

Visualización científica en ordenadores personales

Pedro J. Pascual*, Luis Vázquez **

* Escuela Técnica Superior de Informática, B-308
Universidad Autónoma de Madrid, E-28049 Madrid

** Facultad de Informática
Universidad Complutense de Madrid

Resumen

En los últimos años las prestaciones de los ordenadores personales han aumentado considerablemente. Una de las áreas en que estas mejoras han sido más espectaculares ha sido la de sus capacidades gráficas. Se pueden disponer, a un precio muy razonable, de tarjetas gráficas que pueden competir muy dignamente con estaciones de trabajo especializadas en este campo de hace unos pocos años, que tenían un precio considerablemente mayor. Esto hace que la visualización de resultados científicos, que, en determinados contextos como el de simulación y visualización en tiempo real, puede ser muy exigente tanto en demanda de CPU como de capacidades gráficas, esté cada vez más al alcance de los ordenadores personales. En este informe se repasa la infraestructura tanto a nivel de hardware como de software, como paso previo a la preparación de una línea de trabajo en el campo de visualización de resultados científicos. Asimismo se comenta el uso de robots como alternativa para la visualización cualitativa del comportamiento de sistemas físicos.

Plataformas de hardware

Tradicionalmente, la visualización de datos de altas prestaciones ha estado restringida a unas estaciones de trabajo especializadas en el tratamiento gráfico. De hecho, también es bastante común el que estas estaciones estén conectadas a superordenadores de cálculo que abastecen de resultados a la estación gráfica en tiempo real.

De entre las compañías especializadas en este terreno destaca Silicon Graphics, con una base instalada en el ámbito de estaciones gráficas bastante superior al del resto. Dicha compañía tiene además una buena oferta en el terreno de superordenadores de cálculo paralelo, lo que la convierte en una buena solución global en lo que al cálculo y visualización de resultados científicos se refiere; máxime si tenemos en cuenta que también cuenta con una buena base de software, como veremos más adelante.

Si bien la visualización bidimensional (2D) sigue siendo importante en algunos casos específicos, y es la que domina la literatura científica en cuanto al tipo de gráficas que aparecen en los artículos de investigación, la tendencia es a aprovechar las capacidades de la visualización tridimensional (3D), sobre todo cuando los cálculos se realizan sobre modelos adaptados a la realidad tridimensional. Así, mediante una infraestructura de hardware y software adecuada, llega a ser posible navegar virtualmente a través de los resultados del modelo de cálculo en tiempo real, lo que permite detectar resultados de interés de una manera ágil y cómoda. Esta detección sería bastante tediosa por no decir imposible si nos limitáramos al estudio de los resultados numéricos, e incluso de la multitud de gráficas bidimensionales que se podrían generar para un problema tridimensional en su origen.

No sólo es posible inspeccionar los resultados 3D de una manera más eficiente, sino que ello hace posible el poder variar en tiempo real los parámetros de una simulación, de forma que se puedan corroborar hipótesis acerca del comportamiento del modelo.

Uno de los avances más espectaculares que ha tenido lugar en los últimos años ha sido precisamente la aparición de dispositivos de realidad virtual, desde los más simples como puedan ser las gafas 3D de visión estereoscópica, hasta los más complejos como cascos de realidad virtual o guantes de datos para interaccionar con mundos 3D. El uso de estos entornos hace mucho más fácil e intuitiva la interacción con los modelos sobre los que se realizan los cálculos, facilitando aún más la diagnosis de su validez y la interpretación de sus resultados.

En cuanto a las capacidades de estas estaciones gráficas de altas prestaciones en la actualidad, nos encontramos con tasas de volcado de imágenes del orden de 200 millones de píxeles por segundo (con textura, cálculo de profundidad y suavizado de bordes o *antialiasing*), de dibujo de unos 13 millones de polígonos por segundo, 200 millones de interpolaciones trilineales por segundo, 200 millones de vóxeles por segundo, memorias de vídeo (*frame buffer*) de 80 MB y memoria de texturas de 64 MB. La velocidad del *bus* de comunicación con el subsistema gráfico llega hasta los 2 GB por segundo.

Sin embargo, existen superordenadores especializados en visualización, con hasta 128 procesadores trabajando en paralelo. Con estas plataformas se llega a los 7 Gigapíxeles/seg o 200 millones de polígonos por segundo, con hasta 300 MB de *frame buffer* o 1 GB de memoria de texturas. Si bien la velocidad del *bus* no crece significativamente, si se dispone de varias vías de comunicación entre los procesadores y el subsistema gráfico (*pipes*) trabajando en paralelo.

Por otro lado, los subsistemas gráficos de estas estaciones de trabajo cuenta con unidades de proceso especializadas (GPU) para cálculos propios de visualización fotorrealista tridimensional, como puedan ser transformaciones geométricas o iluminación de superficies.

Si bien los ordenadores personales se encuentran aún bastante lejos en cuanto a sus capacidades gráficas de estas estaciones especializadas, la distancia no es tan grande como la diferencia de precio pudiera hacer pensar. Así, un ordenador personal de última generación dotado de una de las mejores tarjetas gráficas aceleradoras 3D del mercado alcanza los 480 millones de píxeles por segundo (con filtrado y textura), los 15 millones de triángulos por segundo, y 32 MB de memoria para *frame buffer* y texturas. Además, el ancho de banda de comunicación con la CPU del sistema alcanza ya el Gbyte por segundo, gracias al *bus* AGP 4x. Estas tarjetas gráficas ya disponen de una GPU para realizar transformaciones e iluminación de escenas, lo que las acerca aún más si cabe a las estaciones mono procesador de alta gama. Y no sólo eso; ya está previsto que la próxima generación de vídeo consolas tenga aceleración de volcado de superficies curvas, y no sólo de polígonos, así que el disponer de esta capacidad en un PC es cuestión de un corto periodo de tiempo.

Incluso en el campo de la realidad virtual ya se empiezan a hacer tentativas en el mercado de los ordenadores personales, y hay tarjetas con gafas estereoscópicas para dar mayor impresión de realismo a las escenas 3D.

Plataformas de software

En cuanto a la base de software empleado en las estaciones de trabajo gráficas, uno de los pilares más extendidos, sobre todo en el área de visualización fotorrealista, es OpenGL. No es de extrañar este hecho, ya que OpenGL [1] fue lanzado al mercado por Silicon Graphics, que, como ya hemos mencionado, tiene una posición predominante en el área de estaciones gráficas.

OpenGL es una especificación de un API (*Aplicación Programan Interfaces*) para el desarrollo de aplicaciones gráficas, sobre todo orientadas a visualización 3D. Dicha especificación está disponible para todo fabricante de tarjetas o subsistemas gráficos que desee implementar aceleración por hardware de determinados procesos involucrados en la generación de escenas tridimensionales, como pueda ser el mapeo de texturas con corrección de perspectiva, transparencias, etc. El fabricante suministrará un controlador de dispositivo que se colocará entre la aplicación basada en OpenGL y el hardware del subsistema gráfico; este controlador se encargará de que los procesos requeridos por la aplicación que tienen aceleración por hardware sean enviados directamente al subsistema gráfico y se ejecuten allí.

Desde el punto de vista del programador, OpenGL supone un soporte a la realización de programas de visualización tridimensional. Dicho soporte consiste en una serie de funciones o rutinas, así como de los tipos de datos asociados, que simplifican notablemente el desarrollo de este tipo de aplicaciones. OpenGL ofrece un API para, por ejemplo:

- el dibujo de primitivas simples en 3D (líneas, círculos, polígonos, curvas, ...)
- transformaciones de coordenadas, empleadas, tanto para la manipulación y generación de objetos (rotación, escalado, etc.) como para la localización del punto de vista
- coloreado y sombreado de los objetos

- iluminación por fuentes de luz
- mapeado de texturas
- efectos visuales como transparencia y niebla
- curvas y superficies
- etc.

Si bien las imágenes obtenidas no llegan a la calidad que puedan ofrecer técnicas más complejas y costosas computacionalmente como las de *ray tracing* ó *radiosity* (proyección de sombras, objetos reflejados, ...), existen técnicas alternativas, como el *environment mapping*, que permiten simular algunas de las características de estas técnicas, manteniendo el coste de cálculo lo suficientemente bajo como para poder realizar generación de mundos 3D en tiempo real.

La presión de la industria de los vídeo juegos no sólo se ha hecho notar en cuanto al hardware se refiere. En el mundo de los ordenadores personales ha obligado a que Microsoft desarrollara una plataforma software para el desarrollo de este tipo de aplicaciones sobre el sistema operativo Windows, conocida como DirectX [2]. Dicha plataforma permite a las aplicaciones acceder directamente a los elementos de hardware necesarios: dispositivos de entrada como el ratón o el *joystick* (DirectAudio), tarjetas de audio (DirectAudio y DirectMusic), ... y, por supuesto y con mención especial, al subsistema gráfico; esto permite que las aplicaciones se ejecuten con una velocidad notablemente superior a si se usara la capa del sistema operativo para acceder a dichos elementos.

DirectX comenzó en sus primeras versiones suministrando acceso directo al subsistema gráfico 2D, conocido como DirectDraw, dejando a la aplicación todo el trabajo de generación de escenas 3D (lo que se conoce habitualmente como el motor 3D). No obstante, muy pronto se suministró soporte para la visualización 3D, mediante Direct3D, y en la actualidad es ampliamente usado en la industria de los vídeo juegos, compitiendo a veces con OpenGL.

Aunque inicialmente Direct3D era mucho más engorroso en cuanto a la programación se refiere, la situación ha cambiado, y las últimas versiones ofrecen una funcionalidad y comodidad de desarrollo que se acercan a la de OpenGL. De hecho hay un proyecto conjunto entre Silicon Graphics y Microsoft para intentar unificar ambas tecnologías, si bien no existe una certeza absoluta de que llegue a buen fin.

Esto hace que Direct3D sea una alternativa razonable a la hora de desarrollar aplicaciones o entornos de visualización científica en un ordenador personal bajo Windows, sobre todo si se va a orientar el dicho desarrollo a ordenadores con tarjetas gráficas de prestaciones medias. OpenGL mantiene como ventajas el tener más historia y, por tanto, estar más desarrollado en cuanto a funcionalidad, y su portabilidad a otras plataformas de hardware, sobre todo, a las estaciones de altas prestaciones gráficas.

Otro estándar que está muy ligado a Silicon Graphics es VRML (*Virtual Reality Modelling Language*), que está intentando establecerse como el estándar de navegación por mundos tridimensionales mediante los navegadores de Internet. Al igual que HTML es la base para el desarrollo de páginas de hipertexto (cada vez con mayores funcionalidades multimedia y de interacción), VRML es un lenguaje que permite desarrollar mundos 3D, que son visualizados mediante un navegador, y que permiten al usuario moverse por este mundo e interactuar con él.

La funcionalidad básica de VRML es la de la construcción de mundos en base a objetos 3D simples (cubos, esferas, superficies a base de triángulos, etc.) que pueden estar animados y tienen las propiedades habituales de textura, material (características de reflexión y refracción), etc. Las últimas versiones de VRML hacen énfasis también en la interacción con el usuario y el comportamiento de los objetos frente a sucesos.

No sería de extrañar que en un futuro cercano fuera habitual el compartir a través de Internet resultados científicos mediante mundos descritos con VRML, de igual forma que ya es muy común el encontrar artículos científicos en la red en formato HTML.

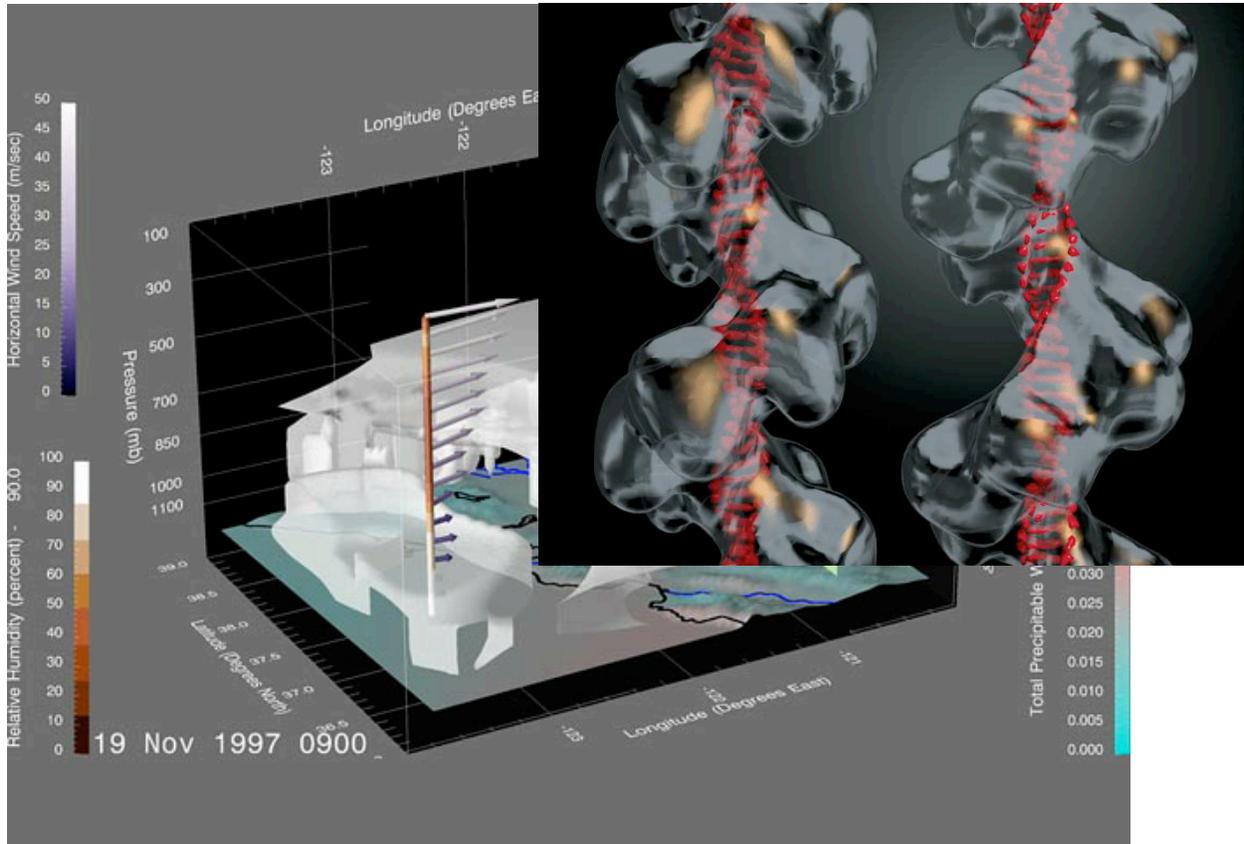


Fig. 1. Datos meteorológicos representados mediante la herramienta OpenDX.

Existen otros estándares en el mundo de las estaciones gráficas, como pueden ser GKS o PHIGS, pero o bien no están tan desarrollados como OpenGL para este tipo de aplicaciones, o bien se han especializado en otras áreas como CAD (*Computer Aided Design*), por lo que apenas han sido migrados al mundo del ordenador personal.

Funcionalidad de programas de visualización estándar actuales

La mayor parte de los resultados de científicos se muestran en la actualidad mediante gráficas en 2D o 3D, de las que existe multitud de variantes: curvas, histogramas, superficies y gráficas 3D, , diagramas de tartas, diagramas polares, toros, etc., todas ellas con multitud de opciones como: distintos tipos de puntos y de líneas, marcas de error, etc.

Este tipo de gráficas es habitual encontrarlo en entornos de trabajo científico, tanto de cálculo simbólico como numérico, como puedan ser Mathematica, Matlab o Maple. A estas herramientas se las ha dotado también de facilidades para poder mostrar la evolución en el tiempo de estas gráficas, para permitir una simulación básica de procesos dinámicos.

Herramientas más especializadas en visualización, como por ejemplo OpenDX [3], sobre todo de procesos de ingeniería en tiempo real, hacen uso de métodos más sofisticados como pueden ser:

- mapas de alturas e isosuperficies
- imágenes en pseudo color
- mapas 3D de campos o vectores
- etc.

Sin embargo, este tipo de visualización de resultados está lejos de los requerimientos de problemas más específicos que suelen caracterizarse por un número elevado de grados de libertad, como pueda ser la mecánica de fluidos, y que normalmente necesitan de desarrollos *ad hoc* para la visualización de resultados de cada problema particular. Estos problemas exigen métodos de visualización mucho más sofisticados, y por ende, unas máquinas más potentes donde llevarlos a cabo, sobre todo porque se desea mostrar la evolución en tiempo real de los cálculos que se realizan en superordenadores.

Si bien este tipo de problemas se ha resuelto tradicionalmente en estaciones gráficas de altas prestaciones, los ordenadores personales están acercándose a las capacidades de dichas estaciones a un ritmo superior al que dichas estaciones están evolucionando. Valga como muestra el ejemplo más común de aplicación de sistemas gráficos especializados: los simuladores de vuelo. Estos aparatos siguen haciendo uso, en el campo profesional, de estaciones gráficas especializadas; pero, si bien el ordenador personal aún se encuentra lejos de los entornos con periféricos sofisticados de estos sistemas (pantallas panorámicas, mandos de interacción realistas, etc.), no lo está tanto en cuanto a la capacidad intrínseca de simulación de las condiciones de vuelo e incluso de la visualización del mundo 3D.

Funcionalidad deseada

Nuestro objetivo es facilitar la interpretación rápida y cómoda de los resultados de simulaciones de sistemas físicos, normalmente modelados por ecuaciones en derivadas parciales o bien por sistemas discretos. Para ello se pretende la realización de aplicaciones de visualización de datos desarrolladas *ad hoc* para cada problema particular, si bien no se descarta el desarrollo de entornos de una cierta generalidad que puedan ser aplicables a una familia de problemas.

Fig. 2. Uso de transparencias en la visualización de una molécula de ADN ([3] OpenDX es la versión de código abierto (open source software version) del producto IBM Open Visualization Data Explorer

Estas herramientas de simulación mostrarán una imagen simplificada del modelo simulado, y en los casos en que sea posible, mostrará su evolución en tiempo real, según se efectúan, los cálculos, de forma que se pueda interpretar los resultados rápidamente y, además, se puedan modificar sobre la marcha los parámetros del modelo y de la simulación numérica.

Así pues, nuestro objetivo es desarrollar herramientas de visualización con las siguientes características más sobresalientes:

- Mostrarán una imagen simplificada del problema bajo estudio mediante objetos simples como esferas, cubos, etc., que, sin embargo, se podrán combinar para construir objetos más complejos
- Los modelos mostrados serán tridimensionales, permitiendo al usuario navegar a través de ellos para centrarse en zonas del área espacial de cálculo que presenten un cierto interés.
- Mostrarán la evolución temporal de estos modelos, siguiendo los resultados de la simulación numérica.
- Se hará uso de técnicas propias de gráficos 3D que sean aceleradas por hardware, como pueden ser el mapeo de texturas, el cambio de las características de los materiales (color, características de reflexión), transparencias, etc.
Mediante estas técnicas podrán, por ejemplo, mostrarse radios de acción o áreas de influencia de objetos del modelo, sin más que crear una niebla coloreada alrededor; indicar si en algún área se ha llegado al estado estacionario, modificando sus características de reflexión; o asignar códigos de pseudo color según el estado en que se encuentra un determinado objeto del modelo.
- Se podrán modificar los parámetros del modelo sobre la marcha para tener así un entorno de simulación interactivo en el que poder detectar rápidamente resultados de interés correspondientes a unos determinados valores de dichos parámetros.
- Se podrán alterar los parámetros de la simulación numérica, como, por ejemplo, los pasos espacial o temporal cuando se requiera una mayor precisión en el cálculo, o realizar un zoom en una zona del espacio o del tiempo; o bien, la alteración de las condiciones de frontera para comprobar su efecto en el resultado de la simulación.

- Las herramientas debieran ser escalables; es decir, deberían poder ejecutarse en un entorno de ordenador personal así como en estaciones gráficas más sofisticadas como las Silicon Graphics de alta gama. De esta forma se podrá realizar un trabajo de estudio previo del problema en ordenadores personales de gama media-alta, mucho más asequibles; cuando se tenga una idea clara de la orientación que se ha de dar al estudio en profundidad del problema, y éste requiera un cálculo intensivo y unas capacidades gráficas elevadas, se migrará a entornos de cálculo y visualización más complejos, y también más difícilmente accesibles.

Visualización no gráfica: uso de robots

Existen alternativas a la visualización de resultados científicos mediante programas gráficos. Una de ellas es el empleo de robots cuyo movimiento o comportamiento esté relacionado con los resultados de un determinado experimento, normalmente numérico.

En la actualidad hay disponibles en el mercado kits de robots relativamente asequibles, como el de la compañía Lego (Mindstorms Robotic Invention System). Si bien estos robots en un principio están diseñados para trabajar con programas que se descargan al propio robot de forma que éste funcione de manera autónoma, también pueden ser controlados desde un ordenador personal mediante una comunicación por infrarrojos. Esto permite que el comportamiento del robot esté controlado por programas más complicados que los que pueden descargarse al robot, cuyo tamaño está bastante limitado. Además, el propio sistema operativo que tiene el robot se puede actualizar y cambiar por otro más orientado a este tipo de usos.

Se pueden realizar así programas escritos en lenguajes estándar como C o Visual Basic, que lleven a cabo simulaciones numéricas más o menos complejas, y cuyos resultados controlen el comportamiento del robot. Para ello también están disponibles las bibliotecas de software de desarrollo (SDK ó Software Development Kit) que simplifican el acceso a los controles del robot.

Además de la unidad de proceso, que ejecuta las órdenes de los programas descargados o las que le comunica un ordenador personal, un robot básico como el de Lego consiste de unos motores que le permiten desplazarse y una serie de sensores: contacto, temperatura, etc. En el funcionamiento más básico, estos motores sólo permiten órdenes de arranque y parada, y graduar su velocidad. Este control es demasiado burdo, por lo que el movimiento resultante no será exacto. Sin embargo, ya hay disponibles motores servo, que permiten un mayor control al poder ordenárseles que giren un número de vueltas (o fracción de vuelta) determinados.

Esto conduce a que este tipo de visualización va a ser cualitativo, ya que no se dispone de un control muy preciso sobre el movimiento del robot, además de requerirse unas tareas previas de calibrado del movimiento, que dependerá de la forma del robot (longitud de ejes de las ruedas, separación de estos ejes, tamaño de la rueda, etc.). Sin embargo, esto, que puede parecer un defecto, puede ser interesante de cara a comparar el comportamiento de distintos sistemas que, en principio, no se cree relacionados. Téngase en cuenta además que el comportamiento del robot no sólo va a depender de los resultados de la simulación sino de la correspondencia que hagamos entre estos resultados y los parámetros que controlan el movimiento del robot: esencialmente, arranque y parada de los motores y magnitud de la velocidad de cada uno. Además de para comparar de forma cualitativa distintos sistemas físicos, el empleo de robots puede también ser útil en experimentos de control de caos, o en la visualización de la dependencia de caos de las condiciones iniciales.



Fig. 3. Uso de un robot para visualización de comportamiento caótico. El robot se mueve de acuerdo al resultado de una ecuación diferencial caótica como el atractor de Chua.

Conclusiones

Tras lo comentado en los apartados anteriores, creemos que es viable el realizar herramientas de visualización de resultados científicos de complejidad media, e incluso alta en ciertos aspectos (uso de transparencias, texturas, niebla, iluminación, etc.) sobre un ordenador personal, enfocado a problemas de complejidad media. No obstante no se descarta tampoco el uso del ordenador personal como vehículo de presentación de resultados de cálculos efectuados en superordenadores conectados a través de una red local, en forma semejante a como trabajan los subsistemas gráficos de alta gama.

Además, estas herramientas se desarrollarán fundamentalmente en OpenGL, de forma que si el ordenador personal muestra sus limitaciones en un momento dado, la migración a un entorno más sofisticado de simulación no será tan dramática. No obstante, no se descarta la exploración de la plataforma DirectX para mejorar el rendimiento de la herramienta en ordenadores personales con tarjetas gráficas de gama media; tampoco se descarta el uso de herramientas específicas de visualización de datos como OpenDX [3].

Por último, también se explorará el uso de dispositivos de realidad virtual, desde los más sencillos, disponibles en ordenadores personales, como las gafas de visión estereoscópica, hasta otros más sofisticados, disponibles en entornos mucho más sofisticados.

Además, tampoco se descarta la exploración del uso de robots para la ilustración cualitativa del comportamiento de sistemas dinámicos complejos, en busca de similitudes de comportamiento o esclarecimiento de conceptos como el control de caos o su dependencia respecto de las condiciones iniciales.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el proyecto POCTI/MAT/40706/2001.

Bibliografía

[1] Se puede encontrar información exhaustiva sobre OpenGL en las páginas web de la compañía Silicon Graphics:
<http://www.sgi.com/software/opengl/>

[2] En las páginas de la Microsoft Development Network se dispone de información sobre Direct3D y DirectX:
<http://msdn.microsoft.com/directx/>

[3] OpenDX es la versión de código abierto (*open source software version*) del producto IBM Open Visualization Data Explorer (<http://www.research.ibm.com/dx>). Se puede encontrar información sobre OpenDX en <http://www.opendx.org>