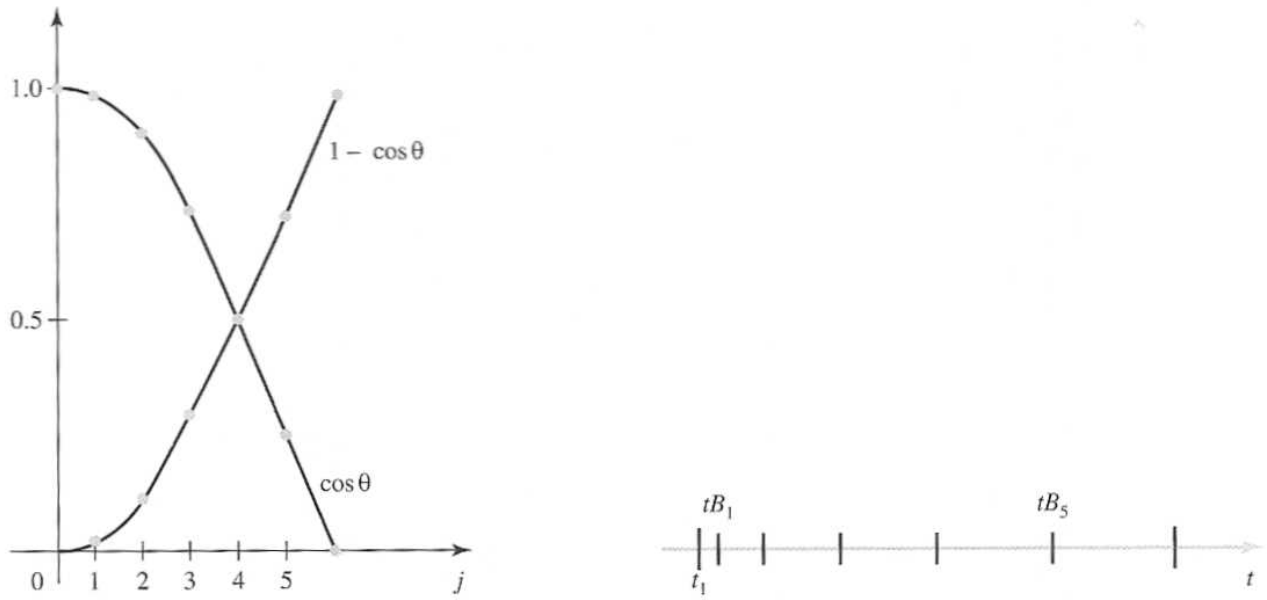


FIGURA 13.15. Una función de aceleración trigonométrica y el correspondiente espaciado de fotogramas intermedios para $n = 5$ y $\theta = j\pi/12$ en la Ecuación 13.7, lo que produce cambios crecientes en los valores de las coordenadas a medida que el objeto pasa de un intervalo temporal a otro.

donde Δt es la diferencia de tiempo entre los dos fotogramas clave. La Figura 13.15 muestra una gráfica de la función trigonométrica de aceleración y del espaciado de los fotogramas intermedios para $n = 5$.

Podemos modelar una velocidad decreciente (deceleración) utilizando la función $\sin \theta$, con $0 < \theta < \pi/2$. El tiempo correspondiente a un fotograma intermedio se determina entonces mediante la fórmula

$$tB_j = t_1 + \Delta t \sin \frac{j\pi}{2(n+1)}, \quad j = 1, 2, \dots, n \tag{13.8}$$

En la Figura 13.16 se muestra una gráfica de esta función y el tamaño decreciente de los intervalos temporales, para cinco fotogramas intermedios.

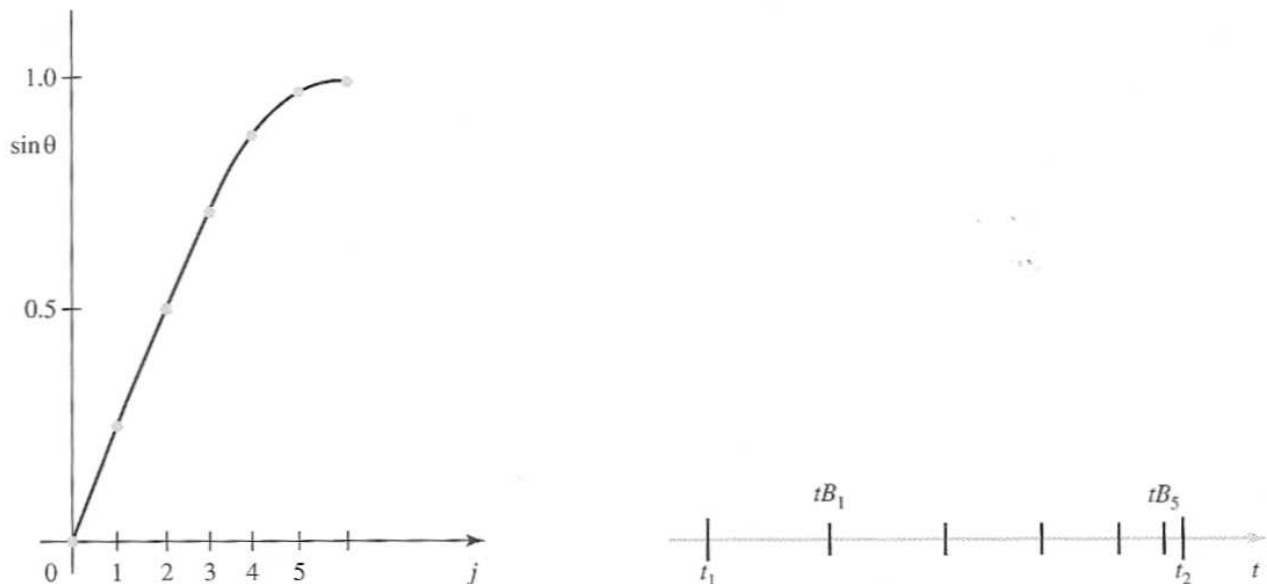


FIGURA 13.16. Una función de deceleración trigonométrica y el correspondiente espaciado de los fotogramas intermedios para $n = 5$ y $\theta = j\pi/12$ en la Ecuación 13.8, lo que produce cambios decrecientes en las coordenadas a medida que el objeto pasa de un intervalo temporal a otro.

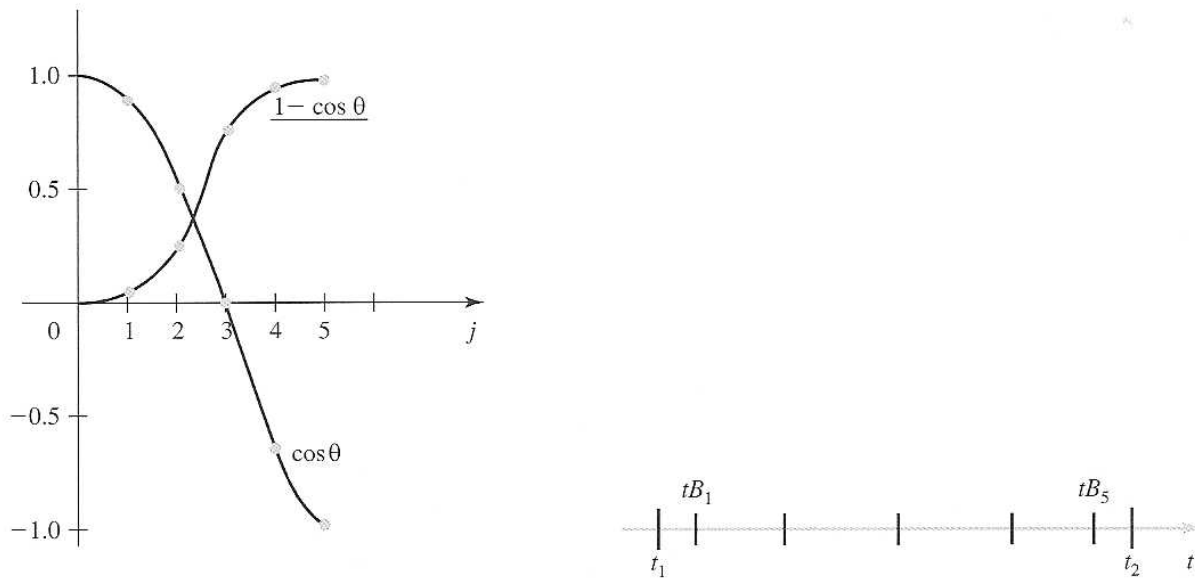


FIGURA 13.17. La función trigonométrica de aceleración-deceleración $(1 - \cos \theta)/2$ y el correspondiente espaciado de los fotogramas intermedios para $n = 5$ en la Ecuación 13.9.

A menudo los movimientos contienen tanto aceleraciones como frenados. Podemos modelar una combinación de velocidad creciente-decreciente incrementando primero el espaciado temporal de los fotogramas intermedios y luego reduciéndolo. Una función para poder conseguir estos cambios de carácter temporal es:

$$\frac{1}{2}(1 - \cos \theta), \quad 0 < \theta < \pi / 2$$

El tiempo correspondiente al fotograma intermedio j -ésimo se calculará ahora como:

$$tB_j = t_1 + \Delta t \left\{ \frac{1 - \cos [j\pi / (n + 1)]}{2} \right\}, \quad j = 1, 2, \dots, n \tag{13.9}$$

donde Δt denota la diferencia temporal entre dos fotogramas clave. Los intervalos temporales para un objeto en movimiento se incrementarán primero y luego se reducirán, como se muestra en la Figura 13.17.

El procesamiento de los fotogramas intermedios se simplifica si modelamos inicialmente objetos «esqueleto» (alámbricos), de modo que puedan ajustarse interactivamente a las secuencias de movimiento. Después de definir completamente la secuencia de animación, puede procederse a obtener la representación de los objetos en imágenes.

13.7 ESPECIFICACIONES DE MOVIMIENTOS

Los métodos generales para describir una secuencia de animación van desde la especificación explícita de las trayectorias de movimiento hasta una descripción de las interacciones que producen dichos movimientos. Así, podemos definir el modo en que una animación debe transcurrir proporcionando los parámetros de transformación, los parámetros de las trayectorias de movimiento, las fuerzas que deben actuar sobre los objetos o los detalles sobre cómo interactúan los objetos con el fin de producir movimientos.

Especificación directa del movimiento

El método más simple para definir una animación consiste en la *especificación directa del movimiento*, lo que nos da los parámetros de transformación geométrica. Con este método, lo que hacemos es configurar explíci-

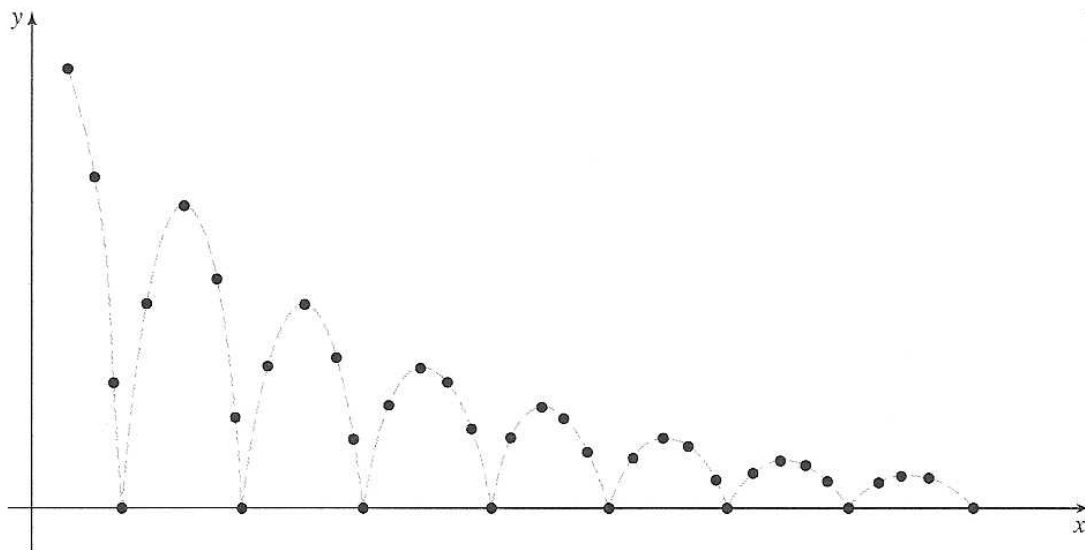


FIGURA 13.18. Aproximación del movimiento de una bola que rebota sobre el suelo mediante una función seno amortiguada (Ecuación 13.10).

tamente los valores de los ángulos de rotación y de los vectores de traslación. A continuación, se aplican las matrices de transformación geométrica para transformar las posiciones de coordenadas. Alternativamente, podríamos usar una ecuación de aproximación en la que aparezcan estos parámetros para especificar ciertos tipos de movimientos. Por ejemplo, podemos aproximar la trayectoria de una bola que rebote en el suelo utilizando una curva seno amortiguada y rectificada (Figura 13.18):

$$y(x) = A|\sin(\omega x + \theta_0)|e^{-kx} \quad (13.10)$$

donde A es la amplitud inicial (altura de la bola sobre el suelo), ω es la frecuencia angular, θ_0 es el ángulo de fase y k es el coeficiente de amortiguamiento. Este método de especificación del movimiento resulta particularmente útil para las secuencias de animación simples programadas por el usuario.

Sistemas dirigidos por objetivos

En el extremo opuesto, podemos especificar los movimientos que deben tener lugar en términos generales que describan de manera abstracta las acciones en función de los resultados finales. En otras palabras, una animación se especifica en términos del estado final de los movimientos. Estos sistemas se denominan *dirigidos por objetivos*, ya que los valores de los parámetros de movimiento se determinan a partir de los objetivos de la animación. Por ejemplo, podríamos especificar que un determinado objeto debe «caminar» o «correr» hasta un destino concreto. O podríamos especificar que queremos que el objeto «agarre» a algún otro objeto especificado. Las directivas de entrada se interpretan entonces en términos de los movimientos componentes que permitirán llevar a cabo la tarea descrita. El movimiento de los seres humanos, por ejemplo, puede definirse como una estructura jerárquica de submovimiento para el torso, los miembros, etc. Así, cuando se proporciona un objeto tal como «caminar hasta la puerta» se calculan los movimientos requeridos del torso y de los miembros para llevar a cabo esta acción.

Cinemática y dinámica

También podemos construir secuencias de animación utilizando descripciones *cinemáticas* o *dinámicas*. Con una descripción cinemática, especificamos la animación proporcionando los parámetros de movimiento (posición, velocidad y aceleración) sin referencia a las causas ni a los objetivos del movimiento. Para una velocidad constante (aceleración cero), designamos los movimientos de los cuerpos rígidos de una escena proporcionando una posición inicial y un vector de velocidad para cada objeto. Como ejemplo, si la velocidad se

específica como $(3, 0, -4)$ km/seg, entonces este vector proporcionará la dirección de la trayectoria lineal de movimiento y la velocidad (módulo del vector) será igual a 5 km/seg. Si también especificamos las aceleraciones (tasa de cambio de la velocidad), podemos modelar arranques, paradas y trayectos de movimiento curvos. La especificación cinemática de un movimiento también puede proporcionarse simplemente describiendo la trayectoria del movimiento. Esto se suele hacer mediante curvas de tipo *spline*.

Un enfoque alternativo consiste en utilizar *cinemática inversa*. Con este método, especificamos las posiciones inicial y final de los objetos en instantes determinados y es el sistema el que se encarga de calcular los parámetros del movimiento. Por ejemplo, suponiendo una aceleración cero, podemos determinar la velocidad constante que permitirá conseguir el movimiento de un objeto desde la posición inicial hasta la posición final. Este método se suele utilizar para objetos complejos proporcionando las posiciones y orientaciones de un nodo terminal de un objeto, como por ejemplo una mano o un pie. El sistema determina entonces los parámetros de movimiento de los otros nodos que hacen falta para conseguir el movimiento deseado.

Las descripciones dinámicas, por el contrario, requieren la especificación de las fuerzas que producen las velocidades y aceleraciones. La descripción del comportamiento de los objetos en términos de la influencia de las fuerzas se suele denominar *modelado físico* (Capítulo 8). Como ejemplos de fuerzas que afectan al movimiento de los objetos podemos citar las fuerzas electromagnéticas, gravitatorias, de fricción y otras fuerzas mecánicas.

Los movimientos de los objetos se obtienen a partir de las ecuaciones de las fuerzas que describen leyes físicas, como por ejemplo las leyes de Newton del movimiento para los procesos gravitatorios y de fricción, las ecuaciones de Euler o de Navier-Stokes que describen el flujo de fluidos y las ecuaciones de Maxwell para las fuerzas electromagnéticas. Por ejemplo, la forma general de la segunda ley de Newton para una partícula de masa m es:

$$\mathbf{F} = \frac{d}{dt}(m\mathbf{v}) \quad (13.11)$$

donde \mathbf{F} es el vector de la fuerza y \mathbf{v} es el vector velocidad. Si la masa es constante, resolvemos la ecuación $\mathbf{F} = m\mathbf{a}$, donde \mathbf{a} representa el vector de aceleración. En caso contrario, la masa será una función del tiempo, como sucede en el movimiento relativista o en el movimiento de naves espaciales que consuman cantidades no despreciables de combustible por unidad de tiempo. También podemos utilizar la *dinámica inversa* para obtener las fuerzas, dadas las posiciones inicial y final de los objetos y el tipo de movimiento requerido.

Entre las aplicaciones del modelado físico se incluyen los sistemas complejos de cuerpos rígidos y también otros sistemas no rígidos como las ropas y los materiales plásticos. Normalmente, se utilizan métodos numéricos para obtener los parámetros de movimiento incrementalmente a partir de las ecuaciones dinámicas, utilizando condiciones iniciales de valores de contorno.

13.8 ANIMACIÓN DE FIGURAS ARTICULADAS

Una técnica básica para animar personajes humanos, animales, insectos y otras criaturas consiste en modelarlas como **figuras articuladas**, que son estructuras jerárquicas compuestas de un conjunto de enlaces rígidos conectados mediante uniones rotatorias (Figura 13.19). En términos menos formales, esto simplemente quiere decir que modelamos los objetos animados como si fueran esqueletos simplificados, los cuales podemos envolver luego con superficies que representen la piel, el pelo, las plumas, las ropas u otros tipos de recubrimientos.

Los puntos de conexión de una figura articulada se sitúan en los hombros, las caderas, las rodillas y otras articulaciones del esqueleto, y esos puntos de unión siguen unas trayectorias de movimiento especificadas a medida que el cuerpo se traslada. Por ejemplo, cuando se especifica un movimiento para un objeto, el hombro se mueve automáticamente de una cierta forma y, a medida que el hombro se mueve, los brazos también lo hacen. Con estos sistemas, se definen diferentes tipos de movimientos, como por ejemplo andar, correr o saltar, y esos movimientos se asocian con movimientos concretos de las uniones y de los enlaces conectados a ellas.

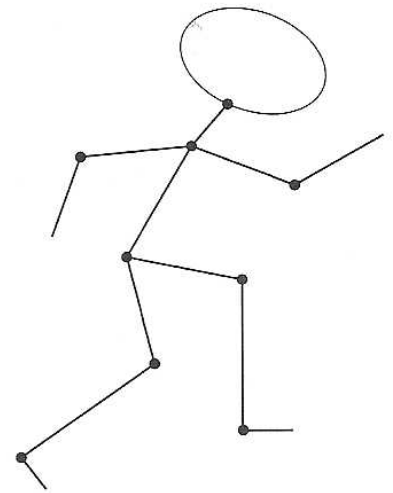


FIGURA 13.19. Una figura simple articulada con nueve uniones y doce enlaces conectados, sin contar la cabeza oval.

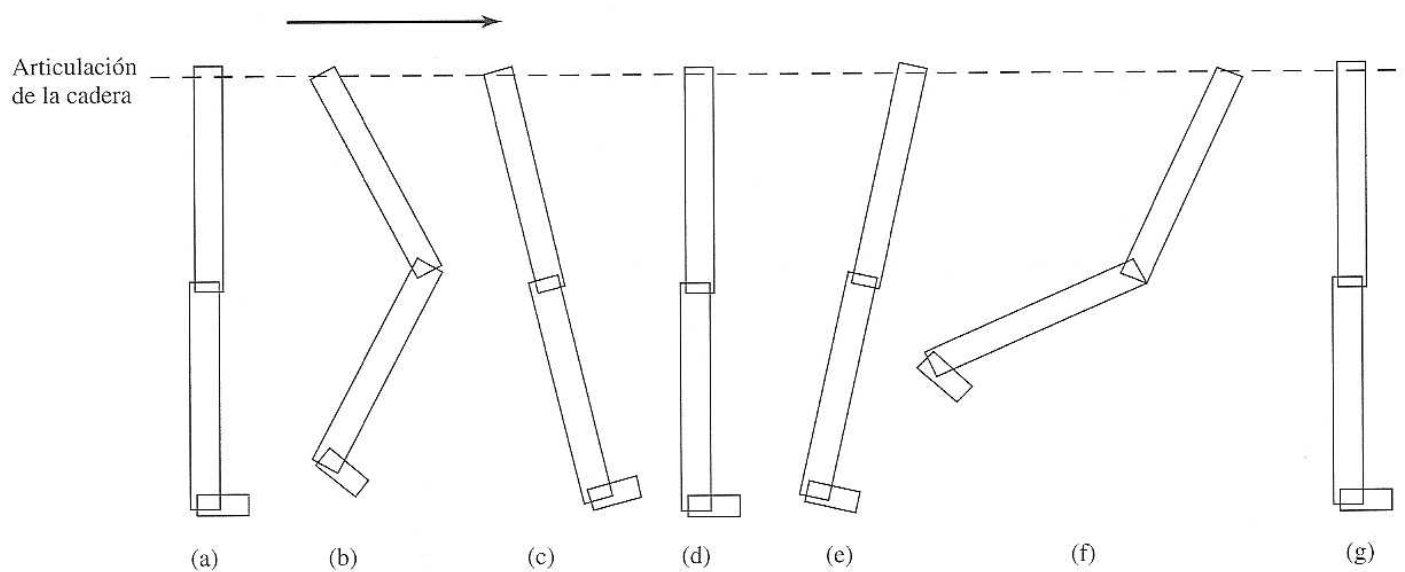


FIGURA 13.20. Posibles movimientos para un conjunto de enlaces conectados que representan una pierna llevando a cabo la acción de andar.

Por ejemplo, podríamos definir como en la Figura 13.20 un conjunto de movimientos para una pierna que estuviera efectuando la acción de andar. La articulación de la cadera se mueve hacia adelante según una línea horizontal, mientras que los enlaces conectados realizan una serie de movimientos en torno a las articulaciones de la cadera, de la rodilla y del talón. Comenzando con una pierna recta (Figura 13.20(a)), el primer movimiento consiste en doblar la rodilla a medida que la cadera se mueve hacia adelante (Figura 13.20(b)). Entonces, la pierna avanza, vuelve a la posición vertical y empuja hacia atrás, como se muestra en las Figuras 13.20(c), (d) y (e). Los movimientos finales son un movimiento amplio hacia atrás y una vuelta a la posición vertical, como en las Figuras 13.20(f) y (g). Este ciclo de movimiento se repite mientras dure la animación, a medida que el personaje recorre una distancia especificada o hasta que transcurra un determinado intervalo de tiempo.

A medida que se mueve un personaje, se incorporan otros movimientos a las diversas articulaciones. Puede aplicarse un movimiento sinusoidal, a menudo de amplitud variable, a las caderas para que éstas se muevan con respecto al torso. De la misma forma, puede aplicarse un movimiento giratorio a los hombros y también la cabeza puede moverse hacia arriba y hacia abajo.

En la animación de personajes se utilizan tanto descripciones cinemáticas del movimiento como descripciones basadas en cinemática inversa. La especificación del movimiento de las articulaciones suele ser una tarea no demasiado complicada, pero la cinemática inversa puede también resultar útil para generar movi-

mientos simples sobre un terreno arbitrario. Para una figura complicada, la cinemática inversa puede no producir una secuencia de animación única, ya que, por ejemplo, puede que sean posibles muchos movimientos rotatorios distintos para un conjunto especificado de condiciones iniciales y finales. En tales casos, puede obtenerse una solución única añadiendo más restricciones al sistema, como por ejemplo el principio de conservación de la cantidad de movimiento.

13.9 MOVIMIENTOS PERIÓDICOS

Cuando construimos una animación con patrones de movimiento repetitivos, como por ejemplo un objeto giratorio, necesitamos asegurarnos de muestrear el movimiento (Sección 4.17) con la suficiente frecuencia como para representar los movimientos correctamente. En otras palabras, el movimiento debe estar sincronizado con la tasa de generación de imágenes, para poder mostrar un número de imágenes por ciclo lo suficientemente alto como para que se perciba el movimiento real. En caso contrario, puede que la animación se visualice de forma incorrecta.

Un ejemplo típico de imágenes de movimiento periódico submuestreadas es la rueda de tren en una película del Oeste que parece estar girando en la dirección incorrecta. La Figura 13.21 ilustra un ciclo completo de la rotación de una rueda de tren, con un radio de distinto color que da 18 vueltas por segundo en el sentido de las agujas del reloj. Si este movimiento se graba en una película a la velocidad normal de proyección de 24 imágenes por segundo, entonces las primeras cinco imágenes correspondientes a este movimiento serían las que se muestran en la Figura 13.22. Puesto que la rueda completa $\frac{3}{4}$ de vuelta cada $\frac{1}{24}$ de segundo, sólo se genera un fotograma de la animación por cada ciclo y la rueda parece estar girando en la dirección opuesta (en sentido contrario a las agujas del reloj).

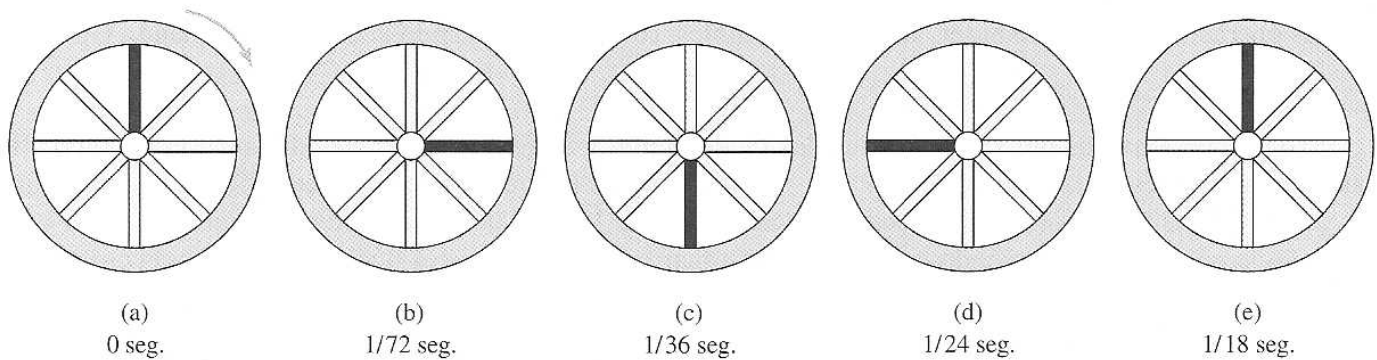


FIGURA 13.21. Cinco posiciones del radio durante un ciclo de movimiento de una rueda que está girando a 18 revoluciones por segundo.

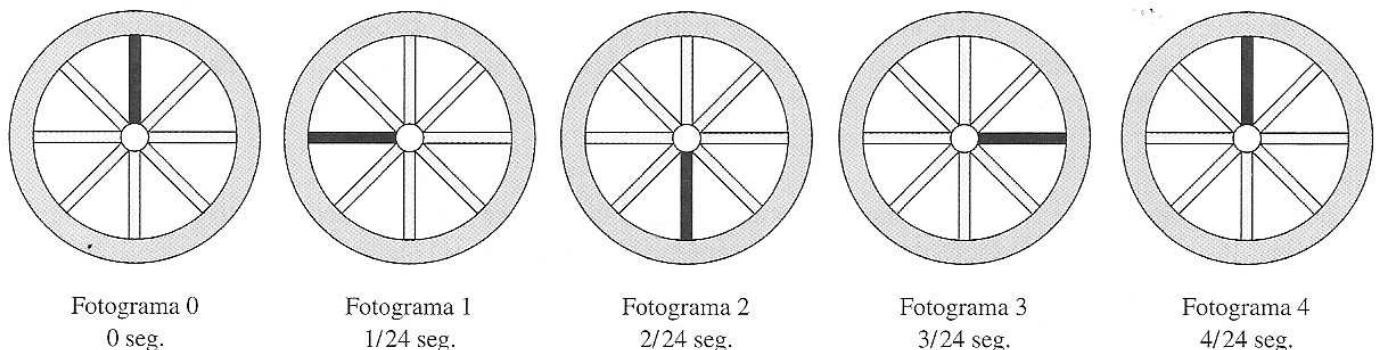


FIGURA 13.22. Los cinco primeros fotogramas de película para la rueda giratoria de la Figura 13.21, producidos con una velocidad de 24 imágenes por segundo.

En las animaciones generadas por computadora, podemos controlar la velocidad de muestreo de un movimiento periódico ajustando los parámetros de movimiento. Por ejemplo, podemos configurar el incremento angular para el movimiento de un objeto giratorio de modo que se generen múltiples fotogramas en cada revolución. Así, un incremento de 3° para un ángulo de rotación produce 120 pasos de movimiento durante una revolución, mientras que un incremento de 4° genera 90 posiciones. Para movimientos más rápidos, pueden utilizarse pasos de rotación más amplios, siempre y cuando el número de muestras por ciclo no sea demasiado pequeño y el movimiento se visualice claramente. Cuando haya que animar objetos complejos, también deberemos tener en cuenta el efecto que el tiempo de generación del fotograma pueda tener sobre la tasa de refresco, como se explica en la Sección 13.1. El movimiento de un objeto complejo puede ser mucho más lento de lo que deseamos si se tarda demasiado en generar cada fotograma de animación.

Otro factor que tenemos que considerar en la visualización de un movimiento repetitivo es el efecto de los redondeos en los cálculos de los parámetros de movimiento. Como hemos observado en la Sección 5.4, podemos reinicializar periódicamente los valores de los parámetros para evitar que la acumulación de los errores produzca movimientos erráticos. Para una rotación continua, podríamos reinicializar los valores de los parámetros una vez por ciclo (360°).

13.10 PROCEDIMIENTOS DE ANIMACIÓN EN OpenGL

En la biblioteca básica hay disponibles operaciones de manipulación de imágenes de barrido (Sección 5.7) y funciones de asignación de índices de colores, mientras que en GLUT hay disponibles rutinas para modificar los valores de las tablas de colores (Sección 4.3). Otras operaciones de animación por barrido sólo están disponibles como rutinas GLUT, porque dependen del sistema de gestión de ventanas que se utilice. Además, algunas características para animación por computadora tales como el doble búfer pueden no estar incluidas en algunos sistemas hardware.

Las operaciones de doble búfer, si están disponibles, se activan utilizando el siguiente comando GLUT:

```
glutInitDisplayMode (GLUT_DOUBLE);
```

Esto proporciona dos búferes, denominados *búfer frontal* y *búfer trasero*, que podemos utilizar alternativamente para refrescar la imagen de pantalla. Mientras uno de los búferes actúa como búfer de refresco para la ventana de visualización actual, puede irse construyendo la siguiente imagen de la animación en el otro búfer. Podemos especificar cuándo hay que intercambiar los roles de los dos búferes mediante el comando:

```
glutSwapBuffers ( );
```

Para determinar si están disponibles las operaciones de doble búfer en un sistema, puede efectuarse la siguiente consulta:

```
glGetBooleanv (GL_DOUBLEBUFFER, status);
```

Se devolverá un valor `GL_TRUE` al parámetro de matriz `status` si hay disponibles en el sistema tanto un búfer frontal como otro trasero. En caso contrario, el valor devuelto es `GL_FALSE`. Para una animación continua, también podemos usar:

```
glutIdleFunc (animationFcn);
```

donde al parámetro `animationFcn` se le puede asignar el nombre de un procedimiento que se encargue de realizar las operaciones de incremento de los parámetros de animación. Este procedimiento se ejecutará de modo continuo cuando no haya sucesos de la ventana de visualización que procesar. Para desactivar la función `glutIdleFunc`, podemos asignar a este argumento el valor `NULL` o el valor `0`.

En el siguiente fragmento de código se proporciona un ejemplo de programa de animación que hace girar de modo continuo un hexágono regular en el plano xy en torno a un eje z . El origen de las coordenadas de pantalla tridimensionales se coloca en el centro de la ventana de visualización, de modo que el eje z pasa a través

de esta posición central. En el procedimiento `init`, utilizamos una lista de visualización para especificar la descripción del hexágono regular, cuya posición central está originalmente en la posición (150, 150) de las coordenadas de pantalla y que tiene un radio (distancia desde el centro del polígono a cualquiera de sus vértices) igual a 100 píxeles. En la función de visualización, `displayHex`, especificamos una rotación inicial de 0° en torno al eje z e invocamos la rutina `glutSwapBuffers`. Para activar la rotación, utilizamos el procedimiento `mouseFcn` que incrementa continuamente el ángulo de rotación en 3° cuando pulsamos el botón central del ratón. El cálculo del ángulo de rotación incrementado se lleva a cabo en el procedimiento `rotateHex`, que es invocado por la rutina `glutIdleFunc` en el procedimiento `mouseFcn`. Detenemos la rotación pulsando el botón derecho del ratón, lo que hace que se invoque `glutIdleFunc` con un argumento `NULL`.

```
#include <GL/glut.h>
#include <math.h>
#include <stdlib.h>

const double TWO_PI = 6.2831853;

GLsizei winWidth = 500, winHeight = 500; // Tamaño inicial ventana visualización.
GLuint regHex; // Definir nombre para lista visualización.
static GLfloat rotTheta = 0.0;

class scrPt {
public:
    GLint x, y;
};

static void init (void)
{
    scrPt hexVertex;
    GLdouble hexTheta;
    GLint k;

    glClearColor (1.0, 1.0, 1.0, 0.0);

    /* Establecer lista de visualización para un hexágono regular de color rojo.
     * Los vértices del hexágono son seis puntos equiespaciados
     * situados sobre una circunferencia. */
    regHex = glGenLists (1);
    glNewList (regHex, GL_COMPILE);
    glColor3f (1.0, 0.0, 0.0);
    glBegin (GL_POLYGON);
        for (k = 0; k < 6; k++) {
            hexTheta = TWO_PI * k / 6;
            hexVertex.x = 150 + 100 * cos (hexTheta);
            hexVertex.y = 150 + 100 * sin (hexTheta);
            glVertex2i (hexVertex.x, hexVertex.y);
        }
    glEnd ();
    glEndList ();
}

void displayHex (void)
{
```

```

    glClear (GL_COLOR_BUFFER_BIT);

    glPushMatrix ( );
    glRotatef (rotTheta, 0.0, 0.0, 1.0);
    glCallList (regHex);
    glPopMatrix ( );

    glutSwapBuffers ( );

    glFlush ( );
}

void rotateHex (void)
{
    rotTheta += 3.0;
    if (rotTheta > 360.0)
        rotTheta -= 360.0;

    glutPostRedisplay ( );
}

void winReshapeFcn (GLint newWidth, GLint newHeight)
{
    glViewport (0, 0, (GLsizei) newWidth, (GLsizei) newHeight);

    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ( );
    gluOrtho2D (-320.0, 320.0, -320.0, 320.0);

    glMatrixMode (GL_MODELVIEW);
    glLoadIdentity ( );

    glClear (GL_COLOR_BUFFER_BIT);
}

void mouseFcn (GLint button, GLint action, GLint x, GLint y)
{
    switch (button) {
        case GLUT_MIDDLE_BUTTON: // Comenzar la rotación.
            if (action == GLUT_DOWN)
                glutIdleFunc (rotateHex);
            break;
        case GLUT_RIGHT_BUTTON: // Detener la rotación.
            if (action == GLUT_DOWN)
                glutIdleFunc (NULL);
            break;
        default:
            break;
    }
}

void main (int argc, char** argv)
{
    glutInit (&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);

```

```

glutInitWindowPosition (150, 150);
glutInitWindowSize (winWidth, winHeight);
glutCreateWindow ("Animation Example");

init ( );
glutDisplayFunc (displayHex);
glutReshapeFunc (winReshapeFcn);
glutMouseFunc (mouseFcn);

glutMainLoop ( );
}

```

13.11 RESUMEN

Podemos construir una secuencia de animación fotograma a fotograma o podemos generarla en tiempo real. Cuando se construyen y almacenan fotogramas independientes de una animación, los fotogramas pueden posteriormente transferirse a una película o mostrarse en una rápida sucesión sobre un monitor de vídeo. Las animaciones que incluyen escenas y movimientos complejos suelen crearse de fotograma en fotograma, mientras que las secuencias de movimiento más simples se muestran en tiempo real.

En un sistema de barrido, pueden usarse métodos de doble búfer para facilitar la visualización del movimiento. Se utiliza un búfer para registrar la pantalla, mientras que se carga en un segundo búfer los valores de los píxeles correspondientes al siguiente fotograma de la secuencia. Después, se intercambian los papeles de los dos búferes, usualmente al final de un ciclo de refresco.

Otro método de barrido para mostrar una animación consiste en realizar secuencias de movimiento utilizando transferencias en bloque de los píxeles. Las traslaciones se llevan a cabo mediante un simple movimiento de un bloque rectangular de píxeles desde una posición de búfer de imagen a otra. Asimismo, las rotaciones en incrementos de 90° pueden llevarse a cabo mediante una combinación de traslaciones e intercambios de filas-columnas dentro de la matriz de píxeles.

Pueden usarse métodos basados en tablas de colores para construir animaciones de barrido simples, almacenando una imagen de un objeto en múltiples ubicaciones del búfer de imagen y utilizando diferentes valores en la tabla de colores. Una imagen se almacena con el color de primer plano y las copias de la imagen en las otras ubicaciones tendrán un color de fondo. Intercambiando rápidamente los valores de color de primer plano y de fondo almacenados en la tabla de colores, podemos mostrar el objeto en varias posiciones de pantalla.

Son varias las etapas de desarrollo necesarias para producir una animación, comenzando con el guión, las definiciones de los objetos y la especificación de los fotogramas clave. El guión es un resumen de la acción, mientras que los fotogramas clave definen los detalles de los movimientos de los objetos para posiciones seleccionadas dentro de una secuencia de animación. Una vez definidos los fotogramas clave, se generan los fotogramas intermedios para conseguir un movimiento suave entre un fotograma clave y el siguiente. Una animación infográfica puede incluir especificaciones de movimiento para la «cámara», además de trayectorias de movimientos para los objetos y personajes que participen en la animación.

Se han desarrollado diversas técnicas para simular y enfatizar los efectos de movimiento. Los efectos de compresión y expansión son métodos estándar para resaltar las aceleraciones, y la modificación del tiempo entre unos fotogramas y otros permite conseguir variaciones de velocidad. Otros métodos incluyen movimientos preliminares de preparación, movimientos de seguimiento al final de la acción y métodos de variación del punto de atención que centran la imagen sobre una acción importante que esté teniendo lugar en la escena. Normalmente, se utilizan funciones trigonométricas para determinar el espaciado temporal de los fotogramas intermedios cuando los movimientos incluyen aceleraciones.

Las animaciones pueden generarse mediante software de propósito especial o utilizando un paquete gráfico de propósito general. Entre los sistemas disponibles para la animación automática por computadora se incluyen los sistemas basados en fotogramas clave, los sistemas parametrizados y los sistemas basados en *scripts*.

Muchas animaciones incluyen efectos de morfismo, en los que se hace que cambie la forma de un objeto. Estos efectos se consiguen utilizando fotogramas intermedios para efectuar la transición, transformando los puntos y líneas que definen un objeto en los puntos y líneas que definen el objeto final.

Los movimientos dentro de una animación pueden describirse por especificación directa del movimiento o pueden estar dirigidos por objetivos. Así, una animación puede definirse en términos de los parámetros de traslación y rotación, o los movimientos pueden describirse mediante ecuaciones o mediante parámetros cinemáticos o dinámicos. Las descripciones cinemáticas del movimiento especifican las posiciones, velocidades y aceleraciones; las descripciones dinámicas del movimiento se proporcionan en términos de las fuerzas que actúan sobre los objetos incluidos en una escena.

A menudo se utilizan figuras articuladas para modelar el movimiento de las personas y de los animales. Con este método, se definen enlaces rígidos dentro de una estructura jerárquica, conectados mediante articulaciones giratorias. Cuando se imprime movimiento a un objeto, cada subparte está programada para moverse de una forma concreta en respuesta al movimiento global.

La velocidad de muestreo para los movimientos periódicos debe producir los suficientes fotogramas por ciclo como para mostrar correctamente la animación. En caso contrario, pueden producirse movimientos erráticos o confusos.

Además de las operaciones de barrido y de los métodos basados en tablas de colores, hay disponibles algunas funciones en GLUT (OpenGL Utility Toolkit) para desarrollar programas de animación. Estas funciones proporcionan rutinas para las operaciones de doble búfer y para incrementar los parámetros de movimiento durante los intervalos de inactividad durante el procesamiento. En la Tabla 13.1 se enumeran las funciones GLUT para generar animaciones con programas OpenGL.

TABLA 13.1. RESUMEN DE FUNCIONES DE ANIMACIÓN OpenGL.

<i>Función</i>	<i>Descripción</i>
<code>glutInitDisplayMode (GLUT DOUBLE)</code>	Activa las operaciones de doble búfer.
<code>glutSwapBuffers</code>	Intercambia los búferes de refresco frontal y trasero.
<code>glGetBooleanv (GL_DOUBLEBUFFER, status)</code>	Consulta al sistema para determinar si están disponibles las operaciones de doble búfer.
<code>glutIdleFunc</code>	Especifica una función para incrementar los parámetros de animación.

REFERENCIAS

Los sistemas de animación por computadora se analizan en Thalmann y Thalmann (1985), Watt and Watt (1992), O'Rourke (1998), Maestri (1999 y 2002), Kerlow (2000), Gooch y Gooch (2001), Parent (2002), Pockock y Rosebush (2002) y Strothotte y Schlechtweg (2002). Las técnicas tradicionales de animación se exploran en Lasseter (1987), Thomas, Johnston y Johnston (1995) y Thomas y Lefkon (1997). Los métodos de morfismos se estudian en Hughes (1992), Kent, Carlson y Parent (1992), Sederberg y Greenwood (1992) y Gomes, Darsa, Costa y Velho (1999).

Hay disponibles diversos algoritmos para aplicaciones de animación en Glassner (1990), Arvo (1991), Kirk (1992), Gascuel (1993), Snyder, Woodbury, Fleischer, Currin y Barr (1993) y Paeth (1995). Para ver una explicación de las técnicas de animación en OpenGL, consulte Woo, Neider, Davis y Shreiner (1999).

EJERCICIOS

- 13.1 Diseñe un guión y los fotogramas clave correspondientes para una animación de una figura articulada simple, como en la Figura 13.19.
- 13.2 Escriba un programa para generar los fotogramas intermedios para los fotogramas clave especificados en el Ejercicio 13.1 utilizando interpolación lineal.
- 13.3 Expanda la secuencia de animación del Ejercicio 13.1 para que incluya dos o más objetos móviles.
- 13.4 Escriba un programa para generar los fotogramas intermedios para los fotogramas clave del Ejercicio 13.3 utilizando interpolación lineal.
- 13.5 Escriba un programa de morfismo para transformar cualquier polígono en otro polígono especificado, utilizando cinco fotogramas intermedios.
- 13.6 Escriba un programa de morfismo para transformar una esfera en un poliedro especificado, utilizando cinco fotogramas intermedios.
- 13.7 Defina una especificación de una animación que incluya aceleraciones y que implemente la Ecuación 13.7.
- 13.8 Defina la especificación de una animación que incluya tanto aceleraciones como deceleraciones, implementando los cálculos de espaciado de los fotogramas intermedios dados en las Ecuaciones 13.7 y 13.8.
- 13.9 Defina la especificación de una animación que implemente los cálculos de aceleración-deceleración de la Ecuación 13.9.
- 13.10 Escriba un programa para simular los movimientos lineales bidimensionales de un círculo relleno dentro de un área rectangular especificada. Hay que dar al círculo una posición y una velocidad iniciales y el círculo debe rebotar en las paredes, siendo el ángulo de reflexión igual al ángulo de incidencia.
- 13.11 Convierta el programa del ejercicio anterior en un juego de frontón, sustituyendo un lado del rectángulo por un lado del rectángulo por un corto segmento de línea que pueda moverse adelante y atrás a lo largo de dicho lado del rectángulo. El movimiento interactivo del segmento de línea simula una raqueta que puede colocarse para evitar que escape la bola. El juego terminará cuando el círculo escape del interior del rectángulo. Los parámetros iniciales de entrada incluyen la posición del círculo, la dirección y la velocidad. La puntuación del juego puede incluir el número de veces que la raqueta golpea a la bola.
- 13.12 Modifique el juego de frontón del ejercicio anterior para variar la velocidad de la bola. Después de un corto intervalo fijo, como por ejemplo cinco rebotes, la velocidad de la bola puede incrementarse.
- 13.13 Modifique el ejemplo de la bola bidimensional dentro de un rectángulo, para convertirlo en una esfera que se mueva tridimensionalmente en el interior de un paralelepípedo. Pueden especificarse parámetros de visualización interactivos para ver el movimiento desde distintas direcciones.
- 13.14 Escriba un programa para implementar la simulación de una bola que rebote utilizando la Ecuación 13.10.
- 13.15 Expanda el programa del ejercicio anterior para incluir efectos de compresión y expansión.
- 13.16 Escriba un programa para implementar el movimiento de una bola que rebote utilizando ecuaciones dinámicas. El movimiento de la bola deberá estar gobernado por una fuerza gravitatoria dirigida hacia abajo y una fuerza de fricción con el plano de tierra. Inicialmente, se proyecta la bola hacia el espacio con un vector de velocidad dado.
- 13.17 Escriba un programa para implementar especificaciones de movimiento dinámico. Especifique una escena con dos objetos o más, con unos parámetros de movimiento iniciales y con unas fuerzas especificadas. Después, genere la animación resolviendo las ecuaciones de fuerza. (Por ejemplo, los objetos podrían ser la Tierra, la Luna y el Sol, con fuerzas gravitatorias atractivas que sean proporcionales a la masa e inversamente proporcionales al cuadrado de la distancia).
- 13.18 Modifique el programa del hexágono giratorio para permitir al usuario seleccionar interactivamente el objeto que hay que girar, a partir de una lista de opciones de menú.
- 13.19 Modifique el programa del hexágono giratorio para que la rotación sea alrededor de una trayectoria elíptica.
- 13.20 Modifique el programa del hexágono giratorio para permitir una variación interactiva de la velocidad de rotación.