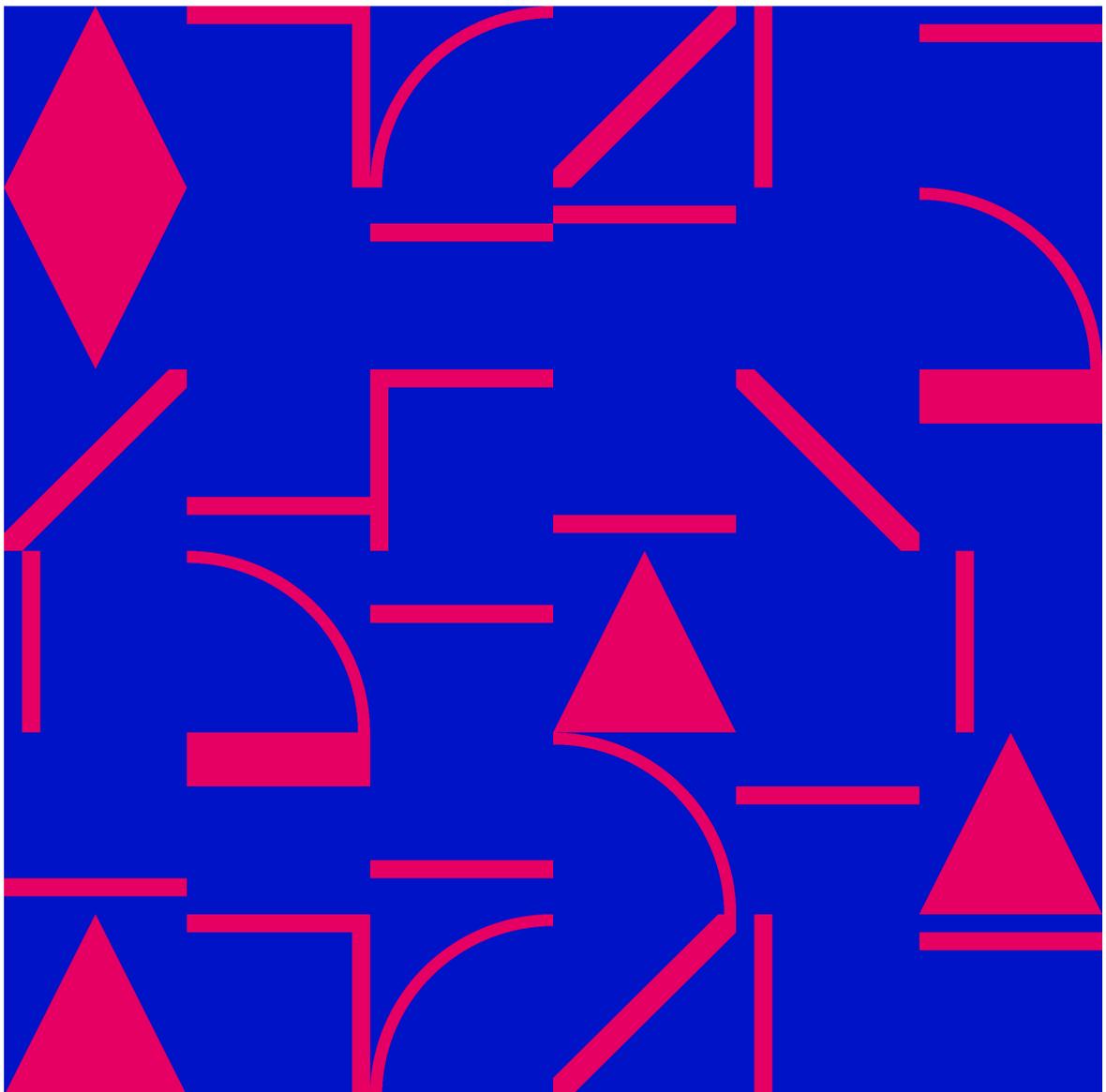

Apuntes de Informática Gráfica

David Escudero-Mancebo



Apuntes de Informática Gráfica

Apuntes de Multimedia

David Escudero-Mancebo
Departamento de Informática
Universidad de Valladolid

ÍNDICE

Introducción	1
Modelado 3D	5
Transformaciones geométricas	13
Transformación de visión	21
Fuentes de luz	31
Sombras	43
Materiales	49
Texturas	61
Rendering	69

Portada generada automáticamente con
<https://github.com/mgiraldo/tenprintcover-py>
Contenidos propiedad del autor
distribuidos con licencia
Creative Commons CC BY-SA.
Tablas e imágenes propiedad de los autores.

Edición 2023

Prólogo

El presente volumen contiene las notas del curso titulado “Informática Gráfica”, ofrecido a los estudiantes del Grado en Ingeniería en Diseño Industrial y Desarrollo de Producto de la Universidad de Valladolid. Los contenidos se corresponden con la parte teórica de la asignatura. La formación se complementa con seminarios prácticos de manejo de un software de modelado y render 3D y con la realización de un proyecto.

Estos apuntes se ponen a disposición de los alumnos, que acceden a ellos empleando la plataforma virtual de la Universidad. Los contenidos se distribuyen por capítulos. Este volumen ha sido editado para ser puesto a disposición del tribunal de la oposición a la que se presentó el autor 2023.

Introducción a un curso de Informática Gráfica

ÍNDICE

I.	Historia de la informática gráfica
II.	Campos de aplicación
III.	Ecosistema de la informática gráfica
IV.	Consideraciones éticas
V.	Visión panorámica
VI.	Ejercicios y cuestiones
	Referencias

2 La informática gráfica es la parte de la informática que se encarga de la representación de gráficas en el ordenador. Se encarga pues de la creación, almacenamiento y visualización de imágenes en el ordenador. Los campos de aplicación son múltiples, y a su descripción dedicaremos gran parte de este tema.

3 Hoy en día la informática gráfica se ha introducido de tal manera en nuestras vidas, que si prescindiéramos de ella, cambiaría la forma de entender medios tan comunes como la televisión, el cine, y la propia informática. Muchos procesos de fabricación dependen en un alto grado del empleo de técnicas de informática gráfica para poder realizarse. La importancia pues de la informática gráfica en la ingeniería y en la sociedad es muy alta debido a la dependencia que hoy por hoy se tiene de la misma.

4 Este tema describe brevemente la historia de las técnicas de Informática Gráfica y detalla los campos de aplicación más importantes de la misma. Al final se hace una visión panorámica de los contenidos del curso.

I. HISTORIA DE LA INFORMÁTICA GRÁFICA

Aunque la carrera de la informática se remonta a los años 40, no fue hasta el año 1961 que comienza a hablarse de informática gráfica. Ivan Sutherland, dentro de su trabajo de tesis doctoral en el MIT, creó un programa llamado *sketchpad*, el cual resultaría decisivo para el desarrollo de las aplicaciones gráficas del ordenador. Este programa permitía dibujar formas simples en el ordenador y almacenarlas posteriormente.

A lo largo de los años 60, los pioneros de la informática gráfica fueron ofreciendo productos como los primeros videojuegos (Russell 1961), simulaciones (Zajac 1963), visión estereoscópica (Sutherland 1966).

En el año 1968, la Universidad de Utah crea el primer laboratorio de investigación en informática gráfica del mundo. En este laboratorio se formaron grandes nombres de la informática gráfica (Jim Clark, fundador de la Silicon Graphics; John Warnock, creador de Postscript y fundador de Adobe) y se realizaron grandes avances científicos como por ejemplo los algoritmos de supresión de superficies ocultas.

Los avances se suceden hasta que en 1973 surge el grupo de interés SIGGRAPH de la asociación ACM (*Association for Computing Machinery*). En Nueva York se crea un laboratorio (el NYIT) que se convierte en puntero a raíz de la incorporación en él de Ed Catmull y Alvy Ray. Sus trabajos se centraron en la creación de herramientas de modelado y animación en 2D y 3D.

En 1979 ocurre un hecho de vital importancia en la evolución de la informática gráfica: George Lucas recluta la personal del NYIT para que experimenten en el laboratorio Lucasfilm. Esto supone la irrupción del mundo privado con vistas claras de sacar altos rendimientos de la informática gráfica.

A finales de los 70 aparecen nuevos avances relativos al realismo de las imágenes sintéticas cuales son el modelo de Cook-Torrance y el trazado de rayos.

A partir de ahí muchos investigadores en todo el mundo han dedicado un esfuerzo considerable la mejora de las técnicas de informática gráfica y a su uso en aplicaciones comerciales. Las compañías han producido hardware cada vez más potente y de menor coste, lo que ha popularizado enormemente el uso de imágenes construidas con la ayuda de ordenadores.

II. CAMPOS DE APLICACIÓN

Las técnicas derivadas de la informática gráfica se utilizan hoy en día en muchas áreas de la industria, los negocios, la educación, el ocio... La lista de aplicaciones es enorme y crece rápidamente. Entre las múltiples aplicaciones en las que la informática gráfica se ha hecho imprescindible están:

Interfaces gráficos de usuario. La mayoría de las aplicaciones que usamos al trabajar en el ordenador usan interfaces de usuario basadas en ventanas (Windows, X-Windows etc...). Los procesadores de texto, las hojas de cálculo, los programas de edición gráfica se favorecen de la utilización de estos entornos gráficos. La informática gráfica juega un papel importante también en la estandarización de las funciones de entrada salida de los datos.

Representaciones gráficas de datos en el mundo de los negocios, y la ciencia y tecnología. Posiblemente sea éste el segundo uso más importante de la informática gráfica; la representación gráfica tanto en 2D como en 3D de datos de funciones matemáticas, físicas y económicas.

Cartografía. La informática gráfica se utiliza para producir representaciones fiables y esquemáticas de espacios naturales y entornos urbanos a partir de datos capturados con sensores. Como ejemplo podemos pensar en los mapas de relieve o de vegetación que se obtienen a partir de datos transmitidos por los satélites.

Medicina. La informática gráfica está jugando un papel cada vez más importante en campos como la diagnosis médica y cirugía. A través de imágenes, el médico diagnostica enfermedades y el cirujano es capaz de realizar intervenciones quirúrgicas con menores riesgos.

Diseño Asistido por Ordenador. En CAD el usuario usa gráficos interactivos para diseñar componentes y sistemas de dispositivos mecánicos, eléctricos y de otros tipos.

Sistemas Multimedia. Como su propio nombre indica, la multimedia implica el uso de más de un medio de comunicación. Por supuesto uno de ellos es el medio visual y es aquí donde la informática gráfica juega un papel primordial.

Simulación. Los simuladores son empleados con frecuencia por el ahorro que suponen en determinadas aplicaciones industriales o militares. En estos sistemas la parte de informática gráfica es fundamental para dar sensación de realidad a la vez que facilitamos su empleo al usuario.

Arte por Computadora. Los métodos de informática gráfica se utilizan de forma generalizada tanto en aplicaciones de bellas artes como en aplicaciones de arte comercial y publicidad. La mayoría de ilustraciones que encontramos en las revistas, carátulas de discos etc... se realizan con paquetes informáticos de **diseño gráfico**.

Entretenimiento. En la actualidad se utilizan comúnmente los métodos de informática gráfica en la producción de cine, videojuegos etc... La industria de los videojuegos se ha convertido en una de las más potentes en el mundo de la informática gráfica y en ella se está desarrollando I+D que supera la simple aplicación de tecnología ya definida.

Educación. Donde se han introducido los métodos que se apoyan en material multimedia e hipertexto para mejorar las actitudes hacia el aprendizaje de los alumnos de edades tempranas. La publicación de contenidos en soporte web es una realidad cada vez más extendida, y estos contenidos se han hecho prácticamente imprescindibles para el profesor de hoy en día.

Tratamiento de Imágenes. Los métodos que se utilizan en la informática gráfica y el **procesamiento digital de imágenes** se traslapan, las dos áreas realizan de forma fundamental, operaciones distintas. En la informática gráfica se utiliza un ordenador para crear una imagen. En el procesamiento de imágenes se aplican técnicas para tratar e interpretar imágenes existentes. Por lo general, la informática gráfica y el procesamiento de imágenes se combinan en muchas aplicaciones.

III. ECOSISTEMA DE LA INFORMÁTICA GRÁFICA

La cresta de la ola tecnológica en informática gráfica, que en un tiempo estuvo en los laboratorios universitarios se ha trasladado progresivamente a las grandes multinacionales. Los ingentes beneficios que se obtienen de este área de trabajo, ha hecho que las inversiones hayan aumentado desde el capital privado; una competencia que muy pocas universidades han podido soportar. Basta ver las actas del SIGGRAF o de otro congreso similar para darse cuenta de que los avances más importantes se hacen hoy en día en laboratorios de I+D de las empresas.

El mundo de la informática en general y el de la informática gráfica en particular han asistido en las últimas décadas a una concentración progresiva de compañías, que, bien sea por fusión, ruina o absorción de unas y de otras, está provocando que el número de empresas dedicadas a unos u otros temas se esté reduciendo progresivamente. El capitalismo somete a las empresas a una presión que les hace competir entre ellas, creándose un ecosistema en el que sobreviven primero los más fuertes

Así, a finales del siglo XX nos encontrábamos con que la capacidad de avance tecnológico en el área de la informática gráfica estaba concentrado en un puñado de multinacionales que podemos enumerar.

Silicon Graphics Que combinando productos hardware y software ha conseguido copar el mercado de la informática gráfica durante dos décadas.

Autodesk Que apoyándose en la herramienta de CAD con mayor éxito en los mercados de Ingeniería y Arquitectura, ofrece herramientas de modelado, animación, e incluso de desarrollo punteras en el mundo PC.

Adobe Que es sin duda la empresa de mayor éxito en el mundo del Diseño Gráfico y Tratamiento de Imagen en entornos Mac.

Corel Competencia de la anterior empresa en el mundo PC.

Macromedia Con productos de excelente calidad en la edición de CD Multimedia y edición de sitios web.

Asimetrix Competencia ferrea del anterior en el mundo de las herramientas de autor.

Micrografx Es una empresa líder en el software de gráficos especializado en el desarrollo de diagramas sobre distintas aplicaciones.

Bentley Que con el producto *Microstation*, es uno de los grandes en el mundo de la ingeniería civil.

Microsoft Que también dispone de herramientas de creación de gráficos sin olvidar que las propias herramientas de *Office* utilizan abundante tecnología propia de la informática gráfica.

Nintendo, Sony, Sega etc..., productoras de videojuegos con gráficos especializados.

En [1] el autor de este mismo artículo afirmaba lo siguiente *Hay muy pocas otras empresas que puedan mencionarse en esta lista. No es sólo que tengan un volumen de ventas menor, sino que no pueden competir con las anteriores y la naturaleza del mercado capitalista hará que sean absorbidas o que desaparezcan tarde o temprano.* Me equivoqué por completo. De hecho el número de empresas nuevas desde que

se escribieron los párrafos anteriores se ha multiplicado. De hecho lo que ha ocurrido es que los programas informáticos de mayor éxito han crecido tanto que su mantenimiento es muy caro. Esto ha propiciado que surjan empresas más pequeñas que ofrecen productos de menor volumen que no sólo son más fáciles de mantener, sino que pueden ser más económicos. Para los usuarios también son ventajosos porque, al reducir la funcionalidad son más fáciles de manejar. Volviendo a la metáfora del ecosistema, si es una primera etapa sobreviven los más fuertes, el tiempo demuestra que, como en biología, no es el tamaño sino la capacidad de adaptación lo que nos hace sobrevivir en los ambientes competitivos.

Este fenómeno se ha visto favorecido por la aparición de productos de software libre que permiten entrar en el mercado de forma rápida. Partiendo de código libre ya desarrollado es más fácil conseguir un producto final que si hay que programarlo desde cero. En gráficos 2D, el producto de referencia en Inkscape, en gráficos 3D el producto en Blender y en procesamiento de imagen Gimp. Estos tres productos son el punto de partida de muchos productos software de éxito en el mundo de la informática gráfica.

IV. CONSIDERACIONES ÉTICAS

El uso de cualquier tecnología tiene un impacto en el entorno social y una capacidad transformadora. Después de haber superado un curso de informática gráfica, el alumno estará en disposición de diseñar aplicaciones y crear grafismos que puedan influir en la creación de un antes y un después en la vida de los usuarios. Como profesionales debemos reflexionar sobre dicho impacto y las consecuencias que se pueden derivar de la aplicación de las tecnologías con las que trabajamos.

Un caso que podemos entender fácilmente es el de los falsos escenarios virtuales creados en publicidad. Es relativamente sencillo construir infografías que hagan que los productos de consumo sean más atractivos de lo que realmente son. En algunos casos extremos podría llegarse a situaciones delictivas de estafa, pero es habitual la falta de normativa, y en esa situación la ética debe poner límites.

Otras situaciones son más sutiles pero no son menos problemáticas. Por ejemplo podemos pensar en la capacidad que tienen los gráficos de atrapar a los usuarios, creando situaciones patológicas entre determinados adolescentes enganchados a los videojuegos y otras aplicaciones.

Tener en cuenta estas consideraciones supone acudir a la ética para poder decidir y juzgar situaciones y productos que pueden impactar en el entorno social. La aplicación de la tecnología en un ámbito concreto no tiene como único fin ganar dinero; es importante tener en cuenta que puede mejorar la calidad de vida de las personas. También es el caso de la informática gráfica, que en muchas aplicaciones está sirviendo para conseguir avances muy importantes en el estado de bienestar.

V. VISIÓN PANORÁMICA

Comentamos brevemente aquí a qué se refieren cada una de las partes en las que se dividen estos apuntes. De este modo, se podrá apreciar la cohesión que mantienen entre sí cada

uno de los temas que las componen. El lector podrá dirigirse a aquellas partes que considere de su interés a partir de la siguiente descripción.

Parte I En este bloque se hacen ver las limitaciones de los dispositivos digitales para la representación de gráficos; este bloque profundiza en el estudio de las estructuras de datos y algoritmos que internamente utilizan los paquetes de modelado 3D; se recuerdan conceptos de geometría y de álgebra lineal imprescindibles para los temas posteriores. Se finaliza viendo como se soluciona el problema de la proyección 3D-2D en el ámbito de la Informática Gráfica. Componen esta parte los siguientes temas: Modelado 3D, Transformaciones geométricas y Visión 3D

Parte II En esta parte se abordan los problemas clásicos relacionados con la iluminación de la escena, incluyendo el modelado de fuentes y la generación y tipos de sombras. Forman esta parte los temas de Fuentes de iluminación y de Generación de sombras.

Parte III En este bloque se abordan temas relacionados con el sombreado de superficies, tanto para simular materiales sin aristas como para llenarlos de realismo con texturas. Componen esta parte los siguientes temas de Sombreado de superficies y de Texturas

Parte IV Finaliza el curso con una revisión de las técnicas de rendering que permiten convertir los modelos 3D en imágenes sintéticas. Esta parte contiene un único tema, el tema de Rendering.

MÁS INFORMACIÓN

Todos los libros referentes de la informática gráfica como por ejemplo [2], [3], [4] tienen un capítulo 1 dedicado a tratar cuestiones introductorias como las abordadas en este tema.

Sobre consideraciones éticas de la informática gráfica en [5] encontramos una revisión de perspectivas éticas que deberían abordarse en cursos de informática gráfica.

VI. EJERCICIOS Y CUESTIONES

1. Elabore una definición propia de informática gráfica.
2. Indique las implicaciones de la informática gráfica en el diseño industrial.
3. ¿En qué se diferencian los términos informática gráfica y diseño gráfico?
4. ¿En qué se diferencian los términos informática gráfica y diseño asistido por ordenador?
5. ¿En qué se diferencian el diseño gráfico y el diseño asistido por ordenador?
6. Encuentre en internet el nombre del investigador más importante en informática gráfica de todos los tiempos.
7. Haga un cronograma en el que aparezca el año de nacimiento y muerte de diversas empresas de informática gráfica.
8. Haga un cronograma en el que aparece el año de lanzamiento de diferentes productos software relacionados con informática gráfica, por ejemplo Flash o 3D Studio.
9. Haga un ensayo sobre el papel de las empresas en la investigación en informática gráfica.

10. Haga un ensayo sobre el peligro de la informática gráfica a la hora de crear imágenes fotorrealista que puedan manipular la opinión pública.
11. Localice vídeos en Internet que presenten la evolución histórica de los videojuegos. El objetivo es reflexionar sobre el cambio tecnológico y el protagonismo de los gráficos generador por ordenador en dicho cambio.
12. Busque infografías de edificios o de planes urbanísticos presentados en prensa o en publicidad por parte de promotoras o de ayuntamientos. Después busque el lugar final en el que se realizó la obra y contraste el resultado final con el resultado prometido en la infografía. En algunos casos puede haber importantes diferencias. ¿Cree que podrían existir implicaciones legales motivadas por dichas diferencias?
13. El *Deep fake* es una tecnología que se está haciendo popular porque, entre otras cosas, ofrece la posibilidad de que cualquiera pueda ser el protagonista de una película dada cambiando el rostro del personaje principal por el del cliente. Existe un debate sobre la posible legislación de dicha tecnología. Enumere pros y contras de legislar dicha tecnología.
14. Cada vez es más frecuente que al comprar un piso de nueva construcción las inmobiliarias nos enseñen un render (una imagen virtual) que representa el acabado final. En 2019 un juzgado madrileño condenó a una promotora porque la imagen que presentaron distaba mucho del resultado final. Existe un debate sobre la posible legislación del uso de la simulación 3D en marketing. Enumere dos pros y dos contras de legislar dicha tecnología.

REFERENCIAS

- [1] D. E. Mancebo, *Fundamentos de informática gráfica*. Cano Pina, 2003.
- [2] S. Marschner and P. Shirley, *Fundamentals of computer graphics*. CRC Press, 2015.
- [3] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [4] D. Hearn, M. P. Baker et al., *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall., 2004.
- [5] M. W. Bannatyne, D. I. Troitsky, and R. A. Hall, "Addressing ethical issues in computer graphics curricula," in *Proceedings Fifth International Conference on Information Visualisation*. IEEE, 2001, pp. 263–268.

Técnicas de modelado de objetos 3D

ÍNDICE

I.	Representación de mallas 3D
II.	Instanciación de primitivas
III.	Representaciones de frontera
IV.	Sólidos de recorrido
V.	Superficies paramétricas
VI.	Geometría sólida constructiva
VII.	Partición espacial
VIII.	Sistemas fractales
IX.	Metaballs
X.	Escultura digital
	Referencias

2 La representación final de un objeto 3D en informática gráfica
 3 es una malla de puntos en un espacio geométrico. La repre-
 3 sentación de un simple objeto puede suponer utilizar miles
 4 de puntos con sus correspondientes aristas y poligonales. Para
 4 hacer más ágil la definición de los objetos se emplean las
 4 técnicas de modelado. En este artículo describimos una serie
 4 de técnicas comunes empleadas en modelado de objetos en
 4 informática gráfica que facilitan la definición de los mismos
 4 sin tener que especificar explícitamente cada nodo de las
 4 geometrías finales.

4 A la hora de representar objetos pueden darse dos situacio-
 4 nes: que se trate de un objeto existente o no. En el primer caso,
 5 podremos disponer de una descripción del mismo considerado
 5 el conjunto de infinitos puntos en un espacio tridimensional.
 5 Este enfoque no es viable porque trabajamos con sistemas de
 5 almacenamiento finito. A menudo lo único que hacemos es
 6 aproximar dicho objeto mediante la combinación de objetos
 6 más simples como planos, esferas etc En el segundo caso,
 7 cuando el objeto no existe, el usuario ha de emprender el
 7 proceso que llamamos modelado. Se requiere entonces de
 8 métodos y técnicas que permitan representar lo más fielmente
 8 posible el objeto que sólo existe en nuestra imaginación. Al
 8 conjunto de técnicas y métodos empleados para definir objetos
 8 en 3D se le llama **técnicas de modelado**.

En este capítulo se estudia el tema del modelado de objetos 3D. Primero haremos una revisión sobre las representaciones internas de los sólidos 3D. Estudiaremos las técnicas de modelado conocidas como: geometría sólida constructiva, sólidos de recorrido, fractales, *metaballs* y las técnicas de escultura digital.

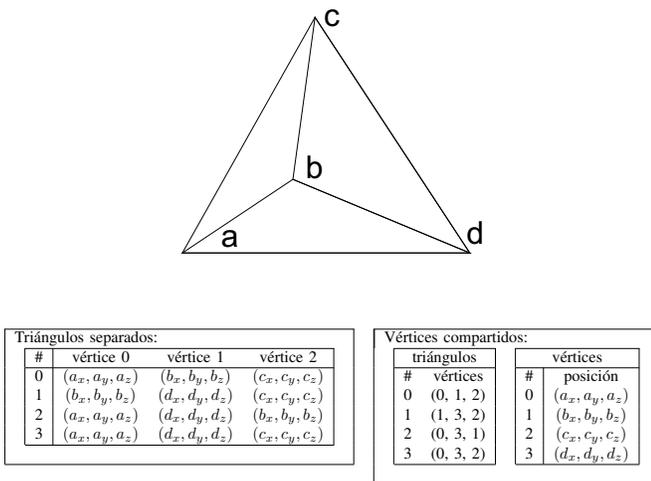


Figura 1: Objeto 3D representado con poligonales. A la izquierda la representación explícita de los polígonos. A la derecha la representación que emplea una lista de vértices.

I. REPRESENTACIÓN DE MALLAS 3D

Un objeto 3D se representa empleando una malla de poligonales que no es más que una colección de aristas, vértices y polígonos conectados de manera coherente, en el sentido que:

- Cada arista conecta dos vértices y cada polígono está cerrado y se define por una secuencia de vértices.
- Cada vértice es compartido por al menos dos aristas y cada arista es parte de, al menos, un polígono.

Los programas software de modelado 3D se deben encargar de garantizar que se cumplan estos requisitos. Cada aplicación emplea su propia representación interna para estos datos. Es frecuente que las aplicaciones empleen una representación para el almacenamiento, otra para uso interno y otra más para interactuar con el usuario. Estas representaciones no tienen por qué coincidir.

A la hora de elegir una representación para los datos es importante tener en cuenta, aparte de velocidad y espacio de almacenamiento, criterios de eficiencia que estarán en función de las operaciones que vayamos a hacer con ellos. A continuación se describen tres representaciones internas distintas para una malla de poligonales, observando las ventajas e inconvenientes de cada una de ellas. A modo de ilustración, veremos a continuación tres tipos de representaciones: la representación explícita, la de punteros a listas de vértices, y la punteros a listas de aristas.

En la **representación explícita** cada polígono se representa como una lista que contiene la lista de coordenadas de los vértices. Una poligonal P , quedará definida como una lista de vértices: $P = [(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)]$. Hay aristas entre los vértices consecutivos y entre el primero y el último. Si necesitamos varios polígonos el método es poco eficiente, ya que muchos puntos van a ser comunes y los repetimos varias veces. Además no hay una representación explícita de las aristas y vértices compartidas. Encontrar el vértice común a una serie de caras es una operación típica,

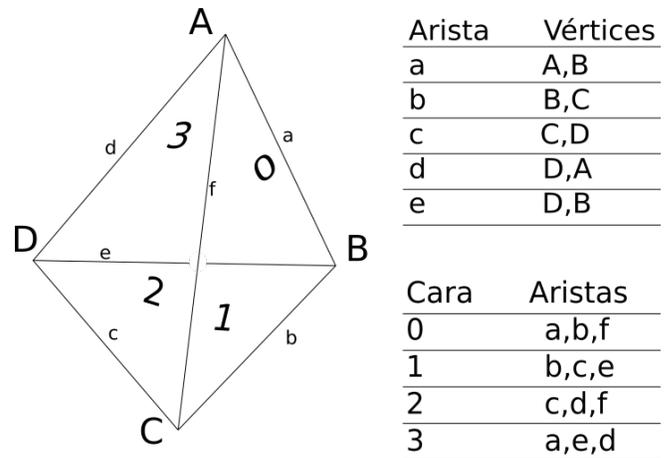


Figura 2: Tabla de poligonales con información de las aristas.

y con este método hemos de buscar en cada uno de los polígonos. La figura 1 muestra una tabla de índices.

Como alternativa surge la **lista de vértices de polígonos**, que almacena cada vértice sólo una vez. Se dispone de una lista de vértices $V = [\bar{v}_1 = (x_1, y_1, z_1), \bar{v}_2 = (x_2, y_2, z_2), \dots, \bar{v}_n = (x_n, y_n, z_n)]$. Un polígono se define especificando una lista de índices que hacen referencia a los vértices del mismo. Un polígono P se representa como $P = (\bar{v}_1, \bar{v}_2, \dots, \bar{v}_n)$, o simplemente $P = (1, 2, \dots, n)$. De este modo, las coordenadas de los vértices se almacenan solamente una vez reduciendo el espacio de almacenamiento. Además, resulta muy sencillo modificar un vértice del objeto. La figura 1 muestra cómo en la figura tabla de la derecha es más compacta y reduce el tamaño de almacenamiento. Sin embargo, aún es difícil encontrar polígonos que compartan arista, y las aristas compartidas son dibujadas dos veces.

Para solventar estos inconvenientes empleamos la **lista de aristas**. Empleamos la lista de vértices V pero representamos los polígonos como un conjunto de índices a la lista de aristas. Cada arista se representa a su vez con los vértices que la determinan y los polígonos a los que pertenece. En la figura 2 se muestra una representación de un sólido basada en la lista de arista. La figura ilustra como el objeto queda representado por un conjunto de vértices, que definen un conjunto de aristas y que definen a su vez las caras de dicho objeto. En este caso, se ha incluido información redundante que facilita el acceso a diferentes componentes como son las caras colindantes y las aristas correlativas.

El triángulo suele ser el polígono base constructivo de las mallas de poligonales. Las **listas de triángulos** determinan las superficies que conforman el sólido. Se emplean triángulos porque es el polígono mínimo elemental: cualquier otro polígono puede construirse en base a triángulos. Las tarjetas gráficas suelen estar preparadas para renderizar triángulos.

Una representación eficaz de objetos en un espacio tridimensional consideraría que un objeto 3D se compone por un conjunto de poligonales que son a su vez un conjunto de tablas. Cada una de estas tablas contiene la representación de un conjunto de vértices, de aristas y de polígonos, generalmente triángulos.

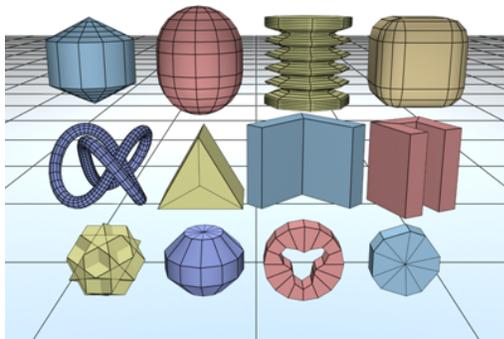


Figura 3: Ejemplos de primitivas básicas (imagen propiedad de Autodesk 3DS).

Para agilizar el proceso de rendering además puede recurrirse a representaciones alternativas. En la figura 2 se muestra una representación en la que se guarda registro de los polígonos que lindan con cada arista. Es frecuente encontrar representaciones en las que se guarda también el **vector normal** a cada uno de los triángulos.

Independientemente de la información guardada en la descripción de la malla de poligonales, el modelado de objetos 3D sería prácticamente inabordable si pretendemos definir los objetos especificando vértice a vértice y arista a arista cada uno de los polígonos que componen el objeto. Para facilitar el trabajo se recurre a las técnicas de modelado que se detallan a continuación: instanciación de primitivas, representaciones frontera, geometría sólida constructiva. . .

II. INSTANCIACIÓN DE PRIMITIVAS

Los programas específicos de modelado 3D definen un conjunto de primitivas, más o menos sofisticadas, que son empleadas para construir los objetos y las escenas. Los objetos que son relevantes para la aplicación de modelado, tienen asociada una primitiva cuyos argumentos son los parámetros de dichos objetos.

Se establecen una serie de primitivas, cada una de ellas relacionada con un tipo de objeto 3D. Cada primitiva contiene un identificador que hace referencia al tipo de objeto y los parámetros. Los parámetros condicionan la forma y el tamaño del objeto y dependen del objeto en sí. Ejemplos de primitivas son:

```
esfera(posición, radio)
cilindro(posición, radio, altura)
```

En estos ejemplos *esfera* y *cilindro* son nombres de primitivas y *posición*, *radio* y *altura* son nombres de parámetros. Las primitivas pueden ser más sofisticadas dependiendo del ámbito de aplicación del software. En la figura 3 se muestran algunas de las primitivas típicas de un programa básico de modelado. Un paquete de CAD puede incluir primitivas para definir elementos mecánicos como pueden ser engranajes o tuercas. Un programa como el 3D Studio incluye primitivas para incluir formas vegetales (árboles, plantas...) y elementos

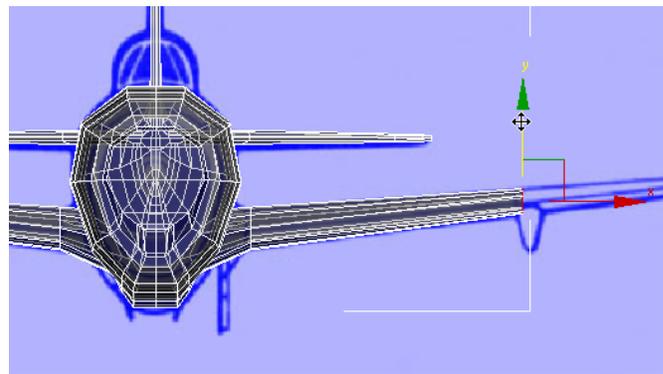


Figura 4: Ejemplos de modelado empleando representaciones frontera (imagen propiedad de Autodesk 3DS).

arquitectónicos (puertas, ventanas, escaleras...). Los parámetros de cada primitiva determinarán la forma y tamaño de dicho elemento.

Un objeto más complejo, cuya forma no se ajusta a la de ninguna de las primitivas elementales, puede ser elaborado mediante la combinación de estas primitivas, y la aplicación de transformaciones geométricas de rotación, traslación y escalado.

Para crear un nuevo tipo de objeto o una nueva primitiva, la única solución posible es modificar el código del software de modelado.

III. REPRESENTACIONES DE FRONTERA

En las representaciones de frontera o *b-reps* los objetos se modelan definiendo las superficies frontera: vértices, caras, aristas. Las caras pueden ser, en principio, cualquier polígono, aunque se suele restringir a polígonos convexos principalmente triángulos. En sistemas de gran capacidad se permiten superficies polinómicas, aunque se complica mucho por la complejidad de cálculos en los cortes entre superficies. Normalmente las superficies curvas se aproximan mediante polígonos.

Para hacer que los cálculos sean más sencillos, se suele permitir sólo aquellos casos en los que cada arista pertenece a dos y sólo dos polígonos. Para que dicha propiedad se cumpla, es necesario que se satisfaga la relación de Euler: $V - A + C = 2$, con V es el número de vértices, A es el número de aristas y C es el número de caras.

Este tipo de representaciones con caras poligonales predominan en los sistemas de modelado 3D. Los métodos de sombreado sirven para producir sensación de suavizado cuando se aproximan las curvas por polígonos.

En 3D Studio, por ejemplo, esta técnica de modelado se llama *poly-modeling*. No es que se especifiquen manualmente los vértices de cada polígono. En realidad se parte de un polígono sencillo como puede ser un plano y se van aplicando sobre él operaciones simples de división y extrusión para acercarse al volumen objetivo. Las operaciones a las que se someten los polígonos o sus artistas se hacen utilizando, generalmente, imágenes del modelo que se quiere modelar que sirven de referencia. Aunque el resultado no sean volúmenes

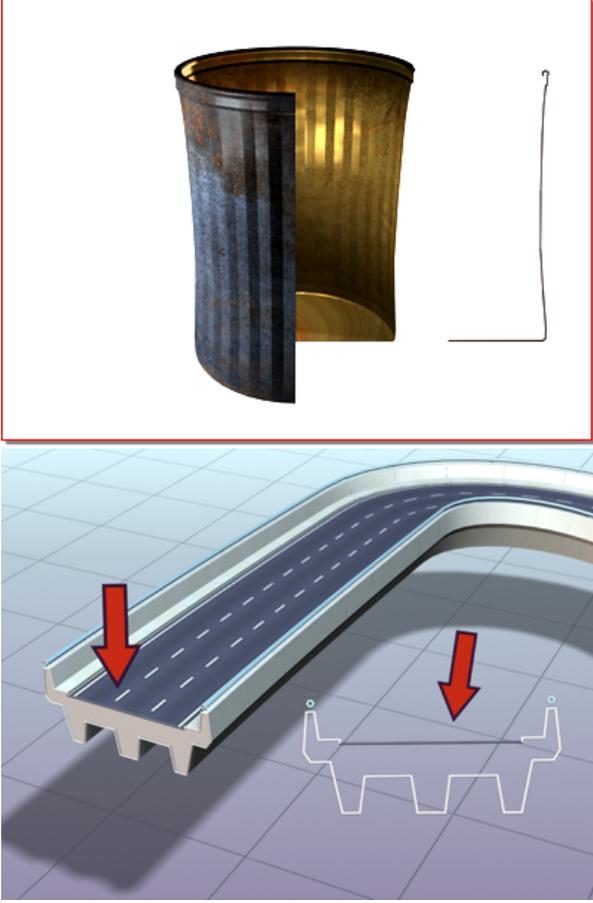


Figura 5: Ejemplos de imágenes construidas con solevados y recorridos de formas 2D (imágenes propiedad de Autodesk 3DS).

de precisión, sí suelen ser de suficiente calidad como para ser empleado en aplicaciones de entretenimiento y de publicidad donde la precisión no sea un condicionante.

IV. SÓLIDOS DE RECORRIDO

En los sólidos de recorrido se define una forma en 2D y se especifica una trayectoria a lo largo del cual se desplaza dicha forma para conformar el volumen 3D. La zona del espacio que recorre la forma inicial es el volumen del nuevo sólido. Cuando el recorrido se realiza a través de una simple línea, hablamos de un sólido de traslación o **extrusión**. Por otro lado, los **sólidos de rotación** se definen haciendo rotar la forma 3D en torno a un eje dado.

Esta posibilidad de construcción de sólidos 3D es ofrecida con frecuencia en los paquetes de modelado sólido. Resulta útil para definir objetos que presentan algún tipo de simetría. La figura 5 ilustra el funcionamiento del método para el caso de la traslación. A partir de una forma simple como es un rectángulo y de una dirección dada, generamos un prisma.

Aunque es una forma de crear objetos muy intuitiva, tiene el inconveniente de que si el sólido de recorrido no se convierte en otro tipo de representación, las formas son muy difíciles de tratar computacionalmente. Normalmente, una vez definido

el volumen, se generan representaciones de frontera antes de realizar la representación final. Las representaciones frontera que se generan se conocen como la piel del sólido de recorrido y se parametrizan por el número de pasos a lo largo del recorrido y por el número de divisiones de la forma 2D. El número de pasos y el número de divisiones de la forma 2D determinan el número de puntos de la malla 3D final.

V. SUPERFICIES PARAMÉTRICAS

Estas superficies son una generalización de las curvas de cúbicas. La expresión genérica de las curvas cúbicas es $Q(t) = G \cdot M \cdot T$, donde G es la matriz de restricciones. Por conveniencia de notación emplearemos s en lugar de t , y tenemos $Q(s) = G \cdot M \cdot S$. Si hacemos que G varíe a lo largo de un camino tridimensional, tenemos

$$Q(s, t) = (G_1(t) \ G_2(t) \ G_3(t) \ G_4(t)) \cdot M \cdot S. \quad (1)$$

Para un t_1 fijo, $Q(s, t_1)$ es una curva porque $G(t_1)$ es constante. Variando t entre 0 y 1, tengo una familia de curvas que componen una superficie. Si $G_i(t)$ son cúbicas, la superficie obtenida se denomina **superficie cúbica paramétrica**.

Como $G_i(t)$ son cúbicas, cada una puede representarse como $G_i(t) = \mathbf{G}_i \cdot M \cdot T$, donde $\mathbf{G}_i = [g_{i1} \ g_{i2} \ g_{i3} \ g_{i4}]$, y g_{ij} es la j -ésima restricción de la curva $G_i(t)$. Como $G_i(t) = G_i^T(t) = T^T \cdot M^T \cdot \mathbf{G}_i^T$, que sustituyendo en la ecuación 1 para cada uno de los cuatro puntos tendré

$$Q(s, t) = T^T \cdot M^T \cdot \begin{pmatrix} \mathbf{g}_{11} & \mathbf{g}_{21} & \mathbf{g}_{31} & \mathbf{g}_{41} \\ \mathbf{g}_{12} & \mathbf{g}_{22} & \mathbf{g}_{32} & \mathbf{g}_{42} \\ \mathbf{g}_{13} & \mathbf{g}_{23} & \mathbf{g}_{33} & \mathbf{g}_{43} \\ \mathbf{g}_{14} & \mathbf{g}_{24} & \mathbf{g}_{34} & \mathbf{g}_{44} \end{pmatrix} \cdot M \cdot S, \quad (2)$$

o

$$Q(s, t) = T^T \cdot M^T \cdot \mathbf{G} \cdot M \cdot S, \quad 0 \leq s, \quad t \leq 1 \quad (3)$$

Que podemos escribir de forma separada para cada componente

$$x(s, t) = T^T \cdot M^T \cdot \mathbf{G}_x \cdot M \cdot S, \quad 0 \leq s, t \leq 1 \quad (4)$$

$$y(s, t) = T^T \cdot M^T \cdot \mathbf{G}_y \cdot M \cdot S, \quad 0 \leq s, t \leq 1 \quad (5)$$

$$z(s, t) = T^T \cdot M^T \cdot \mathbf{G}_z \cdot M \cdot S, \quad 0 \leq s, t \leq 1 \quad (6)$$

Para el caso de las **superficies de Bézier** por ejemplo, debemos sustituir M por M_B . (ver figura 6). El modelo más empleado de este tipo de curvas son los NURBS (*B-splines racionales no uniformes*).

VI. GEOMETRÍA SÓLIDA CONSTRUCTIVA

En la geometría sólida constructiva (*constructive solid geometry* CSG), las primitivas de representación de objetos simples se combinan mediante operaciones booleanas u operaciones de conjuntos que se incluyen directamente en la representación. A partir de unos objetos, creamos nuevos objetos al aplicar sobre ellos operaciones de unión, intersección o diferencia de volúmenes (ver figura 7).

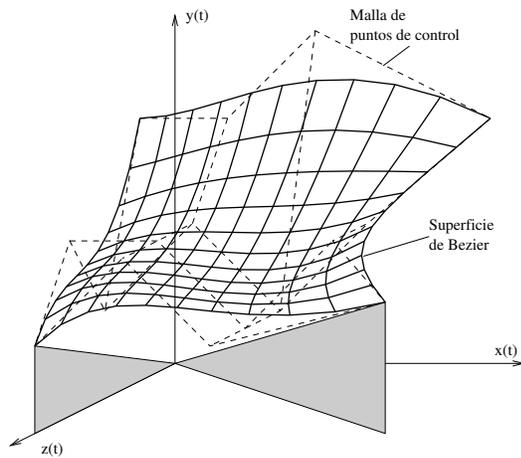


Figura 6: Ejemplo de construcción de una superficie de Paramétrica.

Los objetos se almacenan como árboles donde las ramas son los objetos primitivos y los nodos internos son las operaciones. Algunos nodos son operaciones booleanas, otros son traslaciones, rotaciones y escalados. La figura 7 muestra este proceso. Para tratar los objetos se recorren las ramas primero en profundidad combinando los objetos menores para determinar las propiedades del objeto resultante. La operación de combinación es complicada por tener que considerar intersecciones entre multitud de planos.

VII. PARTICIÓN ESPACIAL

Mediante este método de representación, un sólido es descompuesto en una colección de sólidos menores, que no intersecan, más primitivos que el sólido original, aunque no necesariamente del mismo tipo que el sólido original.

Las imágenes digitales son a las imágenes vectoriales 2D como los sólidos de enumeración de ocupación espacial son a los modelos geométricos de volúmenes 3D. Las celdas se ordenan en un grid de celdas idénticas y ordenadas. Las celdas se denominan **voxels**. Cuando se representa un sólido mediante la técnica de enumeración de ocupación espacial, se controla la presencia o ausencia de voxel en cada posición del espacio. El voxel puede tener asociadas propiedades relativas al material del sólido que lo ocupa como el color, transparencia, textura, etc....

Este método tiene la ventaja de que las operaciones de eliminación de superficies ocultas se simplifican. También es sencillo realizar operaciones booleanas con estos sólidos. Como inconveniente, los sólidos sólo son aproximaciones. El problema es el mismo que el que aparece al rasterizar una primitiva en 2D. Los voxels deben ser pequeños para mejorar la resolución, pero así, el tamaño de la escena crece grandemente. Si se trata de una escena con N voxels de lado, y cada voxel ocupa 1 *byte*, tendremos que almacenar $N \times N \times N$ bytes.

VIII. SISTEMAS FRACTALES

Este término fue acuñado por el matemático francés Benoit Mandelbrot para describir determinados fenómenos de la na-

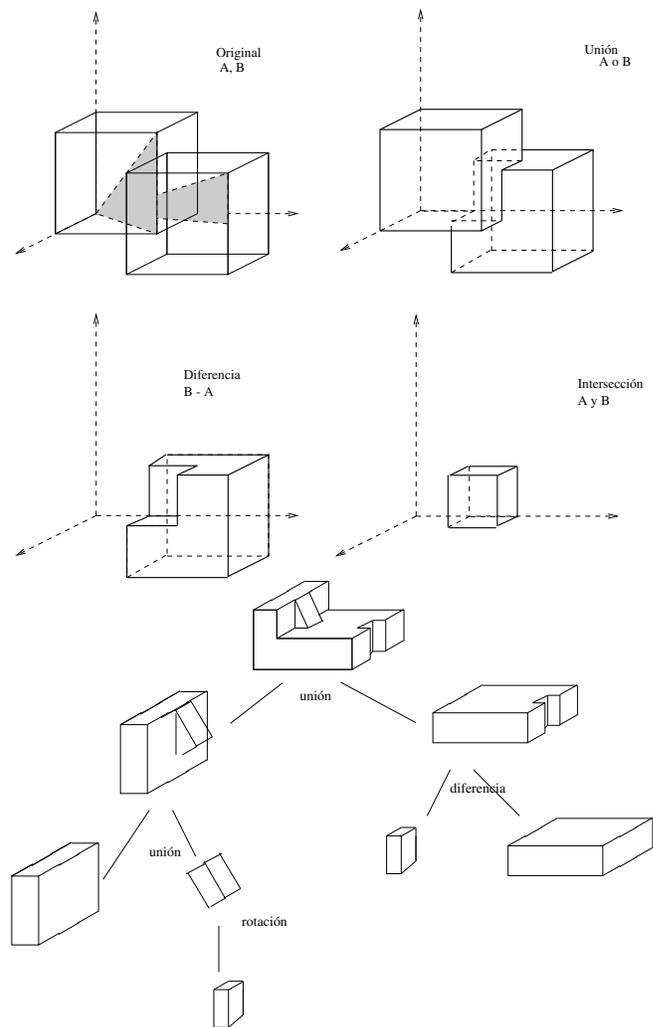


Figura 7: Arriba, ejemplos de figuras realizadas empleando geometría sólida constructiva. Abajo, las figuras se componen al recorrer el árbol y aplicar las operaciones oportunas.

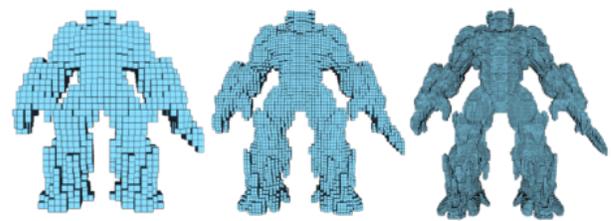


Figura 8: Representación de objetos empleando véxeles a diferentes resoluciones (imagen propiedad de Autodesk Maya).

turalidad, como por ejemplo las líneas costeras. Dichas líneas costeras presentan características similares para cualquier nivel de profundidad que se considere. Así, si consideramos una vista a nivel de piedra, a nivel microscópico o de la costa, la vista es muy similar.

La geometría fractal proporciona una descripción para determinados aspectos de la naturaleza y su tendencia a la **auto-**



Figura 9: Ejemplo de representaciones fractales de formas vegetales. A la izquierda y centro un diente de león. En la figura central se han aplicado factores aleatorios para la longitud de las ramas y los ángulos de los abanicos. Un principio similar se ha utilizado para modelar la hoja de helecho de la izquierda.

similitud. Esto se ilustra en la figura 9. La figura es una representación de un árbol n -ario hasta un cierto nivel de profundidad. Incrementando la profundidad, e introduciendo factores aleatorios en los atributos del dibujo (tamaño de la rama y ángulo de las ramas) se consiguen aspectos próximos a algunas formas encontradas en la naturaleza.

En informática gráfica, las técnicas de geometría fractal han sido empleadas para generar modelos del terreno. Para ello, se subdividen las caras de objetos simples de forma recursiva hasta un determinado nivel de detalle. La subdivisión implica la elección del punto medio entre dos vértices y aplicando una perturbación a lo largo de una línea normal en la arista. El resultado es la subdivisión de la cara original en un conjunto de caras de menor tamaño, cada una con una orientación aleatoria en torno a la orientación de la cara original.

Los métodos de subdivisión se basan en el algoritmo de Fournier [1] que divide recursivamente un segmento. El procedimiento, que se muestra a continuación, divide recursivamente una línea (t_1, f_1) , (t_2, f_2) generando un desplazamiento del punto medio en la dirección perpendicular. El desplazamiento es de un valor aleatorio.

```

Fractal(t1, f1, t2, f2, resolucion, rugosidad)
  real r, tmid, fmid
  if (sqr(t2-t1)+sqr(f2-f1)) < sqr(resolucion))
    polilinea(2, t1, t2, f1, f2)
  else
    r = rand(-1,1)
    tmid = (t1+t2)/2 - rugosidad*(f2-f1)*r
    fmid = (f1+f2)/2 + rugosidad*(t2-t1)*r
    Fractal(t1, f1, tmid, fmid, resolucion, rugosidad)
    Fractal(tmid, fmid, t2, f2, resolucion, rugosidad)
  endif

```

La figura 10 muestra el resultado de aplicar el método a un segmento inicial en sucesivas iteraciones.

Esta misma idea se aplica para generar formas geográficas muy realistas. Partiendo de una superficie, se aplica el método de divisiones a las aristas, formando nuevas superficies. En el punto medio de cada arista, se levanta un nuevo vértice en la dirección normal a la superficie, y a una distancia aleatoria. Repitiendo el proceso hasta una resolución determinada, conseguimos un conjunto de superficies con orientaciones diferentes formando un todo (ver figura 10)

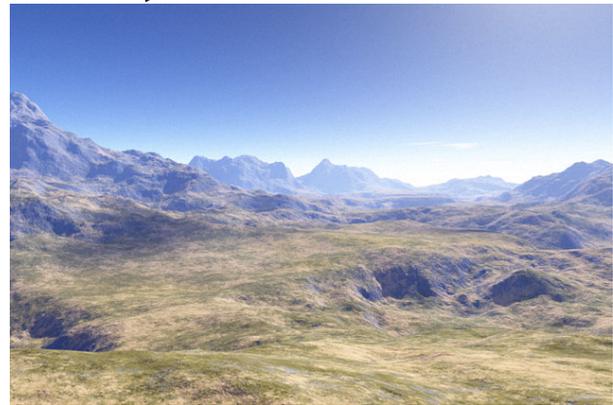
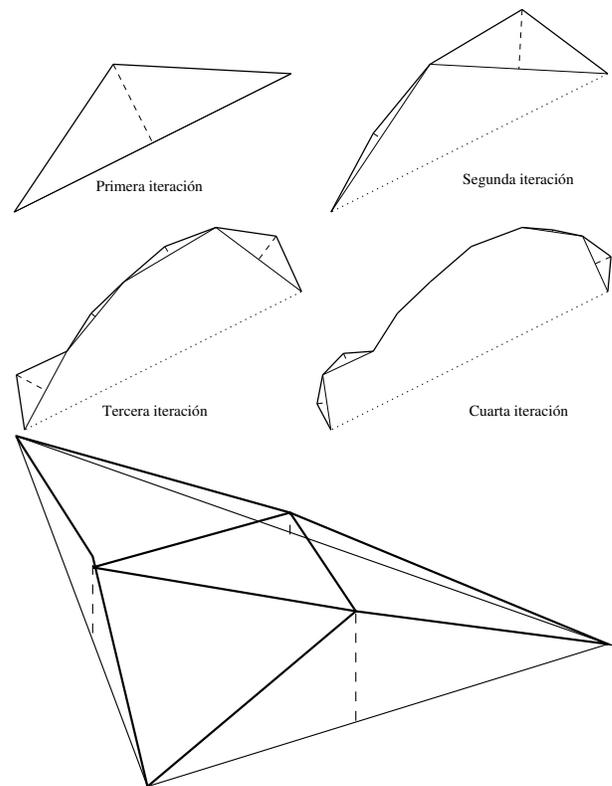


Figura 10: División iterativa de una línea en segmentos para formar un terreno. Abajo un posible resultado conseguido por esta técnica con la herramienta Terragen (imagen propiedad de T Ostrich CC BY-SA vía Wikipedia).

IX. METABALLS

Esta técnica permite construir objetos suavizados que simulan la interacción orgánica entre objetos menores. Es una técnica muy empleada para definir flujos de fluidos como los que se muestran en la figura 11.

La fórmula genérica que se emplea para representar el volumen interior a una superficie metaball es $\sum_{i=0}^n metaball_i(x, y, z) \leq umbral$. El metaball es una envolvente que incluye los n objetos que lo componen. Cada metaball i tiene asociada una fórmula, típicamente $metaball(x, y, z) = 1/((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2)$ en la que x_0 , y_0 y z_0 son el centro del metaball. el metaball puede ser cualquier función, pero tiene sentido emplear alguna

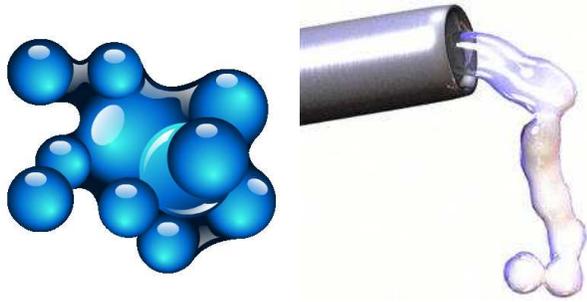


Figura 11: Representación de fluidos empleando metaballs (imágenes propiedad de Autodesk 3DS).

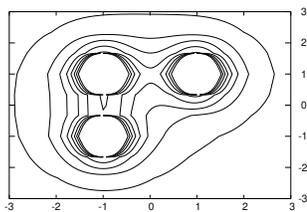


Figura 12: Las distintas curvas se obtienen al dar distintos valores al umbral de la suma de tres metaballs posicionados en los puntos $(1,1)$, $(-1,-1)$ y $(-1,1)$ y que responden a la fórmula $metaball_i = 1/(x - x_0) + (y - y_0)$

función que tenga alguna interpretación física. La fórmula que hemos presentado representa un campo de fuerza que es inversamente proporcional a la distancia. La figura 12 emplea dicha fórmula para en caso 2D.

X. ESCULTURA DIGITAL

En la técnica de escultura digital (*digital sculpting*), un objeto inicial se modela mediante operaciones que simulan el trabajo del escultor al moldear un trozo de manera o de piedra. Una pieza básica, es moldeada mediante la aplicación de presiones, recortes etc. . . .

Para hacer más eficiente el proceso, se suele ayudar de simetrías. Pueden partir de una malla 3D o de un conjunto de voxels. La figura 13 ilustra el proceso creativo de un torso de mujer partiendo de una pieza básica sobre la que se van aplicando operaciones de refinado. En la parte inferior se muestra como el proceso puede alcanzar modelos de alta calidad aplicando procesos interactivos de mejora.

Pueden usar el ratón pero generalmente usan dispositivos de entrada avanzados, como guantes de realidad virtual o similares.

MÁS INFORMACIÓN

Los libros básicos de referencia sobre informática gráfica, todos suelen incluir un capítulo o varios sobre representación

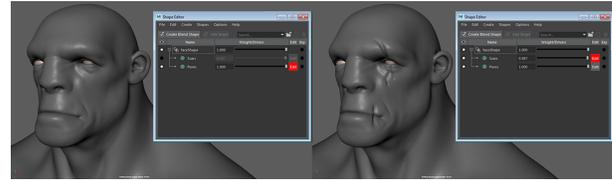


Figura 13: Objetos modelados mediante la técnica de escultura digital (imagen propiedad de Autodesk Maya).

de objetos 3D y modelado. Por ejemplo, [2] reserva el capítulo 8 para cuestiones de representación de mallas 3D; capítulo 10 en [3]; capítulos 3, 4 y 8 de [4];

Para ejercitar cuestiones prácticas de modelado, se recomienda utilizar un software de modelado 3D completo como puede ser 3D Studio [5] o Blender [6].

Para la parte de curvas de Bézier, los libros [7] y [8] están dedicados por completos a este tema. Los libros básicos de referencia también incluyen apartados para tratar las curvas paramétricas, pero en menor profundidad.

El artículo germinal de metaballs es [9]. Para profundizar con digital sculpting, recomiendo utilizar algún software específico como ZBrush [10] o su contrapartida de dominio público FreeD [11].

EJERCICIOS Y CUESTIONES

1. Explique en qué consiste la técnica de partición espacial.
2. ¿A qué se refiere el término representaciones frontera?
3. ¿Qué es un voxel?
4. Enumere 6 técnicas de modelado sólido.
5. Describa la técnica de modelado que se emplea para definir las esferas de la figura 14.

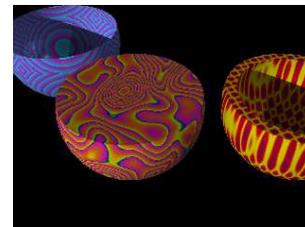


Figura 14: Ejemplos de figuras construidas empleando diferentes técnicas de modelado.

6. Cómo modelaría los objetos que están representados en la figura (parte derecha) empleando GSC.
7. Si tuviera que representar una montaña en el horizonte de una imagen renderizada, nombre dos posibles alternativas para simular dichas montañas.
8. Indique qué es un sólido de barrido.
9. De las diferentes técnicas de modelado de objetos conocidas, al menos dos de ellas se emplean para definir formas vegetales. Indique cuales son indicando de qué manera dichas técnicas abordan el modelado de formas vegetales.
10. ¿Por qué se usan fractales para construir escenas realistas?

11. Las técnicas de modelado que emplean representaciones frontera y las técnicas de *digital sculpting* tienen entre sí una diferencia radical, ¿puede indicar cual es dicha diferencia radical?
12. Si tuviera que representar el suelo de una playa en una escena 3D, nombre dos posibles alternativas para simular dicho terreno.
13. Los *metaballs* se suelen emplear para modelar fluidos. ¿Por qué cree que no se emplea la técnica de *instanciación de primitivas* para modelar fluidos?
14. Haga una estimación de la reducción del tamaño de almacenamiento del poliedro como el de la figura 15 si se utiliza una representación (1) *shared vertices* para los triángulos en la versión: superficie->aristas, aristas->vértices, vértices->puntos (x,y,z) frente a otra aproximación (2) que utilice una representación *separate triangles*: para cada superficie almacena los puntos (x,y,z). Cada coordenada o índice de vértices o aristas ocupa 32 bits.

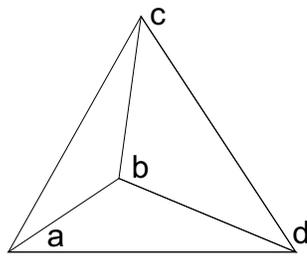


Figura 15: Vértices de poliedro.

15. Imagine que debe representar 1000 cuadrados cuyos vértices no coinciden. Haga una estimación del tamaño de almacenamiento total si se utiliza una representación (1) *shared vertices* para los polígonos en la versión: superficie->aristas, aristas->vértices, vértices->puntos (x,y,z) frente a otra aproximación (2) que utilice una representación *separate polygons*: para cada superficie almacena los puntos (x,y,z). Cada coordenada o índice de vértices o aristas ocupa 32 bits.
16. Repita el ejercicio anterior representando 100 triángulos.

REFERENCIAS

- [1] A. Fournier, D. Fussell, and L. Carpenter, "Computer rendering of stochastic models," *Communications of the ACM*, vol. 25, no. 6, pp. 371–384, 1982.
- [2] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [3] D. Hearn and M. P. Baker, *Computer Graphics, C Version*, segunda ed. Prentice Hall, 1997.
- [4] D. Hearn, M. P. Baker *et al.*, *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall., 2004.
- [5] Kinetix, *3D Studio MAX, Users Guide*. Autodesk, 2020.
- [6] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: <http://www.blender.org>
- [7] B. A. B. Richard H. Bartels, John C. Beatty, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Inc, 1986.
- [8] F. Yamaguchi, *Curves and Surfaces in Computer Aided Geometric Design*. Springer-Verlag, 1988.
- [9] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM transactions on graphics (TOG)*, vol. 1, no. 3, pp. 235–256, 1982.

- [10] S. Spencer, *ZBrush character creation: advanced digital sculpting*. John Wiley & Sons, 2011.
- [11] A. Zoran and J. A. Paradiso, "Freed: a freehand digital sculpting tool," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2613–2616.

Transformaciones geométricas

ÍNDICE

I.	Transformaciones y coordenadas homogéneas	2
II.	Transformadas básicas	2
II-A.	Traslación	2
II-B.	Rotación	2
II-C.	Escala	3
III.	Otras transformaciones	4
III-A.	Corte	4
III-B.	Reflexión	4
IV.	Composición de Transformaciones	4
IV-A.	Rotación desde un punto pivote general	4
IV-B.	Escala de punto fijo general	5
V.	Primitivas de transformaciones	5
VI.	El grafo de escena	5
	Referencias	7

Dada una malla de puntos 2D o 3D o dado un modelo 3D, podemos modificarlo aplicando sobre él una serie de transformaciones geométricas. En este tema veremos cómo, las distintas transformaciones geométricas que podamos aplicar sobre un objeto 3D, se traduce en la creación de una matriz que se utiliza para multiplicar los puntos del objeto 3D a transformar. Las transformaciones básicas son la traslación, la rotación y el escalado. Existen otras más sofisticadas, pero finalmente cada transformación se traduce en una matriz numérica diferente que multiplica a los puntos del objeto 3D.

En este tema primero describimos el concepto de coordenadas homogéneas, necesarios para poder aplicar multiplicaciones de matrices para realizar transformaciones que impliquen traslaciones de objetos. Después describimos las transformaciones básicas y las transformaciones compuestas. Finalizamos el tema presentando las primitivas para la creación y aplicación de matrices de transformación. Finalizamos presentando el concepto de grafo de escena y su relación con la aplicación de matrices de transformación.

I. TRANSFORMACIONES Y COORDENADAS HOMOGÉNEAS

En informática gráfica, realizar la transformación de un punto o vector \bar{v} en otro nuevo punto \bar{v}' se resuelve multiplicando el punto \bar{v} por una matriz. Dado un vector $\bar{v} = (x, y, z)$, si multiplicamos el vector por una matriz A , transformamos el vector en otro nuevo vector $\bar{v}' = (x', y', z')$. Si el vector representa un punto, el resultado es un cambio de posición de dicho punto. El vector \bar{v} se transforma en \bar{v}' al realizar la operación de multiplicación $\bar{v}' = A \cdot \bar{v}$. En dos dimensiones:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \end{pmatrix} \quad (1)$$

Las transformaciones que se pueden expresar de la forma $\bar{v}' = A \cdot \bar{v}$ se llaman **transformaciones lineales**.

Pero no todas las transformaciones con las que nos vamos a encontrar pueden ser expresadas mediante transformaciones lineales. En particular, y como veremos, las transformaciones de traslación necesitan ser expresadas mediante una **transformación afín**, que se expresa como $\bar{v}' = A \cdot \bar{v} + \bar{b}$. En dos dimensiones tendríamos la expresión:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y + b_1 \\ a_{21}x + a_{22}y + b_2 \end{pmatrix} \quad (2)$$

Para poder expresar transformaciones afines empleando una única matriz, se introducen las **coordenadas homogéneas**. Un punto de \mathbb{R}^3 (x, y, z) se representa en coordenadas homogéneas como (x, y, z, w) o como $(x, y, z, 1)$. En \mathbb{R}^2 , el punto (x, y) se representa en coordenadas homogéneas como (x, y, w) o como $(x, y, 1)$. Sólo es un artificio que empleamos para poder representar las transformaciones geométricas como operaciones con matrices. En el caso anterior, la misma transformación quedaría descrita como:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y + b_1 \\ a_{21}x + a_{22}y + b_2 \\ 1 \end{pmatrix} \quad (3)$$

Trabajar en coordenadas homogéneas implica añadir una fila y columna adicional a las matrices de transformación. Si trabajamos en dos dimensiones, las matrices de transformación serán de 3×3 , mientras que si trabajamos en tres dimensiones las matrices de transformación serán de 4×4 .

II. TRANSFORMADAS BÁSICAS

Dada un objeto 3D o 2D podemos modificarlo aplicando sobre él una serie de transformaciones. Las transformaciones básicas son la traslación, la rotación y el escalado. A continuación describiremos en detalle cada una de ellas. Describir la transformación consiste en encontrar la matriz que, al multiplicarla por los puntos del objeto, realice la transformación esperada.

II-A. Traslación

Realizamos una traslación sobre un objeto cuando queremos que cambie su posición a lo largo de una línea recta. La figura 1 muestra como los objetos 2D (la muñeca y el triángulo) se

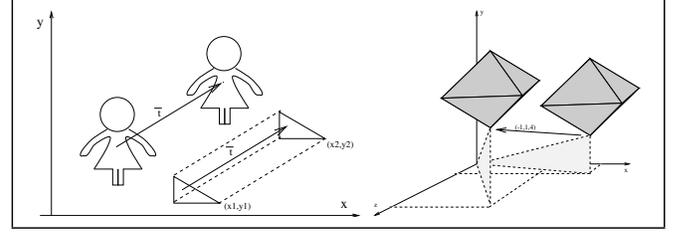


Figura 1: Traslación de un objeto bidimensional (izquierda) y tridimensional (derecha)

desplazan a su derecha siguiendo la dirección indicada por el vector \bar{t} . El objeto 3D de la parte izquierda de la figura 1 se desplaza siguiendo la dirección $(-1, 1, 4)$.

En 2D, si un punto está en la posición (x, y) y la posición final a la que queremos trasladarlo es la (x', y') , entonces el vector de traslación o de cambio $\bar{t} = (t_x, t_y)$ será tal que:

$$x' = x + t_x \quad y' = y + t_y \quad (4)$$

Empleando coordenadas homogéneas la matriz de traslación sería:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = T(t_x, t_y) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (5)$$

\bar{t} es el vector de desplazamiento y $T(t_x, t_y)$ es la matriz de traslación. En 3D, si un punto está en la posición (x, y, z) y la posición final a la que queremos trasladarlo es la (x', y', z') entonces el vector de traslación o de cambio (t_x, t_y, t_z) será:

$$x' = x + t_x \quad y' = y + t_y \quad z' = z + t_z \quad (6)$$

Empleando coordenadas homogéneas la matriz de traslación sería:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = T(t_x, t_y, t_z) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (7)$$

Para trasladar un objeto aplicaremos la transformada a cada uno de los puntos que lo componen.

II-B. Rotación

Realizamos una rotación sobre un objeto cuando queremos que cambie su posición a lo largo de un arco de circunferencia.

Para especificar una rotación es necesario especificar el ángulo de rotación θ , y el punto de pivote (x_r, y_r) . De momento las rotaciones que vamos a especificar en esta sección utilizan en centro de coordenadas $(0, 0)$ como punto pivote. La figura 2 (parte superior) muestra una rotación de un ángulo θ en sentido antihorario de un segmento y de una figura 2D (el teléfono).

Como se aprecia en la figura 2 para un punto (x, y) que queremos rotar hasta el punto (x', y') desde el centro de coordenadas, tenemos que:

$$\begin{aligned} x' &= r \cos(\phi + \theta) = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta) \\ y' &= r \sin(\phi + \theta) = r \cos(\phi) \sin(\theta) + r \sin(\phi) \cos(\theta) \end{aligned} \quad (8)$$

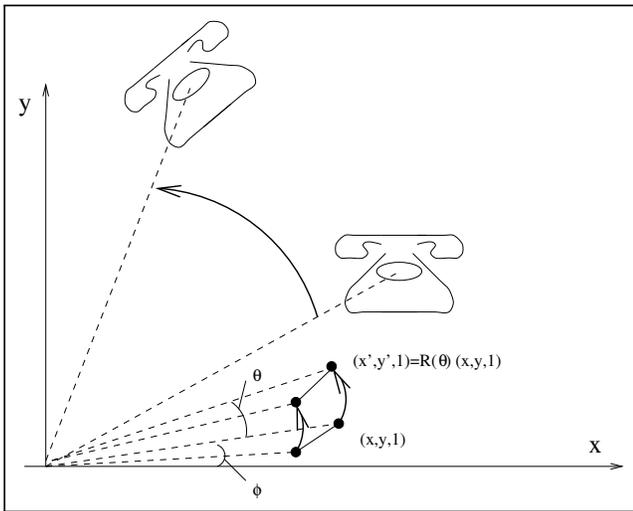


Figura 2: Rotación de un objeto bidimensional

Las coordenadas originales son :

$$x = r \cos(\phi) \quad y = r \sin(\phi) \quad (9)$$

Con lo que al sustituir tenemos

$$\begin{aligned} x' &= x \cos(\theta) - y \sin(\theta) \\ y' &= x \sin(\theta) + y \cos(\theta) \end{aligned} \quad (10)$$

En coordenadas homogéneas la matriz de transformación sería:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = R(\theta) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (11)$$

Para representar la rotación desde un punto pivote distinto al centro del sistema de coordenadas, lo haremos por composición de transformaciones y lo veremos en la próxima sección.

Para especificar una rotación en 3D es necesario especificar el ángulo de rotación θ , y el eje de rotación. Las rotaciones elementales se realizan con respecto a los ejes OX, OY y OZ. En la figura 3 se muestra cómo la posición final de la figura es diferente en función del eje con respecto al cual se hace la rotación.

II-B1. Rotación en el eje OZ: En coordenadas homogéneas la matriz de transformación sería:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = RoZ(\theta) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (12)$$

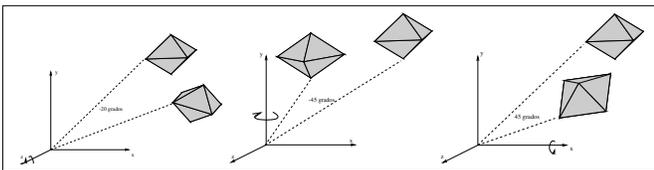


Figura 3: Rotación de un objeto tridimensional con respecto al eje OZ (izquierda), OY (centro) y OX (derecha).

II-B2. Rotación en el eje OX: En coordenadas homogéneas la matriz de transformación sería:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = RoX(\theta) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (13)$$

II-B3. Rotación en el eje OY: En coordenadas homogéneas la matriz de transformación sería:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = RoY(\theta) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (14)$$

II-C. Escala

Una transformación de escala altera el tamaño de un objeto. Para hacer la escala de un objeto multiplicamos los puntos que lo componen por los factores de escala s_x y s_y .

$$x' = s_x \cdot x, \quad y' = s_y \cdot y \quad (15)$$

En forma de transformadas homogéneas, aparece la matriz de transformación:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = E(s_x, s_y) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (16)$$

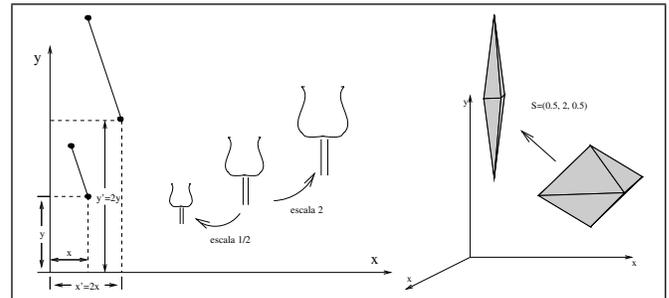


Figura 4: Escalado de objetos y puntos en 2D (izquierda) y en 3D (derecha).

Para hacer la escala de un objeto en 3D multiplicamos los puntos que lo componen por los factores de escala s_x , s_y y s_z .

$$x' = s_x \cdot x, \quad y' = s_y \cdot y, \quad z' = s_z \cdot z \quad (17)$$

En forma de transformadas homogéneas, aparece la matriz de transformación.

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = E(s_x, s_y, s_z) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (18)$$

Los objetos que se transforman aplicando las ecuaciones anteriores se escalan y cambian de posición. Los factores de escala menores que 1 acercan el objeto al centro de coordenadas y los mayores que 1 lo alejan (ver figura 4). Para evitar este efecto, muchas veces indeseable, debe aplicarse una transformación con respecto al punto pivote que veremos en las secciones siguientes.

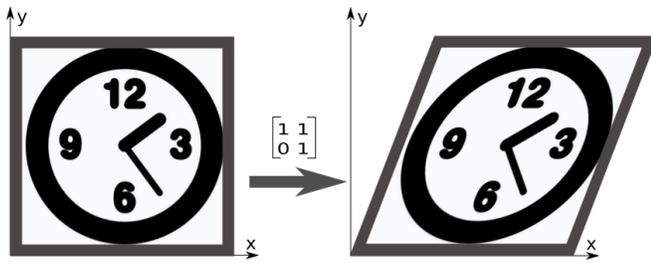


Figura 5: Ejemplo de operación de corte.

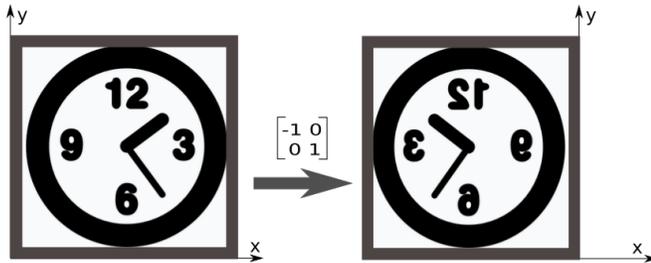


Figura 6: Ejemplo de operación de reflejo.

III. OTRAS TRANSFORMACIONES

III-A. Corte

Una transformación de corte (del inglés *shear*) sirve para desplazar cada punto en una dirección dada en una cantidad que es proporcional a su distancia al origen. Se consigue un efecto de inclinación muy útil para dar la sensación de perspectiva. Las matrices de transformación en la dirección X e Y en el caso 2D son:

$$\text{CorteX}(s) = \begin{pmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{CorteY}(s) = \begin{pmatrix} 1 & 0 & 0 \\ s & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (19)$$

El factor s determina la cantidad de desplazamiento en la dirección dada. La figura muestra un ejemplo de operación de corte 5.

III-B. Reflexión

Una transformación de reflexión realiza la imagen especular de la figura geométrica sobre la que se aplica. Para ello hay que establecer un eje o plano de reflexión dependiendo de si trabajamos en 2D o 3D. Las matrices de reflexión en el eje X y en el eje Y en 2D son respectivamente:

$$\text{ReflexX}(s) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{ReflexY}(s) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (20)$$

La figura 6 muestra una operación de reflejo con respecto al eje y.

IV. COMPOSICIÓN DE TRANSFORMACIONES

Dadas las matrices de transformación elementales $T(t_x, t_y)$, $R(\theta)$, $E(s_x, s_y)$ podemos combinarlas para crear transformaciones más complejas. Para ello empleamos la multiplicación

Figura 7: Ejemplo de operación composición de matrices extraído de [1]

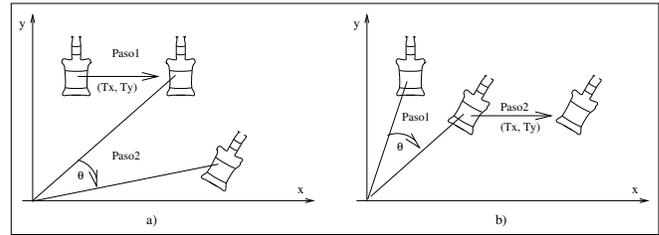


Figura 8: Invertir el orden de las transformaciones altera significativamente el resultado final: en la parte (a) primero se traslada y luego se gira. En la parte (b) primero se gira y luego se traslada.

de matrices. De hecho, la composición de transformaciones puede aplicarse a cualquier matriz de transformación y no sólo a traslaciones, rotaciones o escalas.

De forma general A y B son matrices de transformación entonces $C = A \cdot B$ es una nueva matriz de transformación. El resultado de aplicar C es el mismo que el de aplicar B y luego A al resultante. Dicho de otro modo, si partimos de \vec{v}_1 y queremos aplicar C para obtener \vec{v}_3 , aplicamos $\vec{v}_3 = C\vec{v}_1$, o lo que es lo mismo $\vec{v}_3 = A \cdot B\vec{v}_1$. Si llamamos $v_2 = v_1 \cdot B$, $v_3 = Av_2$. Lo que significa que las operaciones compuestas se pueden realizar paso a paso (multiplicando los vectores resultantes por cada matriz) o de una vez (multiplicando primero las matrices) obteniendo el mismo resultado: $\vec{v}_3 = A(B\vec{v}_1) = (AB)\vec{v}_1$. La figura 7 ilustra el proceso aplicando primero una escala y luego una rotación. La transformación compuesta puede hacerse paso a paso o de una vez multiplicando las matrices.

La multiplicación de matrices cumple la propiedad asociativa pero no la conmutativa. Quiere esto decir que alterar el orden en el que se hacen las transformaciones elementales puede alterar considerablemente el resultado final. En la figura 8 se observa un ejemplo que ilustra este hecho: el resultado es diferente si primero realizamos la traslación y luego la rotación ($T(t_x, t_y)R(\theta)$) o si primero realizamos la rotación y luego la traslación ($R(\theta)T(t_x, t_y)$).

Nos centramos en detallar con transformaciones compuestas cuyo uso es más frecuente: la rotación desde un punto pivote y la escala con respecto a un punto fijo.

IV-A. Rotación desde un punto pivote general

Con la transformación de rotación que vimos en el apartado II-B sólo podemos hacer giros respecto al origen de coordenadas. Para poder girar un objeto desde un punto pivote (x_p, y_p) proponemos la siguiente combinación de transformaciones.

1. Hacemos una transformación de manera que el punto fijo pase a estar en el centro de coordenadas.
2. Giramos el objeto el ángulo indicado con respecto al centro de coordenadas.
3. Traslamos el objeto de manera que el punto pivote regrese a su posición original.

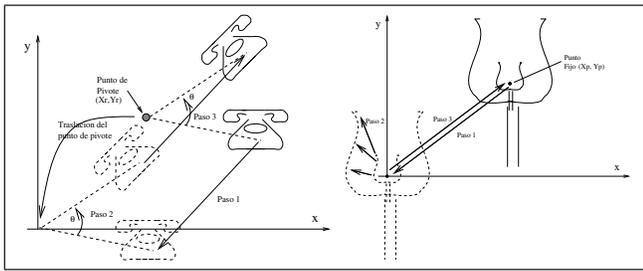


Figura 9: Rotación desde punto pivote (izquierda) y escalado sobre punto fijo (derecha)

Podemos expresar estos puntos como:

$$R(x_p, y_p, \theta) = T(x_p, y_p)R(\theta)T(-x_p, -y_p) \quad (21)$$

En la figura 9 vemos como para rotar la imagen empleando el punto pivote, primero hemos de trasladar la escena hasta hacer coincidir dicho punto con el origen, después realizamos la rotación, y por último deshacemos la traslación.

IV-B. Escala de punto fijo general

La transformada de escala descrita en el apartado II-C tiene el inconveniente de que los objetos cambian de posición. Para evitarlo podemos plantear una nueva transformación llamada transformación de punto fijo (x_p, y_p) de manera que especifiquemos un punto y garantizamos que dicho punto no se traslada. Para realizar esta transformación:

1. Hacemos una transformación de manera que el punto fijo pase a estar en el centro de coordenadas.
2. Realizamos el escalado.
3. Deshacemos la primera transformación.

Podemos expresar esta composición de transformaciones como:

$$E(x_p, y_p, s_x, s_y) = T(x_p, y_p)E(s_x, s_y)T(-x_p, -y_p) \quad (22)$$

En la figura 9 se observa como para escalar la figura representada desde el punto fijo, primero es necesario trasladar dicho punto al origen, después realizar la escala y por último deshacer la traslación.

Aunque estos ejemplos son en 2D, la extensión a 3D es inmediata.

V. PRIMITIVAS DE TRANSFORMACIONES

En un programa de diseño gráfico no vemos las matrices de transformación, pero internamente se crean y se utilizan constantemente. Por ejemplo cuando en Inkscape trasladamos un objeto, al mover el ratón estamos indicando cual el vector de desplazamiento que el programa debe utilizar para crear la matriz correspondiente y aplicarla a cada punto del objeto.

En el caso de SVG, todos los objetos tienen un atributo que se llama *transform* que se utiliza para especificar las operaciones de transformación geométrica que se deben aplicar sobre un objeto dado. Así por ejemplo:

```
<text x="60" y="105"
  transform="translate(30) rotate(45 50 50)">
  Hola Mundo </text>
```

hace que el texto se desplace 30 unidades en x y se rote 45 grados empleando el punto (50, 50) como pivote. Las transformaciones se aplican en orden inverso a como son declaradas, en este caso primero se rota y luego se traslada. Entre las transformaciones que incluye el lenguaje están *translate*, *scale*, *rotate*, *skewX*, *skewY* y *matrix*. Esta última permite especificar literalmente la matriz de transformación completa.

VI. EL GRAFO DE ESCENA

La mayoría de escenas admite una organización jerárquica. La figura 10 extraída de [1] muestra las relaciones jerárquicas de un grupo de objetos sencillos.

En el estándar de especificación de gráficos 3D VRML (*Virtual Reality Modeling Language*, se utiliza el concepto de **nodo** para agrupar objetos. Un nodo puede agrupar diferentes formas (*shapes*) que contienen a su vez los elementos gráficos.

En la figura 10 podría haber un nodo que contenga la carrocería del coche y las ruedas. Otro nodo diferente podría incluir el barco y el coche. La relación jerárquica entre nodos configura el grafo de escena. Utilizar un grafo de escena tiene la ventaja de que diferentes elementos de la escena pueden ser tratados como un todo.

VRML tiene un tipo de nodo especial que se llama **Transform Nodes**. A la hora de aplicar una transformación geométrica, las transformaciones se asocian a un nodo que está en un determinado nivel de la jerarquía. Con este tipo de nodos se declaran una serie de transformaciones que afectan a todos los nodos hijos. En el siguiente ejemplo:

```
Transform {
  scale 2 1 2
  rotation 0 1 0 0.78
  translation 1 1 1
  children [
    ...
  ]
}
```

Todos los objetos que se declaran dentro del apartado *children* se ven afectados por las transformaciones de escala, rotación y traslación que se declaran justo antes. Cuando se computan las transformaciones todas ellas afectan a todos los elementos vinculados como hijos en niveles inferiores de la jerarquía.

MÁS INFORMACIÓN

Los libros básicos de referencia sobre informática gráfica, todos suelen incluir un capítulo o varios sobre transformaciones geométricas. Por ejemplo, el capítulo 5 de [2] está dedicado a este asunto con una interesante extensión dedicada al cómputo eficiente de estas matrices. En [3] se profundiza dedicando dos capítulos, uno para las transformaciones en 2D (capítulo 10) y otro para las transformaciones en 3D (capítulo 11). Además, el capítulo 12 se dedica a cuestiones de implementación que tienen que ver con la sección V dedicada a primitivas gráficas. El capítulo 6 de [1] es el que más se ajusta al nivel presentado en este tema, habiendo reutilizado incluso varias ilustraciones.

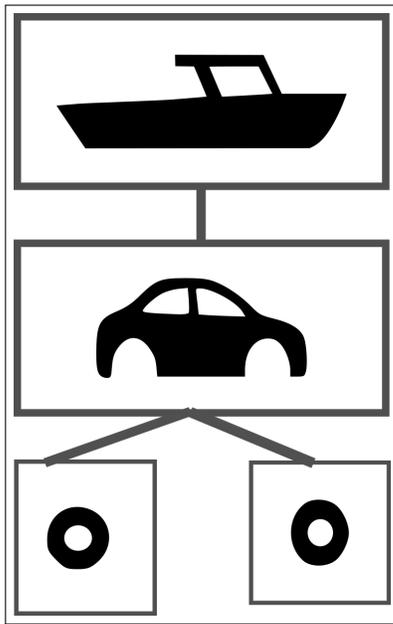


Figura 10: Los objetos que componen una escena, suelen estar organizados jerárquicamente. En este ejemplo, sobre el ferry hay un vehículo con cuatro ruedas. Se espera que al mover el ferry se mueva también el vehículo. Al mover el vehículo se han de mover sus ruedas.

EJERCICIOS Y CUESTIONES

1. Detalle la composición de transformaciones geométricas para hacer una rotación con punto pivote en 2D.
2. Qué son coordenadas homogéneas y por qué se utilizan. Cómo se representa el vector 3D $\mathbf{x} = (3, 7, 4)$ en coordenadas homogéneas?
3. Detalle la transformación de rotación en 2D.
4. Explique la transformación de traslación 2D.
5. ¿Por qué es necesario emplear transformaciones homogéneas en informática gráfica?
6. La multiplicación de matrices no satisface la propiedad conmutativa. ¿Qué implicaciones tiene esto a la hora de aplicar transformaciones geométricas sobre los objetos 3D? Ilústrelolo con un ejemplo.
7. Especificar las transformaciones necesarias para poder dibujar la figura 11 (parte superior izquierda) a partir del segmento definido por los puntos $(0, 0)$ y $(0, 1)$.
8. Desarrolle la matriz de transformación que se debe aplicar a los puntos del corazón que está en la posición $(5, 11)$ para transformarlos en el corazón que está en la posición $(12, 6)$ (figura 11 parte inferior izquierda). Observe que el corazón ha sido escalado al doble en una dimensión y rotado 90 grados. Se está pidiendo la matriz final. La matriz de rotación es en sentido antihorario.
9. Dado el segmento \mathbf{AB} donde $\mathbf{A} = (5, 2)$ y $\mathbf{B} = (3, 1)$, indique el resultado de hacer una rotación de 45° en el sentido antihorario y después una escala definida por el vector de factores de escala $(1, 2)$.
10. Desarrolle la matriz de transformación que se debe aplicar a los puntos del balón de la figura 11 que está en la

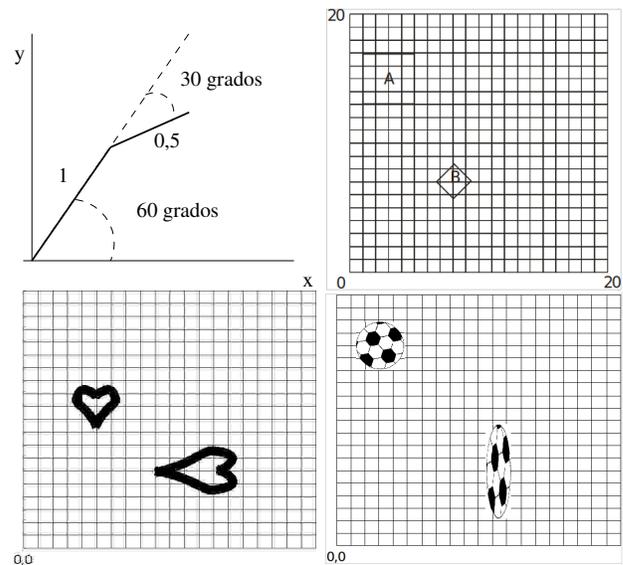


Figura 11: Diferentes figuras transformadas.

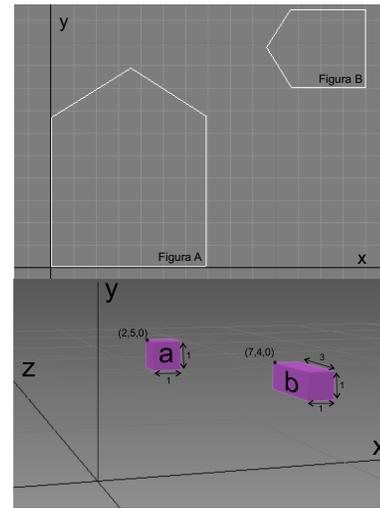


Figura 12: Diferentes figuras transformadas.

- posición $(3, 16)$ para transformarlos en el balón que está en la posición $(11, 6)$. Observe que el balón ha sido escalado a la mitad en una dimensión y al doble en la otra. Se está pidiendo la matriz final.
11. Especifique la secuencia de transformaciones geométricas que transforman la figura A en la figura B (ver figura 12). No es necesario realizar ninguna operación aritmética. Para especificar las transformaciones puede utilizar el siguiente nomenclátor:
 - $T(d_x, d_y)$ matriz de traslación en la dirección (d_x, d_y) .
 - $R(a)$ matriz de rotación del ángulo a en el eje z .
 - $E(e_x, e_y)$ matriz de escalado con factores de escala e_x y e_y .
 Para especificar el orden de la secuencia, indique el orden de multiplicación de las matrices de transformación T , R y E para obtener la matriz de transformación final.
 12. Especifique la secuencia de transformaciones geométricas

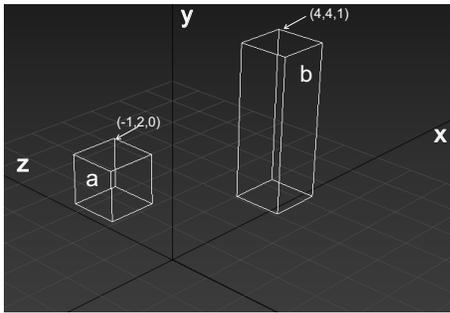


Figura 13: Diferentes figuras transformadas.

necesarias para convertir el cubo a de lado 1 de la figura en el paralelepípedo b (ver figura 12 derecha). Debe detallar cada matriz de transformación y multiplicarlas por los puntos del cubo a para comprobar que la solución es correcta.

13. Especifique la secuencia de transformaciones geométricas necesarias para convertir el cubo "a" de lado 1 en el paralelepípedo "b" de la figura 13 de lados $1 \times 1 \times 3$. Debe detallar cada matriz de transformación y multiplicarlas por los puntos del cubo "a" para comprobar que la solución es correcta.
14. En un interfaz de modelado 3D como 3D Studio, ¿en cuantos pasos podría hacer la transformación del cubo a en el paralelepípedo b de la figura 12 derecha utilizando las transformaciones de traslación y de escalado (no los modificadores)? Relacione cada uno de los pasos con las matrices especificadas en el ejercicio anterior. Si no hay relación, explique por qué.

REFERENCIAS

- [1] S. Marschner and P. Shirley, *Fundamentals of computer graphics*. CRC Press, 2015.
- [2] D. Hearn, M. P. Baker *et al.*, *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.
- [3] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.

Transformaciones de visión

ÍNDICE

I. Sistemas de coordenadas	2	Los programas informáticos de modelado 3D trabajan con escenas compuestas de objetos cuyos puntos se definen en coordenadas cartesianas tridimensionales. Para llegar a ver estos objetos en un plano (la pantalla del ordenador es un plano), es necesario realizar una serie de transformaciones geométricas sobre los puntos de estos objetos. Al conjunto de transformaciones geométricas que permiten ver los datos tridimensionales de los objetos a representar en una proyección plana, se le denomina tubería de visión 3D y a ellas está dedicado el presente tema.
I-A. Sistemas de coordenadas locales	2	
I-B. Sistema de coordenadas mundo	2	En este tema, primero se explica la importancia de tener claro el sistema de referencia con el que se trabaja. Después, se detallan las transformaciones de visión, las proyecciones y el volumen de visión. Finalizamos con la tubería de visión 2D, importante tanto en sistemas de dos dimensiones como en la etapa final de la transformada de visión.
I-C. Sistema de coordenadas de visión	2	
I-D. Cambio de sistema de coordenadas	2	
I-E. Sistema de mano derecha y sistema de mano izquierda	3	
II. Transformación de visión	3	
III. La transformación de cámara	4	
III-A. Especificación del sistema de referencia de visión	4	
III-B. Cambio de sistema de referencia	5	
IV. La transformación de proyección	5	
IV-A. Proyección paralela	6	
IV-B. Proyección perspectiva	6	
IV-C. Transformación de volumen canónico	7	
V. Tubería de vista en 2D	8	
Referencias	10	

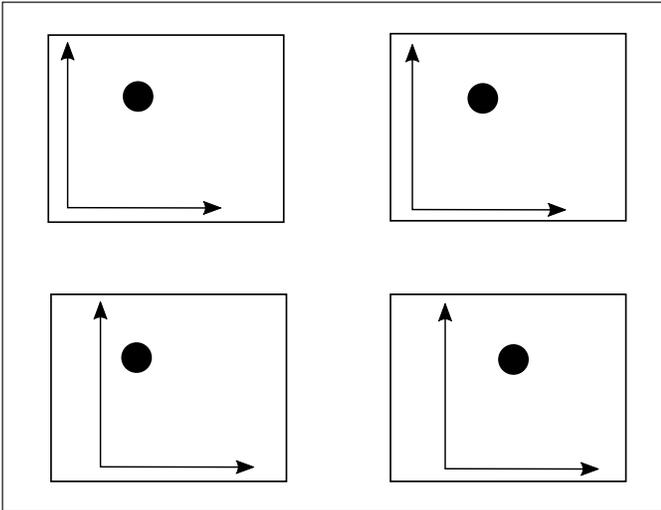


Figura 1: Importancia del sistema de referencia en la asignación de coordenadas: un mismo punto puede tener diferentes coordenadas (a la izquierda) y dos puntos diferentes pueden tener las mismas coordenadas (a la derecha).

I. SISTEMAS DE COORDENADAS

Un sistema de coordenadas o sistema de referencia es un recurso matemático que sirve para posicionar puntos en el espacio. Descartes inventó este recurso en el siglo XVII para su uso en geometría. Para establecer un sistema de referencia se necesita fijar el centro de coordenadas y dos o tres ejes en función de si trabajamos en dos o tres dimensiones. Los puntos se referencian indicando la distancia al centro de coordenadas siguiendo los ejes de coordenadas. La figura 1 ilustra la importancia de la elección del sistema de referencia para posicionar los puntos en el espacio. En la transformada de visión hay tres sistemas de referencia que juegan un papel principal: el sistema de coordenadas local, el sistema de coordenadas mundo y el sistema de referencia de visión. A ellos dedicaremos los siguientes apartados.

I-A. Sistemas de coordenadas locales

Para facilitar el modelado, los vértices de los objetos 3D se posicionan con respecto a un sistema de referencia localizado cerca del objeto, típicamente en el centro o en algún vértice de la base; es el sistema de referencia local.

Los vértices del objeto se anotan con respecto a dicho sistema de referencia local que será diferente para cada objeto. Cuando aplicamos transformaciones geométricas sobre el objeto, las aplicamos utilizando las coordenadas locales de los objetos.

I-B. Sistema de coordenadas mundo

Una vez modelado el objeto hay que ubicarlo en la escena. Todos los objetos que configuran la escena tienen su propio sistema de referencia local. El sistema de referencia global de la escena se llama sistema de referencia mundo.

La construcción de una escena puede ser vista como la construcción de sistema mundo a partir de los sistemas de

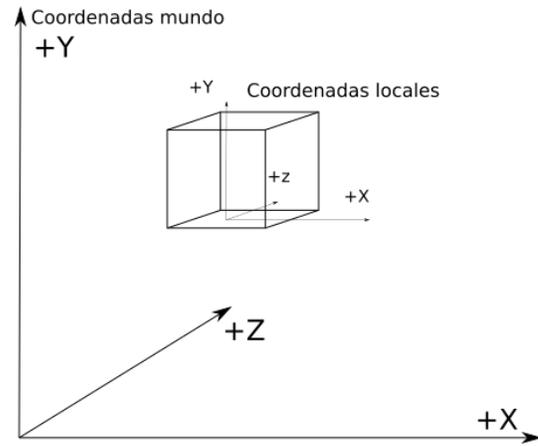


Figura 2: Los puntos del cubo tienen anotaciones diferentes si se emplea el sistema de referencia local (*local coordinates*) o el sistema de referencia mundo (*world coordinates*).

coordenadas locales. Para construir una escena, todos los objetos deben cambiar de sus coordenadas desde el sistema local al sistema mundo.

La figura 2 ilustra las diferencias entre estos sistemas de referencia. Los puntos del cubo se construyen empleando el sistema de referencia local, pero al formar parte de una escena mayor, sus coordenadas deben convertirse de coordenadas locales en coordenadas mundo.

I-C. Sistema de coordenadas de visión

Empleando el sistema de referencia mundo hacemos que los puntos de todos los objetos de la escena estén posicionados con respecto a un sistema común. Sin embargo, para visualizar la escena este sistema de referencia mundo no es suficiente. Para visualizar los objetos nos interesa otro sistema de referencia que es aquel que determina el punto desde el que se observa la escena.

Para entender la importancia de este sistema de referencia, podemos utilizar la metáfora de la cámara. Los distintos objetos pueden estar todos en coordenadas mundo, pero finalmente interesa saber su posición relativa con respecto a la cámara porque es desde ese punto desde el que se visualiza la escena.

El sistema de referencia de visión es importante porque, entre otras cosas, es el que establece qué objetos son visibles y cuáles están ocultos. La figura 3 muestra la relación en sistema de referencia mundo (SRM) y el sistema de referencia de visión (SRV) que tiene su propio centro de coordenadas en el centro de visión (CV). Al posicionar una cámara fijamos CV y al orientarla fijamos los ejes. En la sección II detallamos el proceso completo incluida la formulación.

I-D. Cambio de sistema de coordenadas

El problema del cambio de sistema de referencia aborda cómo obtener las coordenadas de un punto en un sistema de referencia cuando se conocen sus coordenadas en otro sistema diferente. En la figura 4 se ilustra el cambio de sistema de coordenadas para el punto \bar{p} . En el sistema de referencia

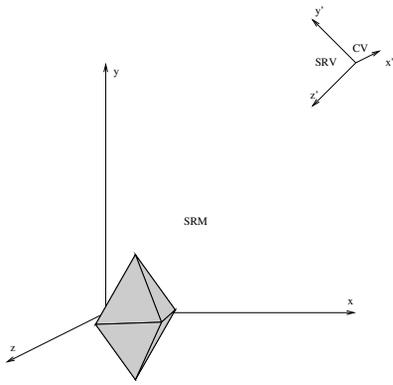


Figura 3: La cámara establece su propio sistema de referencia SRV

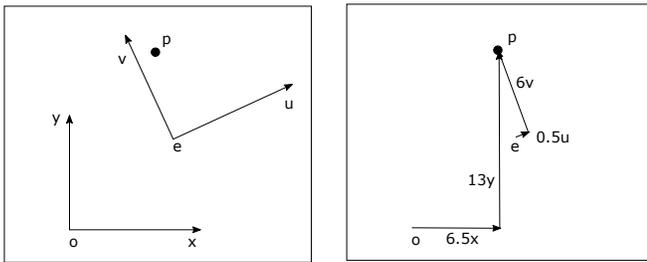


Figura 4: El cambio de sistema de referencia implica un cambio en los valores de las coordenadas

x, y con centro en \bar{o} , el punto tiene como coordenadas $\bar{p} = (x_p, y_p) = (6,5,13)$. A la vez, en el sistema de referencia u, v con centro en \bar{e} , el punto tiene las coordenadas $\bar{p} = (u_p, v_p) = (0,5,6)$.

Como

$$\bar{p}_{xy} = (x_p, y_p) = \bar{o} + x_p \bar{x} + y_p \bar{y} \quad (1)$$

$$\bar{p}_{uv} = (u_p, v_p) = \bar{e} + u_p \bar{u} + v_p \bar{v} \quad (2)$$

Si posicionamos el centro del sistema de referencia de uno de los dos sistemas con respecto al otro por ejemplo $\bar{e} = x_e \bar{x} + y_e \bar{y}$, y hacemos lo mismo con los ejes de referencia $\bar{u} = x_u \bar{x} + y_u \bar{y}$ y $\bar{v} = x_v \bar{x} + y_v \bar{y}$; como $\bar{p}_{xy} = \bar{e} + \bar{p}_{uv}$ llegamos al siguiente sistema de equivalencias:

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_e \\ 0 & 1 & y_e \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_u & x_v & 0 \\ y_u & y_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \begin{pmatrix} x_u & x_v & x_e \\ y_u & y_v & y_e \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} \quad (3)$$

Que de forma simplificada se puede expresar como:

$$\bar{p}_{xy} = \begin{pmatrix} \bar{u} & \bar{v} & \bar{e} \\ 0 & 0 & 1 \end{pmatrix} \bar{p}_{uv} \quad (4)$$

En la sección III-B se desarrollan estas ecuaciones en el caso de tres dimensiones para el caso particular del cambio de sistema de referencia mundo al sistema de referencia de visión.

Si una matriz permite realizar el cambio de coordenadas de un sistema de referencia $\bar{o}, \bar{x}, \bar{y}$ en otro $\bar{e}, \bar{u}, \bar{v}$ la matriz inversa realizará el cambio desde $\bar{e}, \bar{u}, \bar{v}$ hacia $\bar{o}, \bar{x}, \bar{y}$.

$$\begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \begin{pmatrix} x_u & y_u & 0 \\ x_v & y_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -x_e \\ 0 & 1 & -y_e \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \quad (5)$$

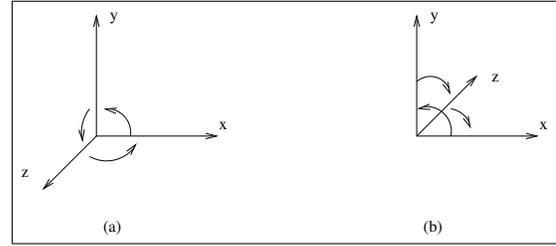


Figura 5: Sistema de ejes de mano derecha (a) y de mano izquierda (b).

I-E. Sistema de mano derecha y sistema de mano izquierda

Las coordenadas de los puntos pueden estar definidas con respecto a dos tipos de sistemas de referencia: los de mano derecha y los de mano izquierda. La diferencia (como vemos en la figura 5) está en la orientación del eje z , y en las direcciones positivas de las rotaciones en los ejes.

Si x está en la dirección del pulgar con los valores en sentido creciente hacia el extremo del dedo e y está en la dirección del dedo índice también con la misma orientación de los valores crecientes, entonces el dedo corazón indicaría la posición del eje z con el mismo criterio con respecto a los valores crecientes apuntando hacia el extremo del dedo.

Los sistemas de mano derecha son más cercanos a la representación de puntos a la que estamos habituados. Los de mano izquierda son importantes en algunos sistemas en los que aumentar el valor de z aporta connotación de lejanía, por ejemplo el sistema de referencia de visión en el que a mayor valor de z mayor distancia al centro de visión.

II. TRANSFORMACIÓN DE VISIÓN

Mediante la transformación de visión proyectamos una escena compuesta por un conjunto de puntos definidos en un espacio tridimensional en un plano bidimensional que acabará siendo la imagen de la escena que visualizamos en la pantalla.

Transformar los puntos de la escena, que están en un sistema cartesiano local (x, y, z) en otro sistema de coordenadas de imagen, que está en pixels de la pantalla se realiza siguiendo los pasos que se ilustran en la figura 6 y que se detallan a continuación.

Paso 1 Cambio de coordenadas a sistema mundo: Las primeras transformaciones que se realizan sobre cada objeto son las que se realizan en el proceso de modelado y de formación del grafo de escena. Cuando se crea un objeto normalmente se define en un sistema de coordenadas que es local y dependiente de objeto. Mediante un cambio de sistemas de coordenadas (dependiente de cada objeto) todas las coordenadas de los objetos de

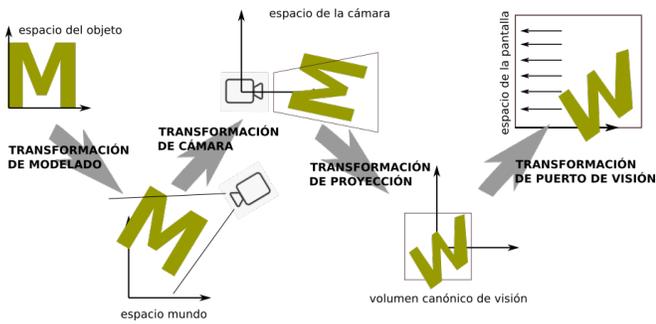


Figura 6: Pasos que llevan desde un objeto definido en su propio sistema de coordenadas a la vista del mismo en la imagen final.

la escena pasan a estar en el mismo sistema de referencia, llamado sistema de referencia mundo. Las fórmulas genéricas para realizar el cambio de sistema de referencia han sido presentadas en la sección I-D para el caso 2D y se volverán a mostrar para hablar del cambio de sistema de referencia mundo a sistema de referencia de visión en el apartado III-B

- Paso 2 **Transformación de cámara**, que se emplea para cambiar la representación de los puntos desde un sistema de coordenadas mundo a otro dependiente de la cámara o del punto de observación. Depende de la posición y orientación de la cámara (la pose). Se han de especificar los parámetros de visión: posición de la cámara y localización del plano de visión. Con estos parámetros se define el sistema de coordenadas de visión y después se hace el cambio de coordenadas mundo hacia sistema de coordenadas de visión. Esta transformación se describe en la sección III-A.
- Paso 3 **Transformación de proyección** que proyecta los puntos visibles en el espacio de la cámara en un plano con coordenadas (x, y) con valores normalizados entre 0 y 1. Para realizar este paso, es fundamental definir primero el tipo de proyección que se va a emplear: perspectiva o paralela. Como resultado, el volumen de visualización se proyecta en un plano bidimensional que es el plano de visión. Más concretamente se proyecta en una zona rectangular que se denomina ventana de visión.
- Paso 4 **Transformación del puerto de visión o de ventana de visión** donde el espacio normalizado se proyecta en una posición concreta de la pantalla. En esta transformación se tiene en cuenta el recorte de las tres dimensiones limitando la parte de la escena que queremos representar o volumen de visión. Se ofrecen más detalles, incluida la formulación, en la sección IV.

En informática gráfica, cada transformación se representa y ejecuta aplicando una multiplicación matricial. Cada una de las transformaciones listadas anteriormente tiene su reflejo en

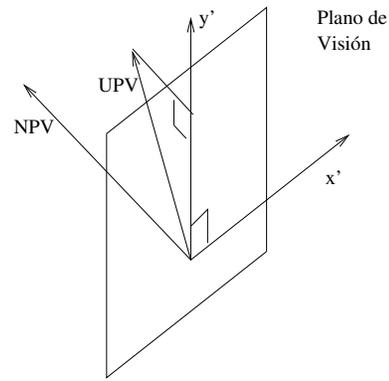


Figura 7: Especificación del SRV a partir de los vectores NPV y UPV

la definición de una matriz, de manera que un punto (x, y, z) en coordenadas mundo acaba siendo un punto (x', y', z') al multiplicar dicho punto por una matriz M_{SRV} que representa el cambio al sistema de referencia de visión; una matriz M_P servirá para obtener los nuevos proyectados (x'', y'', z'') al multiplicar los puntos (x', y', z') ; otra matriz M_{VC} convierte los puntos en el sistema SRV en puntos dentro de un volumen canónico. Finalmente, dichos puntos se convierten en puntos dentro de un puerto de visión en el dispositivo al multiplicarlos por una nueva matriz M_{2D} . En las siguientes secciones detallamos cada una de las transformaciones que se han presentado en esta sección.

III. LA TRANSFORMACIÓN DE CÁMARA

Para realizar la transformación de cámara, primero hay que especificar el sistema de referencia de visión desde el cual se verá la escena (posición y orientación). Después se realiza un cambio de sistema de referencia para pasar de coordenadas mundo a coordenadas de visión. En los siguientes apartados detallamos ambos pasos.

La transformación de cámara es importante porque permite conocer qué objetos están detrás y qué objetos están delante con respecto a la cámara, lo cual es fundamental para saber qué superficies deben ser representadas.

III-A. Especificación del sistema de referencia de visión

Los objetos de una escena tridimensional se definen en un sistema de coordenadas que denominamos *sistema de referencia mundo* SRM. El observador necesita conocer las coordenadas de los puntos con respecto a su propia referencia. Así, el ente que mira (sea un ojo, una cámara ...) debe establecer el sistema de referencia con respecto al cual se posicionan los objetos. Llamaremos a este sistema de *sistema de referencia de visión* SRV.

Para poder establecer las coordenadas de los objetos de la escena en el SRV, debemos posicionar y orientar una cámara en el SRM. Posicionar implica establecer el centro de referencia del SRV. A este punto lo llamaremos *centro de visión* o CV. Orientar la cámara implica establecer la dirección de los vectores ortogonales x' , y' y z' , que forman la base

del SRV. Para fijar dichos vectores, empleamos un *plano de visión* o PV con su correspondiente vector normal (*normal al plano de visión* o NPV). El plano de visión será empleado posteriormente para proyectar las escenas. El vector NPV hace el papel de vector \bar{z}' o vector de profundidad en el SRV. Los vectores \bar{x}' y \bar{y}' , son ortogonales a \bar{z}' y por lo tanto coplanarios a PV. Para encontrarlos, se define un vector llamado UPV (Up plano de visión). La proyección de este vector sobre el NPV define la dirección de \bar{y}' . El vector \bar{x}' ha de ser perpendicular a \bar{y}' y \bar{z}' . El SRV es un sistema de referencia de mano derecha al contrario que el sistema de coordenadas mundo, que suele ser un sistema de referencia de mano izquierda. Así las cosas, dados NPV y UPV puedo obtener \bar{x}' , \bar{y}' y \bar{z}' aplicando las expresiones:

$$\bar{z}' = NPV/|NPV| \quad (6)$$

$$\bar{x}' = \bar{z}' \wedge UPV \quad (7)$$

$$\bar{y}' = \bar{x}' \wedge \bar{z}' \quad (8)$$

En las figuras 3 y 7 se ilustra cada uno de los elementos referidos en los párrafos anteriores. La ecuación 6 sirve para normalizar el vector NPV (hacer que su módulo valga la unidad); la ecuación 7 permite obtener un vector perpendicular a UPV y NPV, es un vector coplanar con respecto al plano de visión; la ecuación 8 permite obtener un vector perpendicular a NPV y \bar{x}' sobre el que se proyecta UPV.

III-B. Cambio de sistema de referencia

Una vez posicionado y orientado el SRV, el siguiente paso es hacer el **cambio de base**. Para ello aplicaremos la matriz de cambio de base que se calcula a partir de las coordenadas de los vectores que definen el SRV y con el CV. Sean los vectores de la base del SRV \bar{x}' , \bar{y}' , \bar{z}' , de coordenadas y el vector CV:

$$\begin{aligned} \bar{x}' &= a_x \bar{x} + b_x \bar{y} + c_x \bar{z} \\ \bar{y}' &= a_y \bar{x} + b_y \bar{y} + c_y \bar{z} \\ \bar{z}' &= a_z \bar{x} + b_z \bar{y} + c_z \bar{z} \\ \overrightarrow{CV} &= a \bar{x} + b \bar{y} + c \bar{z} \end{aligned} \quad (9)$$

Llamemos **O** al centro de coordenadas del SRM y **O'** al CV (ver figura 8). Dado un punto \bar{p} cualquiera, se verifica que:

$$\bar{p} = \overrightarrow{CV} + \bar{p}' \quad (10)$$

pero

$$\begin{aligned} \bar{p} &= p_x \bar{x} + p_y \bar{y} + p_z \bar{z} \\ \overrightarrow{CV} &= a \bar{x} + b \bar{y} + c \bar{z} \\ \bar{p}' &= p'_x \bar{x}' + p'_y \bar{y}' + p'_z \bar{z}' \end{aligned} \quad (11)$$

por tanto, sustituyendo en 10 tenemos:

$$p_x \bar{x} + p_y \bar{y} + p_z \bar{z} = a \bar{x} + b \bar{y} + c \bar{z} + p'_x \bar{x}' + p'_y \bar{y}' + p'_z \bar{z}'$$

En esta expresión aparecen vectores representados en los sistemas de referencias SRV y SRM. Para expresarlos todos en el mismo sistema aplicamos las ecuaciones de 9:

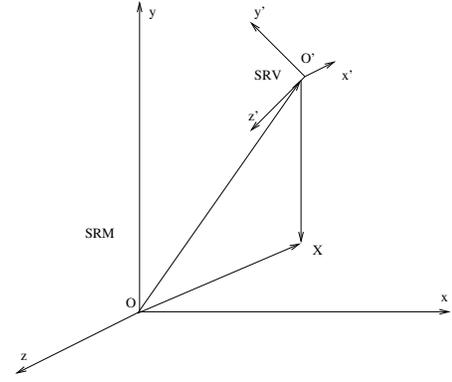


Figura 8: Punto X es los sistemas de referencia SRV y SRM

$$\begin{aligned} p_x \bar{x} + p_y \bar{y} + p_z \bar{z} &= a \bar{x} + b \bar{y} + c \bar{z} \\ &+ p'_x (a_x \bar{x} + b_x \bar{y} + c_x \bar{z}) \\ &+ p'_y (a_y \bar{x} + b_y \bar{y} + c_y \bar{z}) \\ &+ p'_z (a_z \bar{x} + b_z \bar{y} + c_z \bar{z}) \end{aligned} \quad (12)$$

que podemos escribir así:

$$\begin{aligned} p_x \bar{x} + p_y \bar{y} + p_z \bar{z} &= (a + p'_x a_x + p'_y a_y + p'_z a_z) \bar{x} \\ &+ (b + p'_x b_x + p'_y b_y + p'_z b_z) \bar{y} \\ &+ (c + p'_x c_x + p'_y c_y + p'_z c_z) \bar{z} \end{aligned} \quad (13)$$

que de forma matricial es:

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} a_x & a_y & a_z & a \\ b_x & b_y & b_z & b \\ c_x & c_y & c_z & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{pmatrix} = M_{SRV}^{-1} \cdot \begin{pmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{pmatrix} \quad (14)$$

La matriz que nos permite obtener las coordenadas en el SRV a partir de las coordenadas en el SRM es M_{SRV} .

$$M_{SRV} \cdot \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = M_{SRV} \cdot M_{SRV}^{-1} \cdot \begin{pmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{pmatrix} = \begin{pmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{pmatrix} \quad (15)$$

Este último paso es muy importante porque normalmente solemos tener la posición y orientación de la cámara en coordenadas mundo. Esta información nos da M_{SRV}^{-1} , por lo que es necesario invertir esta matriz para pasar los puntos \bar{p} definidos en coordenadas mundo a puntos \bar{p}' definidos en el sistema de referencia de visión.

IV. LA TRANSFORMACIÓN DE PROYECCIÓN

Una proyección transforma puntos en un sistema de dimensión n en otro de dimensión menor que n . En el caso que nos ocupa, transforma puntos en 3D en puntos en 2D.

Una proyección se define por unas rectas llamadas **proyectores** que emanan desde el **centro de proyección (CP)**, que pasan por cada punto de objeto y que intersectan con un **plano de proyección (PP)** para formar la **proyección**.

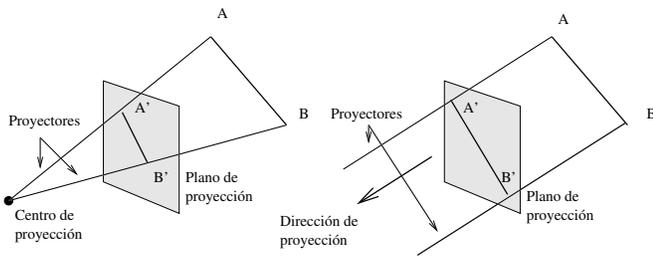


Figura 9: Ejemplo de proyección en perspectiva y de proyección paralela

Si la proyección se produce contra un plano y no en esferas ni superficies curvas, se denomina **proyección geométrica plana (PGP)**.

Existen dos tipos de PGP:

- Perspectiva que se produce cuando la distancia entre CP y PP es finita.
- Paralela cuando esa distancia tiende al infinito. Los proyectores son paralelos entre sí.

En la perspectiva se especifica un CP. En la paralela especificamos **la dirección de proyección (DP)** (ver figura 9)

La perspectiva tiene como ventaja que es más realista, mientras que la paralela tiene como ventaja que los ángulos se conservan.

La proyección paralela puede ser ortogonal u oblicua en función del ángulo que forman la normal al PP y la DP. En la ortogonal, este ángulo es de 90 grados, mientras que en la oblicua es distinto de 90 grados.

En los siguientes apartados veremos cómo podemos representar en forma matricial tanto la proyección paralela como la proyección en perspectiva.

IV-A. Proyección paralela

La proyección paralela se define con una dirección de proyección y un plano de proyección. A la hora de realizar la operación de proyección, los puntos de la escena están definidos en el SRV. Los parámetros que definen la proyección, también están definidos en dicho sistema. Así, especificaremos la dirección de proyección con un vector libre que denominaremos *vector de dirección de proyección (VDP)*. El plano de proyección que emplearemos será el plano X'Y' del SRV.

Cada punto $\bar{p} = (x', y', z')$ de la escena tiene por proyector la recta:

$$r_{\bar{p}} = \bar{p} + \lambda \bar{v} \quad (16)$$

$r_{\bar{p}}$ no es otra cosa que ecuación genérica para una recta que pasa un punto \bar{p} y que tiene una dirección \bar{v} . La intersección con el plano de proyección se produce cuando $z' = 0$ en la recta, por lo que podemos escribir

$$\begin{aligned} x_p &= x' + \lambda v_x \\ y_p &= y' + \lambda v_y \\ z_p &= z' + \lambda v_z = 0 \end{aligned} \quad (17)$$

De donde se sigue que $\lambda = -z'/v_z$, de modo que reescribimos las fórmulas como

$$\begin{aligned} x_p &= x' - \frac{z'}{v_z} v_x \\ y_p &= y' - \frac{z'}{v_z} v_y \\ z_p &= 0 \end{aligned} \quad (18)$$

Que nos lleva a definir la proyección paralela de forma matricial como:

$$\begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -v_x/v_z & 0 \\ 0 & 1 & -v_y/v_z & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad (19)$$

Si se trata de una proyección ortogonal, entonces los proyectores son perpendiculares al plano $z' = 0$. En este caso sólo hemos de eliminar la componente z' por lo que aplicamos la transformación

$$\begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad (20)$$

IV-B. Proyección perspectiva

La proyección perspectiva se define con un centro de proyección CP y un plano de proyección PP. De nuevo definiremos los parámetros en el SRV y emplearemos el plano $z = 0$ como plano de proyección. Fijo el CP= (x_{cp}, y_{cp}, z_{cp}) , los proyectores de cada punto $\bar{p} = (x', y', z')$ de la escena se definen como:

$$r_{\bar{p}} = \bar{p} + \lambda(\bar{C}P - \bar{p}) \quad (21)$$

$r_{\bar{p}}$ no es otra cosa que ecuación genérica para una recta que pasa dos puntos, \bar{p} y $\bar{C}P$. La intersección con el PP se produce cuando $z' = 0$ por lo que podemos escribir

$$\begin{aligned} x_p &= x' + \lambda(x_{cp} - x') \\ y_p &= y' + \lambda(y_{cp} - y') \\ z_p &= z' + \lambda(z_{cp} - z') = 0 \end{aligned} \quad (22)$$

de donde $\lambda = -z'/(z_{cp} - z') = z'/(z' - z_{cp})$, de modo que reescribimos las fórmulas como

$$\begin{aligned} x_p &= x' + \frac{z'}{z' - z_{cp}}(x_{cp} - x') \\ y_p &= y' + \frac{z'}{z' - z_{cp}}(y_{cp} - y') \\ z_p &= 0 \end{aligned} \quad (23)$$

Si desarrollamos

$$\begin{aligned} x_p(z' - z_{cp}) &= z' \cdot x_{cp} - x' \cdot z_{cp} \\ y_p(z' - z_{cp}) &= z' \cdot y_{cp} - y' \cdot z_{cp} \end{aligned} \quad (24)$$

Que nos lleva a definir la proyección perspectiva de forma matricial como:

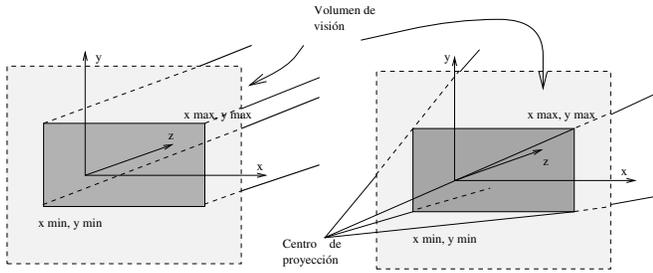


Figura 10: Definición del volumen de visión: la ventana definida en el plano de visión delimita el volumen de visión.

$$\begin{pmatrix} X_p \\ Y_p \\ Z_p \\ W \end{pmatrix} = \begin{pmatrix} -z_{cp} & 0 & x_{cp} & 0 \\ 0 & -z_{cp} & y_{cp} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -z_{cp} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad (25)$$

La forma homogénea $(x_p, y_p, z_p, 1)$ se obtiene al dividir cada componente por W .

$$(x_p, y_p, z_p, 1) = (X_p, Y_p, Z_p, W)/W \quad (26)$$

siendo $W = (z' - z_{cp})$. El uso del factor W no es más que un recurso matemático para poder emplear cálculo matricial en estas transformaciones.

IV-C. Transformación de volumen canónico

Una cámara solamente captura una zona limitada del espacio 3D. Llamaremos **volumen de visión** a la zona de la escena que queremos representar en el plano de visión.

El volumen de visión suele especificarse mediante la definición de una ventana rectangular sobre el plano de visión. Los proyectores que pasan por las esquinas de la ventana son las aristas de un prisma en el caso de la proyección paralela y de una pirámide en el caso de la proyección perspectiva (ver figura 10).

Para delimitar el volumen de visión en la tercera dimensión, empleamos dos planos de corte: el plano cercano (PC) y el plano lejano (PL) (ver figura 11). En el caso de la proyección paralela, el volumen de visión es paralelepípedo. Con perspectiva, el volumen de visión es un tronco de pirámide. Al volumen de visión se le conoce comúnmente con el nombre de *frustum* o pedazo de mundo.

El volumen de visión plantea problemas de corte o *clipping* de objetos. Para simplificarlo, se añade una transformación que convierte el volumen en un paralelepípedo rectangular de lado uno que llamaremos volumen canónico.

Transformar el volumen de visión en un volumen canónico tiene las siguientes ventajas:

- La operación de recorte es directa.
- Facilita los algoritmos de ocultación.
- Lleva implícita la proyección perspectiva.

A la hora de llevar a cabo esta transformación realizamos primero un cambio de orientación del volumen que se aprecia en la figura 12. Este cambio es una deformación donde los valores de las coordenadas x e y se modifican en función

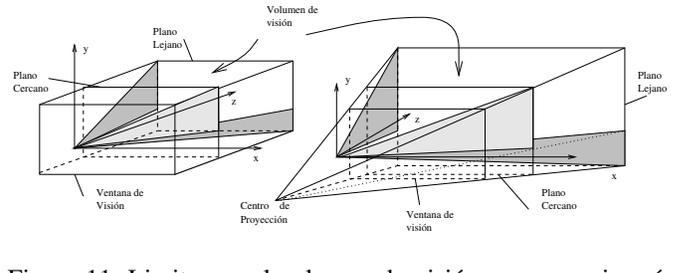


Figura 11: Limitamos el volumen de visión a un espacio más reducido, con el empleo de dos planos de lejanía y de cercanía

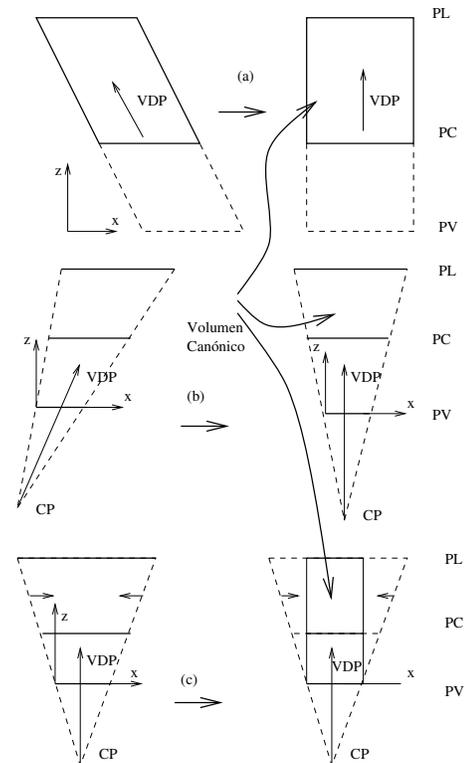


Figura 12: Transformación a volumen canónico en los distintos tipos de proyección

del valor de z . Después se escala el volumen de visión para que tenga un lado unidad. La matriz de la transformación es función del tipo de proyección que queremos realizar.

1. Si la proyección es paralela, la matriz de la transformación es como la empleada para realizar la proyección paralela en el apartado anterior, pero sin anular los valores de z (ver figura 12.a)

$$\begin{pmatrix} x'' \\ y'' \\ z'' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -v_x/v_z & 0 \\ 0 & 1 & -v_y/v_z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = M_P \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad (27)$$

2. Con proyección perspectiva ocurre lo mismo y la matriz es:

$$\begin{pmatrix} x'' \\ y'' \\ z'' \\ W \end{pmatrix} = \begin{pmatrix} -z_{cp} & 0 & x_{cp} & 0 \\ 0 & -z_{cp} & y_{cp} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -z_p \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = M_P \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad (28)$$

Siendo $(x', y', z', 1)$ un punto cualquiera de la escena en el SRV, $(x'', y'', z'', 1)$ el punto transformado, y los valores de v_x, v_y, v_z y de x_{cp}, y_{cp}, z_{cp} definidos como en el tema anterior. En este caso no se pierde el valor de z como ocurría en las ecuaciones 20 y proy 19.

A continuación se convierte el volumen a volumen canónico. Si z_{cerc} es la profundidad del plano cercano, z_{lej} es la profundidad del plano lejano, $x_{min}, y_{min}, x_{max}, y_{max}$ son los límites de la ventana de visualización, entonces las transformaciones que hemos de aplicar son

- $T(0, 0, -z_{cerc})$ para posicionar el plano cercano en el plano de visión.
- $T(-x_{min}, -y_{min}, 0)$ para posicionar la ventana en centro de referencia del volumen canónico.
- $S(\frac{1}{x_{max}}, \frac{1}{y_{max}}, \frac{1}{z_{lej}})$ para hacer que los lados valgan la unidad.

En forma matricial tendríamos:

$$\begin{pmatrix} x''' \\ y''' \\ z''' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{x_{max}} & 0 & 0 & -x_{min} \\ 0 & \frac{1}{y_{max}} & 0 & -y_{min} \\ 0 & 0 & \frac{1}{z_{lej}} & -z_{cerc} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ z'' \\ 1 \end{pmatrix} = M_{VC} \begin{pmatrix} x'' \\ y'' \\ z'' \\ 1 \end{pmatrix} \quad (29)$$

Cuando los puntos han sido llevados al volumen canónico, la proyección a realizar sobre el plano de visión es siempre la paralela ortogonal anulando la componente z . El valor z conviene no perderlo para usarlo en los algoritmos que necesitan conocer la profundidad.

V. TUBERÍA DE VISTA EN 2D

El caso en dos dimensiones es más sencillo porque el problema de posicionar la cámara y de definir el volumen de visión no existen. Sin embargo, desde que definimos nuestros objetos gráficos en 2D hasta que aparecen en una zona de nuestro visualizador, se realizan una serie de transformaciones sobre los puntos de los objetos. El conjunto de transformaciones se denomina tubería de vista y a ella dedicamos la presente sección.

Cuando definimos los objetos 2D, lo hacemos especificando las coordenadas de los puntos que lo componen. Las coordenadas con las que definimos los objetos se denominan *coordenadas de trabajo* (CT). Estas coordenadas no tienen por qué ser necesariamente coordenadas cartesianas sino que pueden ser de otro tipo como polares.

Los sistemas de representación gráfico suelen trabajar con coordenadas cartesianas. Ya hemos visto que las matrices de transformación están definidas para este tipo de coordenadas en concreto. De este modo, debemos traducir las *coordenadas de trabajo* en *coordenadas mundo* (CM). Normalmente solemos trabajar directamente en *coordenadas mundo* que se definen como aquellas coordenadas cartesianas en las que está definida la imagen.

De toda la escena que hemos creado en coordenadas mundo seleccionamos el fragmento rectangular que queremos visualizar. Este fragmento es lo que se denomina **ventana**. La ventana viene definida por las coordenadas de la esquina inferior izquierda y las de la esquina superior derecha en coordenadas mundo. La parte de la imagen que no está dentro de la ventana no se representa.

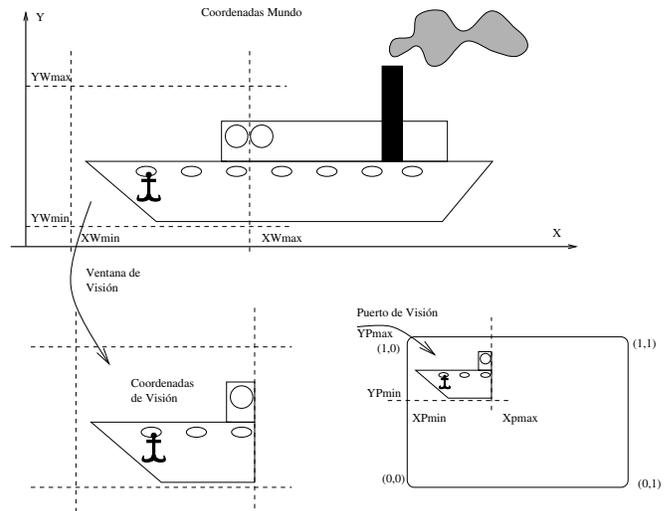


Figura 13: Concepto de puerto y ventana en la tubería de transformación 2D.

A veces podemos definir una ventana con otra orientación diferente. En este caso, podemos definir un sistema de referencia de acuerdo a la nueva orientación dentro del sistema de coordenadas mundo y después definir un rectángulo. Las nuevas coordenadas se denominan *coordenadas de vista*.

En la figura 13, vemos que la ventana de visión se define con los puntos de la esquina inferior izquierda (xw_{min}, yw_{min}) y la esquina superior derecha (xw_{max}, yw_{max}) . Si llamamos a los puntos del objeto en coordenadas mundo (p_x, p_y) , y a los puntos en coordenadas de visión (p'_x, p'_y) , si posicionamos el centro del sistema de referencia de visión en la esquina inferior izquierda de la ventana (xw_{min}, yw_{min}) , bastaría con hacer una traslación del nuevo sistema de referencia al original. Si además queremos que las nuevas coordenadas sean coordenadas normalizadas, debemos realizar una escala en ambos ejes, resultando:

$$\begin{pmatrix} p'_x \\ p'_y \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{S_x} & 0 & 0 \\ 0 & \frac{1}{S_y} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -xw_{min} \\ 0 & 1 & -yw_{min} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} \quad (30)$$

con $S_x = xw_{max} - xw_{min}$ y $S_y = yw_{max} - yw_{min}$. Ahora sólo falta especificar la zona de la pantalla en la que queremos que se visualice la imagen o **puerto de visión** o simplemente puerto. La pantalla de los dispositivos se referencia en coordenadas normalizadas de dispositivo. Así, la esquina inferior izquierda se corresponde con el punto (0,0) y la esquina superior derecha se corresponde con el punto (1,1). Es necesario normalizar estas coordenadas para hacer que la representación en los dispositivos sea independiente del tipo de dispositivo empleado. El puerto de visión se especifica en *coordenadas normalizadas de dispositivo* detallando la esquina inferior izquierda y la esquina superior derecha del puerto.

Así pues, se establece otra transformación desde *coordenadas de vista* en *coordenadas normalizadas de vista* (CNV). Sabemos que el rectángulo definido con la ventana, debe representarse en el rectángulo definido en el puerto. Esto

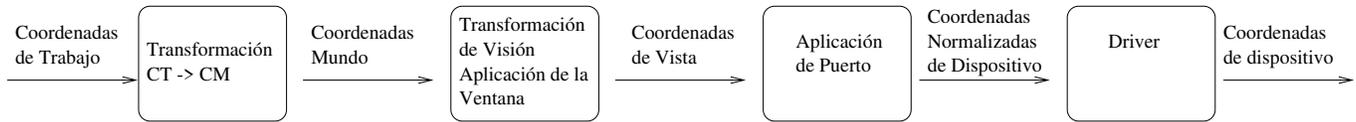


Figura 14: Tubería de transformación de vista bidimensional.

supone una nueva transformación para crear las *coordenadas normalizadas de vista*.

De forma parecida al desarrollo que presentamos para convertir puntos en coordenadas mundo y coordenadas de visión, definimos el puerto con los puntos de la esquina inferior izquierda (xp_{min}, yp_{min}) y la esquina superior derecha (xp_{max}, yp_{max}) . Si llamamos a los puntos del objeto de visión (p'_x, p'_y) , y a los puntos en coordenadas normalizadas de dispositivo (p''_x, p''_y) , hay que aplicar una nueva escala para ajustarse al tamaño del puerto, y después trasladar los puntos a la esquina inferior izquierda del puerto (xp_{min}, yp_{min}) . Estas serían las ecuaciones a aplicar:

$$\begin{pmatrix} p''_x \\ p''_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & xp_{min} \\ 0 & 1 & yp_{min} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} xp_{max} - xp_{min} & 0 & 0 \\ 0 & yp_{max} - yp_{min} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p'_x \\ p'_y \\ 1 \end{pmatrix} \quad (31)$$

Pero los visualizadores no emplean *coordenadas normalizadas de vista* sino que cada uno referencia los puntos de la pantalla con su propio sistema. Por ejemplo una pantalla del ordenador numera los pixels de izquierda a derecha y de arriba abajo. Es necesario pues especificar una nueva transformación para obtener *coordenadas de dispositivo*. Esta transformación es realizada directamente en los *drivers de dispositivo*. Expresar esta transformación de forma matricial tiene su aquel porque los dispositivos numeran los pixels de izquierda a derecha y de abajo a arriba, siendo la fila cero la fila superior. En un monitor de W pixels de ancho y de H pixels de alto, primero aplicamos la escala, después un reflejo en el eje de ordenadas y finalmente desplazamos el centro de coordenadas desde la esquina inferior izquierda hasta la esquina superior izquierda.

$$\begin{pmatrix} p'''_x \\ p'''_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & H \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} W & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p''_x \\ p''_y \\ 1 \end{pmatrix} \quad (32)$$

La figura 14 esquematiza todo el proceso que acabamos de comentar. Las coordenadas mundo se corresponde con (p_x, p_y) , las coordenadas de vista con (p'_x, p'_y) , las coordenadas normalizadas de dispositivo con (p''_x, p''_y) , y por último, las coordenadas de dispositivo ya en pixels como $(round(p'''_x), round(p'''_y))$, necesario porque los pixels se numeran con valores enteros y p'''_x y p'''_y son números reales.

MÁS INFORMACIÓN

Los libros básicos de referencia sobre informática gráfica, todos suelen incluir un capítulo o varios sobre transformaciones geométricas. Por ejemplo, Hearn dedica el capítulo 6 en [1] y [2] y el capítulo 7 en [3] a este asunto. En todos los casos la formulación es muy parecida a la seguida es este documento.

El capítulo 6 de [4] se aproxima mucho a la aproxima seguida en este texto.

Las distintas ediciones de Foley también tratan el asunto en profundidad en el capítulo 6 de [5] y [6] y en el capítulo 12 de [7].

Gortler ofrece una visión particular en el capítulo 10 de [8]. Se trata de un enfoque que tiene en cuenta el ángulo de apertura de la cámara, mucho más cercano a las necesidades de las cámaras empleadas en los programas de modelado 3D.

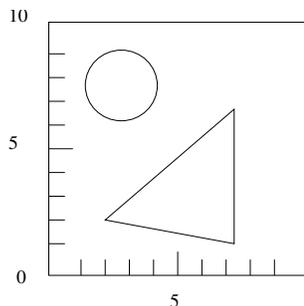
Los conceptos de puerto y ventana 2D aparecen explicados en detalle en [9] con ampliaciones sobre la problemática del *clipping* o recorte de puntos que quedan fuera de la ventana de visión.

CUESTIONES Y PROBLEMAS

1. Enumere los parámetros de sistema de referencia de visión 3D.
2. Desarrolle el cálculo de la matriz empleada para pasar los puntos de una escena de coordenadas mundo a coordenadas de visión. ¿Por qué hay que aplicar esta transformación para hacer el rendering?
3. ¿Qué es *volumen de visión* y cómo se define?
4. ¿Qué es *plano lejano* y *plano cercano*?
5. ¿Qué parámetros hay que establecer para fijar la posición de la cámara en la transformación de visión 3D?
6. Tenemos una vista de un objeto 3D definida por una Ventana de Visión (VV) y un Centro de Proyección (CP). Si acercamos CP al objeto, ¿este se verá más grande o más pequeño? Razone su respuesta.
7. Indique los elementos necesarios para definir el volumen de visión 3D en una proyección en perspectiva.
8. En la representación de objetos en 3D, distinguimos entre *coordenadas mundo* y *coordenadas locales*. ¿En qué se diferencian?
9. ¿Qué es el *grafo de escena* y para qué sirve?
10. Indique las diferencias entre el *puerto de visión* y la *ventana de visión*.
11. Indique el papel que juegan la ventana de visión y el centro de proyección en la proyección de visión.
12. Tiene que proyectar el segmento **AB**, definido por los puntos **A** = (4, 5, 8) y **B** = (7, 9, 12) en el plano $z = 2$ (proyección paralela normal a dicho plano) en una ventana definida por los puntos **W_{min}** = (3, 4, 2) y **W_{max}** = (6, 8, 2) y con un plano cercano $z = 6$ y plano lejano $z = 10$. Calcule los puntos **A'** y **B'** resultado de dicha proyección.
13. Queremos proyectar el segmento definido por los puntos **P1** = (4, 5, 3) y **P2** = (0, -3, 4) en el plano $z = 0$. Obtener por un lado los puntos proyectados resultantes

$P1'$ y $P2'$ empleando una proyección perspectiva cuyo centro de proyección sea $CP = (1, 2, -3)$, y los puntos resultantes $P1''$ y $P2''$ empleando una proyección paralela de dirección $d = (0, 0, 1)$. Emplee en todo momento coordenadas mundo.

14. Queremos proyectar un cubo definido por los puntos $(0, 0, 0)$, $(1, 0, 1)$, $(1, 0, 0)$, $(0, 0, 0)$, $(0, 1, 0)$, $(1, 1, 1)$, $(1, 1, 0)$, $(0, 1, 0)$ en el plano $z = 4$. Dibuje el resultado empleando una proyección paralela en la dirección $(1, 25, 1, 25, 2)$. Emplee en todo momento coordenadas mundo.
15. Queremos proyectar el segmento definido por los puntos $(1, 0, 1)$ y $(0, 1, 0)$ en una esfera de centro $(3, 2, 1)$ y de radio 2 empleando una proyección perspectiva de centro de proyección $(-2, -2, -2)$. Indique cómo resolvería el problema un ordenador.
(La ecuación de una esfera de radio r y centrada en (a, b, c) es $(x - a)^2 + (y - b)^2 + (z - c)^2 = r^2$)
16. Queremos proyectar el segmento definido por los puntos $P1 = (1, 0, 1)$ y $P2 = (0, 1, 0)$ en el plano definido por la ecuación $Ax + By + Cz + D = 0$ con una proyección perspectiva de centro de proyección $CP = (-1, -1, -1)$. Indique las coordenadas de los puntos proyectados.
17. Resuelva el ejercicio anterior considerando que utilizamos un *plano lejano* y un *plano cercano* en $z = 0,5$ y $z = 2$ respectivamente.
¿y con el plano cercano es $z = 7$ y el plano lejano $z = 11$?
¿y con el plano cercano es $z = -7$ y el plano lejano $z = 11$?
18. Realice la proyección del segmento definido por los puntos $A = (3, 5, 8)$ y $B = (5, 3, 10)$ sobre el plano $z = 5$ utilizando una ventana definida por los puntos $W_{\min} = (3, 6, 5)$ y $W_{\max} = (6, 8, 5)$ y con un plano cercano $z = -1$ y plano lejano $z = 2$. Proyección paralela plana.
19. El volumen de visión para proyectar una escena 3D ha quedado limitado por las ventanas que definen puntos $(-1, -1, 1)$ y $(1, 1, 1)$ en el plano cercano y $(-2, -2, 2)$ y $(2, 2, 2)$ en el plano lejano. ¿Cuál es el centro de proyección? Justifique su respuesta.
20. Disponemos de la imagen representada en la figura. Queremos representar la ventana definida por $(0, 1)$ $(7, 9)$ en el puerto $(0, 0, 25)$ $(0, 5, 1)$. Representar la figura que visualizamos.



21. Tenemos que proyectar la figura de una

pequeña casa 3D definida por los vértices $(0, 0, 0)$, $(0, 1, 0)$, $(0, 5, 1, 5, 0)$, $(1, 1, 0)$, $(1, 0, 0)$ y $(0, 0, 2)$, $(0, 1, 2)$, $(0, 5, 1, 5, 2)$, $(1, 1, 2)$, $(1, 0, 2)$. Calcule los puntos proyectados empleando una proyección paralela en la dirección $(1, 0, 1)$ sobre el plano $z=4$. Dibuje el resultado.

- Calcule los puntos proyectados empleando una proyección en perspectiva empleando el centro de proyección $(0, 5, 0, -2)$ sobre el plano $z=-1$. Dibuje el resultado.
22. Relacione los pasos fundamentales de la transformación de visión con el objeto cámara de un software como 3D Studio.
23. Para una ventana definida por los puntos $(1, 1)$ y $(2, 2)$ encontrar el segmento que resulta del recorte de la recta $(2, 2)$ $(0, 1)$. Para ello emplear los algoritmos de recorte estudiados.
24. Aplicando el algoritmo de Básico Incremental calcular los puntos del segmento de recta definida por los puntos $(4, 5)$ $(10, 8)$.
25. Aplicando el algoritmo de Bresenham calcular los puntos del segmento de recta definida por los puntos $(4, 5)$ $(10, 8)$.
26. Las fórmulas empleadas para calcular puntos con el algoritmo de Bresenham sólo valen cuando la pendiente está entre 0 y 1. Establecer las fórmulas cuando las pendientes tienen valores distintos.
27. Calcular los puntos de una circunferencia de centro $(-1, 3)$ y de radio 3.
28. Para el polígono de puntos $(1, 1)$, $(4, 5)$, $(6, -1)$, $(4, 2)$, $(1, -3)$, determinar los puntos de relleno.

REFERENCIAS

- [1] D. Hearn and M. P. Baker, *Computer Graphics, C Version*, segunda ed. Prentice Hall, 1997.
- [2] —, *Gráficas por Computadora*, segunda ed. PHH Prentice Hall, 1995.
- [3] D. Hearn, M. P. Baker et al., *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall,, 2004.
- [4] P. Shirley, *Fundamentals of Computer Graphics*. Natick, MA, USA: A. K. Peters, Ltd., 2002.
- [5] J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips, *Introduction to Computer Graphics*. Addison-Wesley Publishing Company, 1994.
- [6] J. D. Foley, A. Van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics Principles and Practice. Second Edition in C*. Addison-Wesley Publishing Company, 1996.
- [7] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [8] S. J. Gortler, *Foundations of 3D computer graphics*. MIT Press, 2012.
- [9] Z. Xiang and R. Plastock, *Computer Graphics*, segunda ed. Schaum's, 2000.

Fuentes de iluminación

ÍNDICE

I. Aspectos básicos sobre iluminación	2	Las fuentes de iluminación son un integrante esencial de las escenas 3D. En este artículo mostramos la utilidad de estos elementos en la representación de escenas. Se presentan los distintos tipos de fuentes de iluminación a través de las diferentes características que las definen. Además, explicamos cómo se modelan las fuentes de luz en informática gráfica. Finalizamos presentando diferentes configuraciones típicas de iluminación de escenas que tienen en cuenta tanto la posición de las fuentes de luz como su tipo.
I-A. Calidad de la luz	2	
I-B. Dirección de la luz	2	
I-C. Cobertura de la luz	3	
I-D. Intensidad de la iluminación	3	
I-E. Color	3	
II. Tipos de fuentes de luz	3	Si somos capaces de ver objetos en el mundo real es porque hay fuentes de iluminación. La energía que se genera en las fuentes de iluminación, sean naturales como el sol, sean artificiales como las bombillas, permite que la luz se refleje en los objetos y llegue hasta nuestro sistema de visión. Sin esas fuentes de iluminación no veríamos escenas, sólo oscuridad. De igual manera, en informática gráfica, para poder visualizar los modelos 3D, necesitamos fuentes de iluminación que pueden estar representadas de forma explícita dentro del grafo de escenas o de forma implícita con valores de iluminación global. En este tema nos centramos en presentar la representación explícita de fuentes de luz dentro de escenas 3D y su uso adecuado.
II-A. Modelo radial	4	
II-B. Focos	4	
II-C. Luz direccional	4	
II-D. Área o volumen de luz	5	
II-E. Redes fotométricas	5	
III. Modelo de una fuente de luz	5	Primero presentamos una serie de aspectos básicos sobre iluminación que determinan los tipos de fuentes de iluminación (sección I). Después nos centramos en presentar los tipos de fuentes de luz que ofrecen los programas informáticos de modelado de escenas 3D (sección II). Continuamos apuntando cuáles son los parámetros de las fuentes de iluminación que deben ser modelados computacionalmente para que puedan ser incluidas como un componente en los sistemas de modelado 3D (sección III). Completamos el tema presentando algunas claves para la iluminación efectiva de escenas, tanto de interior como de exterior (secciones IV y V).
III-A. Parámetros básicos	5	
III-B. Atenuación	6	
III-C. Direccionalidad de la fuente de luz	6	
III-D. Medidas de intensidad	7	
III-E. Temperatura y color de la fuente de luz	7	
III-F. Suma de efectos y saturación	8	
IV. Iluminación de escenas de interior	8	
IV-A. Iluminación de estudio fotográfico	8	
IV-B. Claroscuro	8	
IV-C. Contraluz	8	
IV-D. Iluminación de tres puntos	9	
IV-E. Iluminación de ventana	9	
IV-F. Iluminación con lámparas y luminarias	9	
V. Iluminación de exteriores	9	
V-A. Componentes	9	
V-B. Importancia de la hora del día	10	
Referencias	11	

Figura 1: Diferencia entre una fuente de luz dura y una fuente blanda.

Figura 2: Diferencia entre una fuente de luz dura y una fuente blanda.

I. ASPECTOS BÁSICOS SOBRE ILUMINACIÓN

Distinguiamos cinco aspectos fundamentales a la hora de tipificar las fuentes de iluminación en un escenario de iluminación. Estos cinco aspectos son:

Calidad de la luz : que nos permite distinguir entre luces duras o blandas con importantes implicaciones en el tipo de sombra proyectada.

Dirección de la luz : que nos permite tener en cuenta la posición de la fuente de luz y la dirección de emisión de la luz con respecto a los objetos que componen la escena.

Cobertura de la luz : que tiene en cuenta el espacio sobre el cual la fuente de luz tiene algún efecto.

Intensidad de la luz : que permite controlar la energía luminosa que emite la fuente de luz.

Color : que permite cambiar el color de la luz emitida aportando importantes connotaciones en la escena.

En las siguientes subsecciones desarrollamos cada uno de estos aspectos.

I-A. Calidad de la luz

Distinguiamos entre luces duras y luces blandas (ver figura 1). Estos dos tipos de fuentes de luz producen un efecto diferente en la iluminación y en la sombra proyectada. Cada una de ellas tiene un uso diferente que conviene conocer.

I-A1. Luz dura: La luz dura se consigue empleando fuentes de luz compactas. Como consecuencia son fáciles de controlar restringiendo su efecto y enfocando el haz con precisión.

Al incidir sobre los objetos, permiten definir la geometría de los mismos de forma clara. Las sombras son duras, y pueden utilizarse para resaltar los perfiles y las texturas de los objetos.

El hecho de que aparezcan sombras bien definidos puede ser un problema cuando las geometrías son toscas o cuando las sombras se proyectan sobre el mismo objeto porque oscurecen partes del mismo que pueden ser importantes.

Por todo esto, este tipo de fuentes son útiles como fuentes de luz principal. Como fuentes de luz de relleno, también pueden aportar alguna ventaja por su fácil control, pero al arrojar sombras adicionales son descartadas en beneficio de las luces blandas.

I-A2. Luz blanda: Las fuentes de luz blanda se consiguen empleando fuentes voluminosas. Su radio de actuación es difícil de controlar y no son fáciles de focalizar.

A diferencia de lo que ocurre con las fuentes duras, la iluminación es más gradual aportando más tonos en la superficie de los objetos. No hay sombras duras, con lo que no se originan perturbaciones ni resaltes de la textura en los objetos.

Figura 3: Coordenadas empleadas para posicionar la cámara.

Al desaparecer las sombras duras, tienen el inconveniente de que el modelo geométrico puede desaparecer por completo. En entornos muy iluminados, su efecto se pierde por completo.

I-B. Dirección de la luz

Para posicionar las fuentes de luz se utiliza un sistema de coordenadas polares en unidades horarias que tiene en cuenta tanto la posición del objeto a iluminar como la posición de la cámara (ver figura 21). Las fuentes de luz se referencian con respecto a la posición de la cámara, de manera que ésta siempre está en las coordenadas 6H/3V. El eje que va desde la cámara al objeto a iluminar determina los planos en los que se fijan los volantes de coordenadas. El volante horizontal y vertical son perpendiculares entre sí.

Figura 4: Connotación aportada por la dirección de la fuente de luz. El mismo personaje puede pasar de tener un aspecto afable a tener un aspecto terrible.

Como se muestra en la figura 4, la dirección de la luz aporta una connotación clara a la presentación final. Algunas direcciones tienen nombre debido a su uso masivo en fotografía. A continuación describimos algunos de estos tipos de iluminación.

I-B1. Iluminación frontal: En la iluminación frontal la fuente de luz se sitúa en la posición de la cámara, esto es en la posición 6H/3V. La iluminación frontal hace que no se proyecten sombras sobre la superficie iluminada, evitando así que se muestren detalles sobre la misma. Es un recurso muy empleado en retratos porque elimina las imperfecciones de la piel como pueden ser las arrugas (ver figura 5). Las escenas resultantes son muy planas, con ausencia de relieves y detalles.

I-B2. Iluminación lateral: En la iluminación lateral o con ángulo la fuente de iluminación se aleja de la coordenada 6H tomando valores mayores que 3H o menores de 9H y próximos a ellos. Este tipo de iluminación es útil para hacer aparecer detalles de la superficie. Cuanto más próxima a 3H o 9H esté la posición del foco, más resaltadas quedarán las texturas. La contrapartida es que, cuando las luces son duras, una parte de la superficie quedará en sombra. La figura 5 muestra un ejemplo de su uso. Este tipo de iluminación es útil para hacer juegos y efectos con las sombras y crear claroscuros.

I-B3. Contraluz: En un contraluz el objeto está entre la cámara y la fuente de iluminación. La fuente de luz ocuparía entonces la posición 12H/9V. Normalmente sólo iluminarán los bordes de los objetos con lo cual es muy útil para mostrar siluetas. Los objetos traslúcidos se suelen ver muy favorecidos por este tipo de iluminación. Al iluminar los bordes del objeto, suele ser muy efectiva en objetos con una textura vellosa como las cabezas con pelo o los objetos de tela o terciopelo. La figura 5 muestra algunos ejemplos.

Figura 5: Ejemplos de iluminación frontal (arriba a la izquierda); de iluminación lateral (arriba centro y derecha); de contraluz (abajo a la izquierda); y de iluminación inferior y superior (abajo centro y derecha).

Figura 6: Ejemplos de cobertura de luz. Al cambiar la zona cubierta por el foco, podemos hacer que la atención del observador se centre de diferentes detalles.

I-B4. Iluminación desde arriba o desde abajo: En este caso, cambiamos la altura de fuente de luz (hasta 12V o 6V en los casos extremos). Cuando se aplican sobre personas se generan efectos de misterio muy explotados en las películas del género de terror. La iluminación cenital (12V) se emplea en iluminación de objetos en interior cuando el área de la fuente de iluminación es grande en comparación con el tamaño del objeto. Desarrollaremos este tipo de iluminación en la sección IV-A.

I-C. Cobertura de la luz

La cobertura de una fuente de luz hace referencia al espacio que es iluminado por dicha fuente de luz. Muchos focos permiten apuntar el haz de luz hacia una zona determinada de la escena y dispersar o concentrar más o menos dicho haz de luz.

La cercanía de la fuente de luz a los objetos es uno de aspectos que determina la cobertura de la fuente de luz, de tal manera que cuanto más alejada esté la fuente de luz, mayor cobertura. La cobertura puede cambiarse también empleando solapas que acotan la zona de la escena afectada por la fuente.

Acotar la cobertura nos puede permitir realzar algún detalle de la escena iluminada apuntando el haz de luz hacia la zona concreta que nos interese realzar. También podemos acentuar los relieves dirigiendo la fuente de iluminación sobre pantallas u objetos que están por detrás del objeto. La figura 6 ilustra estos usos.

I-D. Intensidad de la iluminación

La intensidad de la iluminación se refiere a la cantidad de luz que hay en la escena. Se debe evitar presentar escenas demasiado oscuras (subexpuestas) y también presentar escenas con demasiada iluminación (sobreexpuestas). En ambos casos el control deficiente de la luz impide ver detalles importantes en la escena iluminada (ver figura 7).

El hecho de que la escena esté más o menos iluminada depende de la intensidad y número de fuentes de luz y también del tiempo de exposición. Si exponemos el sensor de iluminación de la cámara durante más tiempo del debido a la luz de la escena, obtenemos imágenes sobreexpuestas. El control de exposición es un término que se deriva de

Figura 7: Ejemplos de variaciones de intensidad luz. Arriba mostramos la diferencia entre una escena subexpuesta, sobreexpuesta (a la derecha) o iluminada correctamente. Abajo mostramos la variedad posible con una intensidad adecuada en todos los casos.

la fotografía tradicional, pero los algoritmos de informática gráfica también pueden implementar este recurso.

Cuando la escena está iluminada adecuadamente, la intensidad aporta información fundamentalmente del contexto en el que está el objeto.

La manera de conseguir más o menos intensidad de iluminación depende de si estamos hablando de luz natural o luz artificial. Con la luz artificial podemos acercar los objetos a la fuente de iluminación, regular su potencia luminosa (cuando sea posible) o añadir o eliminar fuentes de luz. La intensidad luminosa en un punto de la escena es inversamente proporcional al cuadrado de la distancia. Cuando es luz natural podemos buscar zonas de sombra y utilizar cortinas o reflectores cuando sea posible.

Valores bajos de intensidad de la iluminación hacen que objetos que nuestro sistema de visión considera que son blancos, en realidad sean grises. Las cámaras fotográficas y los programas de edición digital permiten ajustar el *balance de blanco* para que podamos establecer las zonas de la escena cuyo color debería ser considerado blanco. Una vez fijada esta referencia, el resto de tonos de la escena se adapta.

I-E. Color

El color de las fuentes de iluminación se asume que, por defecto, es blanco. Emplear fuentes de luz con un determinado color suele aportar una connotación a la escena que no podemos ignorar. Por ejemplo una luz roja suele asociarse a situaciones de peligro.

Incluso cuando la fuente de luz es blanca, hay que tener en cuenta que puede haber matices a tener en cuenta. En el caso de la luz natural, la temperatura del emisor hace que luz blanca tome un tono más anaranjado (luz fría) o azulado (luz caliente). También el entorno puede condicionar el color de la luz ambiente, ya que los múltiples reflejos de la luz en el entorno hacen que ésta pueda adoptar diferentes tonos. Tal es el caso de una imagen tomada en el interior de un bosque, donde la tonalidad dominante será el verde por el reflejo de la luz en las hojas de los árboles.

En luz artificial se emplean bolsas con gel de color, que, atravesadas por la luz emitida por la fuente permiten fotografiar escenas muy llamativas.

II. TIPOS DE FUENTES DE LUZ

Los programas de modelado 3D ofrecen diferentes tipos de fuentes de luz para iluminar las escenas. En esta sección describimos los cinco tipos más frecuentes que son las luces omnidireccionales o *point lights*, los focos o *point lights*, las



Figura 8: Modelo de fuente de luz puntual y ejemplo de efectos.

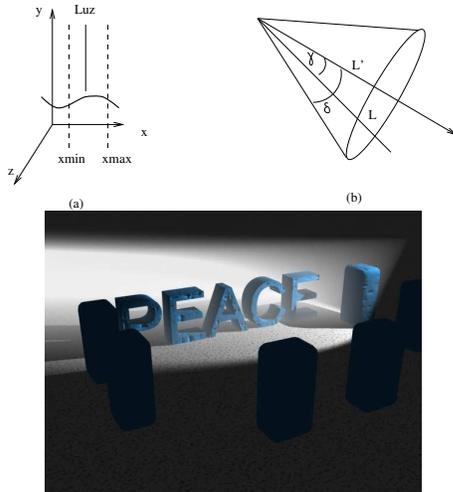


Figura 9: Modelo de fuente de luz spotlight y ejemplo de efectos.

luces direccionales, las luces de tipo área o volumen y las luces modeladas con diagramas goniométricos.

II-A. Modelo radial

En el modelo radial o puntual u omnidireccional o de luz isotrópica o *spotlight*, la fuente de luz emite en todas las direcciones a partir de un punto prefijado (ver figura 8).

Se modelan estableciendo su posición \bar{L} en sistema de coordenadas mundo y un valor de intensidad I o de color. Estas fuentes de luz quedan bien representadas con un punto desde el que emanan radios de luz de igual intensidad en todas las direcciones.

II-B. Focos

En los focos o *spotlight*, la fuente de luz emite a partir de un punto pero las direcciones en las que se emita luz se acotan, generalmente al interior de un cono. Se limita pues la cobertura de la fuente de iluminación (ver figura 9).

Estas fuentes de luz se modelan dando su posición \bar{L} , la dirección a la apuntan \bar{L}' y un ángulo de apertura del cono. En este tipo de fuentes de luz, la intensidad se suele hacer depender del ángulo con respecto a \bar{L}' , de manera que la intensidad es máxima en los puntos de la escena alineados



Figura 10: Modelo de fuente de luz direccional y ejemplo de efectos.



Figura 11: Modelo de fuente de luz de tipo área y ejemplo de efectos.

con \bar{L}' , decreciendo paulatinamente hasta desaparecer en los puntos que están fuera del espacio de cobertura.

II-C. Luz direccional

En el modelo de luz direccional la fuente de luz emite en una única dirección. A diferencia de los modelos anteriores, la fuente de iluminación no emite rayos de luz de forma radial desde un punto dado. En este caso, los rayos de luz emanan en una única dirección (ver figura 10).

También se permite acotar el volumen de acción de la fuente de iluminación. En este caso, en lugar de utilizar un cono se emplea un cilindro o un prisma, de manera que la cobertura de la luz queda reducida al volumen interior de dicho cilindro o de dicho prisma.

Se modelan dando su posición \bar{L} , la dirección a la apuntan \bar{L}' y una apertura o zona de actuación: prisma o cilindro. Al igual que sucede con las luces *spotlight*, también aquí podemos hacer que la intensidad sea máxima en los puntos alineados con \bar{L}' y que vaya decreciendo paulatinamente conforme se aleja de dicha dirección en el volumen de cobertura.

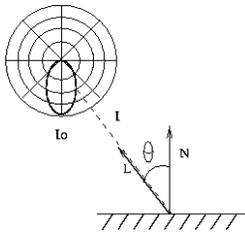


Figura 12: Ejemplos de diagramas goniométricos.

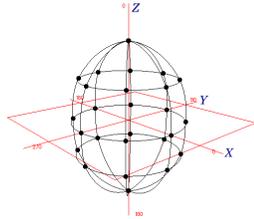


Figura 13: Modelos de luz de red en tres dimensiones. A la izquierda se muestra el proceso de obtención de valores en diferentes direcciones del espacio. A la izquierda la representación de una red fotométrica en 3D (imagen propiedad de Autodesk 3DS).

II-D. Área o volumen de luz

En este tipo de fuentes de iluminación, se permite dar una forma al emisor de luz estableciendo el área o volumen que ocupa. Este tipo de fuentes de luz son necesarios para representar determinados emisores como pueden ser tubos fluorescentes o incluso ventanas, donde la luz no se emite desde un único punto sino que se distribuye a lo largo de una superficie o volumen (ver figura 11).

Este tipo de fuentes de iluminación se suele implementar distribuyendo diferentes fuentes de luz omnidireccionales o direccionales a lo largo de la superficie o volumen previamente establecido. Aparte de la posición en la que se colocan \vec{L} y la dirección hacia donde apunta (en caso de no utilizarse emisores omnidireccionales), es importante establecer la distribución de emisores a lo largo de la superficie, esto es, indicar cuántos emisores hay y qué posiciones ocupan dentro de la superficie o volumen.

II-E. Redes fotométricas

Una luz de red fotométrica representa de forma explícita la distribución de la intensidad luminosa en las distintas direcciones que emanan de la posición en la que se ubica la fuente. Se emplean datos obtenidos empleando dispositivos fotométricos que miden la cantidad de luz que hay en diferentes puntos alrededor del emisor. Esta información se almacena en ficheros formato IES que después son empleados para calcular la intensidad luminosa que afecta a los distintos puntos del espacio de la escena. Este fichero sirve para calcular intensidad I en función del vector \vec{L} .

Un diagrama goniométrico es una representación visual de la distribución de la intensidad de la fuente de luz según el ángulo vertical. Con los diagramas goniométricos modelamos

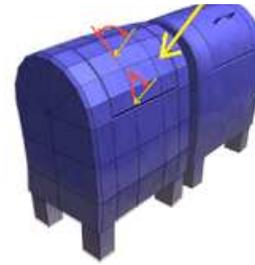


Figura 14: El efecto de la fuente de luz en las superficies de la escena depende de la posición relativa de la fuente de luz con respecto a tales superficies (imagen propiedad de Autodesk 3DS).

la distribución espacial de la intensidad de la fuente de luz. A partir de un punto central, que representa la posición del emisor se posiciona una red en forma de tela de araña (en inglés *web*). Sobre esta red se traza una curva que representa la intensidad luminosa en torno al emisor. El ángulo determinado por el vector \vec{L} permite establecer el valor de la intensidad I_θ calculando el punto de intersección con la curva (ver figura 12). La intensidad luminosa emitida por una fuente se mide en dicho diagrama como I_θ . En cada punto consideramos un valor de iluminación por parte de una fuente determinada, que depende del ángulo de incidencia de dicha fuente luminosa. La curva del diagrama goniométrico puede ofrecer valores absolutos de intensidad luminosa como en el diagrama de la bombilla de la parte derecha o unidades relativas, en forma de porcentaje como en el diagrama de la izquierda: en el centro del diagrama la intensidad es 0 y en el límite es el 100% de la intensidad I_0 .

La distribución de la intensidad luminosa es un campo es un espacio tridimensional. Un diagrama goniométrico es una capa en dicho campo. La red fotométrica mostrada en la figura 12, puede extenderse a una tercera dimensión como se muestra en la figura 13. En este caso, no sólo se tendrá en cuenta el ángulo vertical para conocer el valor de la intensidad de la fuente de luz a tener en cuenta sino que todas las coordenadas de posición de cada punto del espacio son importantes para conocer el valor de intensidad de la fuente de luz que afecta en dicho punto.

III. MODELO DE UNA FUENTE DE LUZ

En esta sección primero presentamos cuáles son los parámetros básicos empleados para modelar las fuentes de iluminación. Después explicaremos cómo se consigue hacer que las fuentes de iluminación puedan ser direccionales, esto es, que apunten hacia una dirección dada; cómo se modela el hecho de que la intensidad luminosa en un punto dado de la escena disminuya con la distancia a la fuente de iluminación; el color o distribución espectral de la fuente de iluminación y el efecto de la saturación o sobreexposición.

III-A. Parámetros básicos

Como hemos visto en la sección anterior, independientemente del tipo de luz empleado, hay dos parámetros que

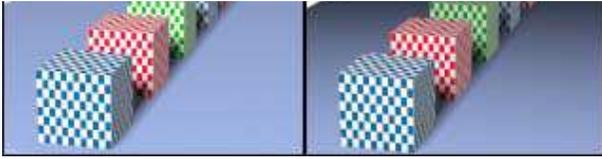


Figura 15: Atenuación de la intensidad luminosa con la distancia (imagen cortesía de Autodesk 3DS).

siempre se tienen en cuenta a la hora de modelar la fuente de iluminación: la **intensidad** I y su **posición** \vec{L} en el espacio. Es importante hacer notar en este momento que, a la hora de conocer los efectos de la fuente de luz en la escena, los valores de intensidad y de posición que realmente importan son los valores relativos con respecto a los puntos de la escena que son iluminados. Así, en el caso de la intensidad, hemos visto que los diagramas goniométricos permiten representar el hecho de que la luz no se expande de igual manera en todas las direcciones en muchas fuentes de luz. Con respecto a la posición, también es importante tener en cuenta que, una vez posicionada la luz en la escena, será importante obtener un vector que representa la dirección de incidencia \vec{L} y que depende del punto de la escena que se esté analizando en cada momento.

Por ejemplo en el modelo de reflexión difusa, el efecto de una fuente de luz en un plano dado de una superficie concreta de la escena se computa como:

$$I = I_p k_d (\vec{N} \cdot \vec{L}) \quad (1)$$

Donde I_p es la intensidad de la fuente de luz, k_d depende del material o del color de la superficie, y el producto $(\vec{N} \cdot \vec{L})$ representa el ángulo relativo entre la orientación de la superficie (representado por el vector normal \vec{N} a la misma) y la dirección de incidencia de la luz \vec{L} . Cuando \vec{L} y \vec{N} están alineados el producto es igual a 1 y la iluminación de la superficie es máxima. Cuando \vec{L} y \vec{N} son perpendiculares el producto es igual a 0, lo que implica que la fuente de luz no afecta en la iluminación de la superficie. La figura 14 ilustra cómo la iluminación en las superficies depende de la orientación relativa de dichas superficies con respecto a la posición de la fuente de luz.

Junto con la intensidad y la posición, el modelo de la fuente de luz podrá incluir información relativa a su **geometría** (si se trata de un tubo fluorescente, una bombilla o una ventana, por ejemplo); la **distribución de intensidad luminosa** en los distintos puntos de su superficie o volumen; la distribución de la intensidad luminosa que emana la luz (típicamente representada por una red lumínica) y la **distribución espectral** de la luz emitida, que tiene que ver con tipo de luz (la luz del día tiene una distribución espectral diferente a la que tiene una bombilla o a una luz alógena).

III-B. Atenuación

Existen muchos factores que atenúan la intensidad de la luz emitida por una fuente de luz. Un objeto estará más iluminado que otros en función de la distancia a la fuente de luz. Cuanto

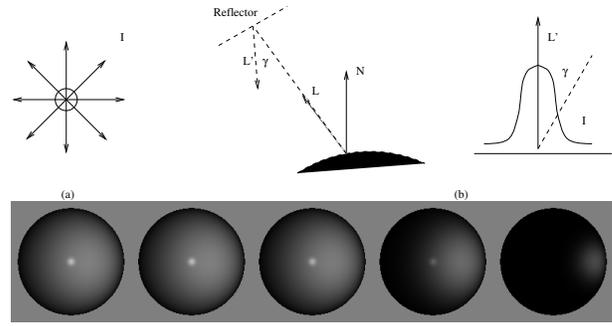


Figura 16: Fuente direccional empleando el modelo de Warn. En la parte superior se esquematiza el modelo de Warn: la fuente de luz puntual mostrada en a) se posiciona sobre el reflector mostrado en b) y el resultado es que la concentración de luz es mayor en las direcciones perpendiculares a la superficie reflectora. En la parte de abajo, Comparación de variaciones del factor p , que toma los valores 201, 208, 264, 712, 4296 de izquierda derecha; en la fila apuntamos al centro (fila superior), a la derecha (fila central) y arriba (fila inferior). Constante $k_a = 0$, $I_p = 1$, $k_d = 0,4$, $k_s = ,4$, $n = 100$

más lejos, menos iluminación. Esto no es considerado por el modelo básico expuesto anteriormente. Para incluirlo, se introduce lo que se denomina factor de atenuación de la fuente de luz f_{att} . De este modo:

$$I = f_{att} I_p k_d (\vec{N} \cdot \vec{L}). \quad (2)$$

El factor de atenuación f_{att} se incluye para hacer que la intensidad I sea función de la distancia entre el punto iluminado y la fuente de luz. Como la energía luminosa decae con el cuadrado de la distancia podemos hacer $f_{att} = \frac{1}{d_L^2}$. La fórmula modela bien el hecho de que a mayor distancia el valor de f_{att} se hace menor y al ser menor, la iluminación final I también será menor en comparación con I_p . Sin embargo, esta fórmula es problemática tanto cuando d_L tiene valores grandes como cuando tiene valores muy pequeños. Se observa empíricamente que cuando la luz está a una distancia considerable, el efecto de la atenuación desaparece, sin embargo en la fórmula a distancias grandes, f_{att} se hace cero suponiendo que la luz se apaga. Además, aplicando la fórmula, si acercamos la luz a los objetos, el factor tiende a infinito lo cual es ridículo también. Para solventar este problema se propone la fórmula:

$$f_{att} = \min \left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1 \right). \quad (3)$$

Con esta nueva fórmula, primero evitamos que el factor de atenuación sea un valor excesivamente grande porque el máximo es 1, que se produce cuando los objetos están muy cerca. Además, para distancias grandes, tenemos un valor umbral que determina que la luz no se apaga.

III-C. Direccionalidad de la fuente de luz

Para conseguir simular que una fuente de luz apunta en determinada dirección sin utilizar una red fotométrica se puede emplear, por un lado el modelo de Warn que permite

concentrar la iluminación en torno a la dirección, y por otro la acotación del volumen de actuación.

En el **modelo de Warn** la fuente de luz se considera como un punto sobre una superficie especular reflectante (figura 16). El resultado es que esta superficie se ilumina concentrando la energía luminosa en la dirección \bar{L}' normal a la superficie especular. La intensidad de la luz en un punto cuya dirección con respecto a la fuente de luz viene determinado por \bar{L} será:

$$I_L = I_{L'} \cos^p \gamma = I_{L'} (-\bar{L} \cdot \bar{L}')^p, \quad (4)$$

El valor del exponente p sirve para simular fuentes de luz direccionales de manera que a mayor valor de p más se concentra la luz alrededor de \bar{L}' .

Además de emplear este modelo, podemos restringir los efectos de la fuente de luz a un volumen determinado de la escena. Para ello describimos dos técnicas básicas:

- El primero de los efectos se denomina de **solapa** o de *puerta de granero*. Sirve para confinar la iluminación a una parte de la escena que ocupa un volumen prismático o cilíndrico. Sólo se evalúa la iluminación en un punto de la escena si dicho punto está dentro del volumen (figura 8 (a)).
- Otro efecto es el denominado **de cono**. Se define un cono con vértice en la fuente de luz y generatriz δ con respecto a \bar{L}' . Sólo se evalúan los efectos de la fuente de luz cuando $\gamma < \delta$ (figura 8 (b)).

III-D. Medidas de intensidad

Hasta ahora hemos utilizado un valor I para referirnos a la intensidad lumínica de una fuente de luz. En muchos modelos de iluminación I es número real normalizado (entre 0 y 100 o entre 0 y 1) que indica si la iluminación es mayor (valores próximos a 100) o menor (valores próximos a 0). Se trata de una aproximación de gran utilidad pero poco realista. Los modelos fotorrealistas emplean unidades físicas mucho más precisas.

La luz es un fenómeno físico de naturaleza electromagnética. La **radiometría** es la ciencia que se ocupa del estudio de la medida de la radiación electromagnética. La **radiometría** es la ciencia que se ocupa del estudio de la medida de la radiación electromagnética. Se considera que la luz tiene naturaleza corpuscular, de manera que se emplea el **fotón** como partícula elemental responsable de las manifestaciones cuánticas del fenómeno electromagnético. La **fotometría**, a diferencia de la radiometría, se encarga de la luz y del impacto en la visión humana.

Con respecto a las unidades que expresan el valor de la intensidad (en el sistema internacional), está el **lumen** que sirve para medir el flujo luminoso o potencia emitida por la fuente de luz que es capaz de percibir el sistema de visión humana. La **candela** (símbolo cd) mide la intensidad luminosa en una dirección dada. El **lux** (símbolo lx) es la iluminancia o nivel de iluminación. Equivale a un lumen/m². Mide la iluminancia que causa una luz en una superficie a una cierta distancia y en la dirección que apunta a la fuente.



Figura 17: Ejemplo del efecto de la temperatura del color de las fuentes (imagen de dominio público vía Wikipedia).



Figura 18: Efecto de sobre-exposición debida al uso de múltiples fuentes de luz en la misma escena.

III-E. Temperatura y color de la fuente de luz

Hasta ahora estamos considerando que la luz es siempre blanca. Una luz blanca tiene componentes de igual valor para las diferentes frecuencias del espectro visible. Ni la luz natural ni la luz artificial suelen ser blancas. La luz que consideramos blanca tiene un matiz de color que llamamos temperatura de color. La temperatura del color de una fuente de luz es la temperatura de un cuerpo radiante negro ideal que irradia luz con un matiz (hue) comparable al matiz de la fuente de luz.

Tiene que ver con el espectro de la luz visible. La luz será fría con temperaturas altas y el aspecto es azulado (sol, monitor...). La luz será cálida con temperaturas bajas y el aspecto es anaranjado: velas, bombillas... .

En luz artificial, podemos controlar el color de la fuente de luz colocando filtros. Para simular este efecto, en informática gráfica se emplea el modelo RGB que descompone la luz en tres componentes fundamentales, una para el rojo R, otra para el verde G y otra para el azul B. Cuando empleamos una fuente de luz con color, en lugar de emplear un único valor I de intensidad, empleamos un valor para cada una de las componentes I_R, I_G, I_B .

Figura 19: Ejemplos de objetos iluminados con la técnica de iluminación de estudio.



Figura 20: Ejemplo de iluminación con técnicas de claroscuro (imágenes cortesía de D. Martínez y E. Rodríguez).

III-F. Suma de efectos y saturación

Si disponemos de múltiples fuentes de luz los efectos de cada fuente de luz sobre los puntos de la escena se suman:

$$I = \sum_{1 \leq i \leq m} f_{att_i} I_{p_i} [k_d(\bar{N} \cdot \bar{L}_i)] \quad (5)$$

Pero puede ocurrir que la suma de todas las componentes haga que el valor de la intensidad supere el máximo. Para evitar este inconveniente, podemos limitar el valor de I a un máximo, o podemos localizar el valor máximo de I en toda la escena y después normalizar dividiendo cada valor de intensidad de la escena por dicho máximo. Para no perder información de color, es conveniente emplear el modelo como el HSV que separan el color de la intensidad, de manera que podemos respetamos el matiz (H) y la saturación (S) y modificar el valor (V).

IV. ILUMINACIÓN DE ESCENAS DE INTERIOR

IV-A. Iluminación de estudio fotográfico

La iluminación de estudio es ampliamente utilizada en publicidad. Los objetos aparecen iluminados en toda su extensión con una luz blanda. Una única sombra blanda aparece al pie del objeto. En ocasiones esa sombra blanda se sustituye o se complementa con un reflejo.

Las claves para conseguir este tipo de iluminación son:

- La luz principal se coloca en posición cenital 12V ligeramente hacia 1V. Es una luz blanda, por tanto se consigue con una luz de cierto volumen.
- Las luces secundarias vienen del contexto, que es un espacio cerrado con paredes y suelo de color blanco. Los laterales y el techo deben ser de blanco mate.
- El suelo puede ser de blanco satinado para aportar el reflejo del objeto.

La figura 19 muestra ejemplos de uso de este tipo de iluminación.

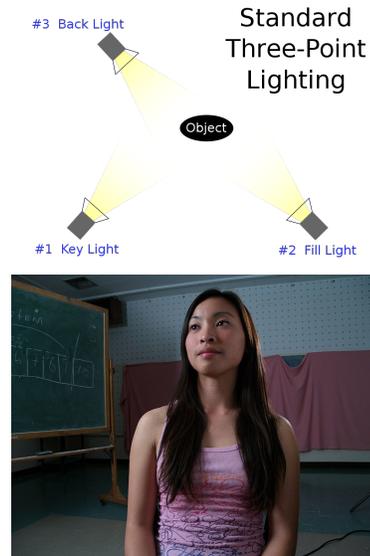


Figura 21: Posición de la cámara en iluminación de tres puntos y resultado (imágenes de dominio público y propiedad de J. Jeffrey CC BY vía Wikipedia).

IV-B. Claroscuro

El claroscuro es una técnica empleada para realzar los contornos de los objetos iluminados. Suelen obtenerse resultados efectistas.

- La luz principal se coloca en posición lateral 9H o 3H. Esta luz hará que las partes iluminadas directamente proyecten sombras sobre el resto del objeto.
- Una luz secundaria se proyecta contra una pantalla que se ubica detrás del objeto. Esta luz no ilumina directamente al objeto. Esta luz crea una falsa sombra que permite ver los relieves del objeto en la parte no iluminada.
- Se pueden incluir luces blandas y de menor intensidad si se considera imprescindible iluminar alguna zona de la superficie del producto que haya quedado oculta.

IV-C. Contraluz

El contraluz realza el contorno de los objetos y se puede utilizar cuando se quiere poner el acento la forma del objeto y no en los detalles de las superficies.

Para conseguir se deben utilizar los siguientes parámetros:

- Luz principal colocada en la parte posterior del objeto 12H 9V. Esta luz conviene que tenga un área lo suficientemente grande o una cobertura suficiente como para iluminar al objeto en toda su extensión.
- Si se quiere ver algo de la superficie del producto es imprescindible añadir luces secundarias y de relleno como en la iluminación de tres puntos. A diferencia de la iluminación de tres puntos, la intensidad de estas luces ahora será menor.

La figura 5 muestra ejemplos de uso de este tipo de iluminación.



Figura 22: Esquema de iluminación con técnicas de ventana y ejemplos. A la izquierda un ejemplo con ventana norte y a la derecha con ventana sur (imágenes cortesía de D. Martínez Pérez).

IV-D. Iluminación de tres puntos

La iluminación a tres puntos es un modo estándar de posicionar las fuentes de luz muy empleada en el cine de hace algunas décadas. Consta de los siguientes componentes:

1. Luz principal 4-5H o 7-8H 1-2V. Establece la dirección de la luz. Crea las sombras principales. Revela formas y texturas. Determina ampliamente la exposición.
2. Luz de relleno: de tipo luz difusa. Elimina sombras. Reduce el contraste tonal.
3. Contraluz 11H-1H 10v-11V. Contorno de luz que marca la silueta

Este tipo de iluminación no es ninguna norma. De hecho está en desuso.

IV-E. Iluminación de ventana

La iluminación de ventana es una luz muy interesante porque la mayoría de obras de arte la han utilizado. Muchas fotografías de productos en su entorno natural es la iluminación que utilizan.

Para conseguirla debemos seguir los siguientes pasos:

1. La luz principal es la de la propia ventana. Distinguimos la iluminación de ventana norte y la iluminación de ventana sur. En la iluminación de ventana norte, la ventana es una fuente de luz pero los rayos de sol no entran directamente por la ventana. En la ventana sur, hay haces de luz solar entrando por la ventana. En ambos casos, la ventana actúa como una fuente de luz de área rectangular que ocupa todo el espacio de la ventana. En la iluminación de ventana sur, una fuente adicional simula la luz del sol y tendrá un intensidad muy superior a la de la propia ventana.
2. Es muy importante en este tipo de render que el objeto se sitúe dentro de un espacio cerrado que simule una habitación cerrada. Es importante porque la luz que proviene de la ventana sufrirá diversos rebotes al incidir con las paredes, techo, suelo y otros objetos y llenará de color los distintos puntos del objeto a iluminar.
3. Conviene que el objeto esté cerca de la ventana, en una posición lateral por ejemplo 4H 2V.

Cuando sólo hay una ventana los contrastes son mayores que cuando hay varias. En la iluminación de ventana sur se



Figura 23: Variedad de iluminación exterior en función de la hora del día (imágenes propiedad de Autodesk 3DS).

puede interponer un objeto tipo reja o vegetación entre la fuente de iluminación que simula el sol y la ventana para proyectar sombras al interior de la habitación que añadan contexto a la escena final. La figura 22 muestra ejemplos de ambos tipos de iluminación de ventana.

IV-F. Iluminación con lámparas y luminarias

La iluminación empleando luminarias o lámparas es un escenario difícil de describir debido a la gran variedad de opciones que se pueden presentar. Se pueden dar múltiples opciones que tienen que ver tanto con la posición de las fuentes de luz como con el color.

Conviene evitar en lo posible este tipo de renders debido a la dificultad y a la existencia de alternativas mucho más efectivas. Cuando sea necesario, es importante utilizar más de un punto de luz y aprovechar múltiples rebotes en el contexto, por ejemplo, acercando las luminarias a las paredes.

V. ILUMINACIÓN DE EXTERIORES

La iluminación exterior depende fuertemente de la hora del día y de la estación del año. Además, el contexto también es determinante; por ejemplo, una imagen en exterior en la costa mediterránea tiene unas condiciones de iluminación diferentes a una imagen en la costa cantábrica aunque la estación del año y la hora coincidan.

La atmósfera de la tierra dispersa las longitudes de onda más cortas de luz del sol causando el efecto del azul del cielo y el enrojecimiento de la luz. Cuando más aire, más dispersión. En el atardecer y amanecer hay más dispersión y con ella más enrojecimiento de la luz. Esta es la razón por la que las condiciones de iluminación varían tanto entre las diferentes horas del día y estaciones del año.

Las nubes son translúcidas, lo que significa que ejercen el efecto de dispersión en la luz. Al dispersar la luz del sol, convierten una fuente de luz dura en una fuente de luz blanda. El color también se ve afectado por esta dispersión cambiando en función de la nube es más o menos densa.

V-A. Componentes

La iluminación de exteriores se apoya en dos fuentes de luz especiales: la luz del sol y la luz del cielo. Una tercera componente es la del contexto, ya que la luz del sol y del cielo rebota en los objetos participando también de la iluminación global. A continuación presentamos cada una de estas componentes:

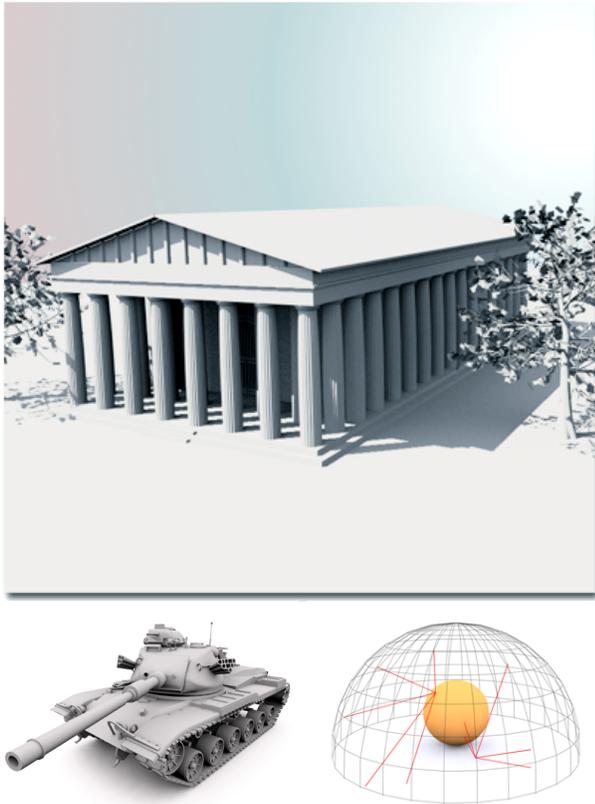


Figura 24: Recursos y empleados para simular los efectos de iluminación exterior. Arriba se muestra el efecto de la luz del sol y el cielo. Abajo se muestra el *dome* y el efecto de sombras débiles que provoca (imágenes propiedad de Autodesk 3DS).

V-A1. *Luz del sol*: La luz del sol (*sunlight*) ha de satisfacer las siguientes restricciones:

- Es una luz direccional. Aunque el sol es una fuente de luz radial, por estar a gran distancia, el efecto es el de una luz direccional.
- No hay atenuación con la distancia.
- El ángulo depende de la hora de día.
- El color es el amarillo salvo al amanecer y atardecer que es anaranjado o rojo. Cuidado porque una cosa es el color de la fuente de la luz y otra el color de la luz emanada. La luz que nos llega del sol es luz blanca.
- Es una sombra dura pero puede tener una penumbra leve. El tamaño de la penumbra depende de la distancia de la sombra al objeto que la produce.

V-A2. *Luz del cielo*: La luz del cielo (*skylight*) atenúa la sombra del sol. Aporta color a la escena, generalmente azul ligero. Se utiliza un *dome* o cúpula sobre la que se coloca una textura. Los rayos de luz provienen en este caso desde todos los puntos de la cúpula. Aporta sombras muy suaves en torno a los objetos.

V-B. *Importancia de la hora del día*

V-B1. *Mediodía*: Cuando el sol está en el cenit está en una de las peores posiciones para colocar una fuente de luz

porque las sombras se proyectan sobre los objetos fotografiado. La luz es además la más intensa del día blanqueando colores y estropeando contrastes. La luz cenital es muy poco reveladora de las formas, sobre todo en la parte superior de los objetos.

V-B2. *Media mañana y atardecer temprano*: En el atardecer y amanecer, la luz del sol se hace menos intensa, lo que favorece que el azul del cielo se vea en las zonas sombreadas. Se consideran las mejores horas para tomar imágenes. Como una hora antes o después de la puesta y salida del sol se conoce como "golden hours" por la calidad de las fotografías que pueden obtenerse. La saturación de los colores es muy intensa.

V-B3. *Amanecer y atardecer*: Cuando el sol está ocultándose o saliendo la luz se vuelve anaranjada a la vez que se debilita. La luz del cielo cobra protagonismo a la vez que la luz del sol pierde fuerza. Las sombras son muy largas y las texturas muy aparentes. Si hay nubes, estas toman un tono anaranjado muy colorido, y este tono puede colorear las sombras llevándolos a tonalidades moradas o rosas.

V-B4. *Anochecer*: El momento justo cuando el sol se ha ocultado hace protagonista al cielo de la iluminación con sombras muy cortas y luz muy tenue. La marca rosa que se pone en el espacio que deja el sol llama *alpenglow*. Esta marca proyecta luz sobre los objetos tiñiéndolos de rosa.

V-B5. *En sombra abierta*: En sombra abierta, el sol no incide directamente. Si estos espacios están iluminados es por la dispersión de la luz del cielo por un lado y por los múltiples rebotes del contexto por el otro. Es una luz difusa que provoca sombras blandas.

V-B6. *Escenas nocturnas*: El *sunlight* se puede emplear para la luna. El color ahora es amarillo o anaranjado. El *skylight* es un azul poco intenso. Aunque no haya sol, el cielo en sí genera iluminación que viene de la luz del sol dispersa o de la luna. La luz del cielo por esto será siempre más luminoso que la tierra cuando miramos al horizonte.

En escenas nocturnas, es conveniente acompañar con luces y destellos para incrementar el contraste. Una ventana encendida al fondo de la escena o los reflejos sobre la calle mojada ayudan a dar sensación de volumen en las escenas nocturnas.

MÁS INFORMACIÓN

Los libros básicos de referencia sobre informática gráfica, no suelen incluir un capítulo específico sobre modelado y representación de fuentes de iluminación. Por ejemplo, en [1] no aparece un capítulo reservado a este asunto, sólo el capítulo 14 trata algunos aspectos mezclándolo con otros temas de rendering e iluminación de superficies; En [2] estamos en la misma situación donde sólo el capítulo 16 señala algunos aspectos combinándolos con otros relativos a rendering.

Hearn [3] se dedica el capítulo 26 a presentar cuestiones relativas a la física de la luz.

Donde puede encontrarse más información sobre cuestiones de iluminación es en otra serie de libros más cercanos al uso de programas informáticos por parte de creativos. [4] dedica el capítulo 1 a distinguir los tipos de fuente de luz y la parte III al uso de las fuentes en contextos de iluminación de escenas. [5] dedica los cinco primeros capítulos a desarrollar los temas tratados en el presente artículo.

Es muy recomendable la lectura del libro completo [6] en cuestiones relativas a la iluminación de escenas. Con un tono muy divulgativo profundiza en asuntos relativos a iluminación que deben tenerse en cuenta para avanzar en este dominio con cierta solidez. Se recomienda acceder a manuales y blogs de fotografía para entender la importancia de manejar correctamente las fuentes de iluminación para conseguir buenos resultados.

Otros manuales más técnicos son interesantes para apartados específicos como el referido a los diagramas goniométricos. Son manuales de iluminación relacionados directamente con el ámbito industrial como [7], [8].

EJERCICIOS Y CUESTIONES

1. ¿Qué se entiende por calidad de una fuente de luz?
2. Si decimos que una fuente de luz está colocada en la posición 6V 8H, ¿Puede hacer un esquema para ubicarla con precisión?
3. Explique para qué sirve el modelo de Warn. No es necesario desplegar la formulación. Sólo indique para qué sirve.
4. Explique el modelo de iluminación de Phong. Describa cada uno de los parámetros. (Sólo el modelo de iluminación de Phong).
5. Detalle el modelo de Warn de fuente de luz.
6. En iluminación e informática gráfica, ¿qué es un *dome* y para qué se utiliza?
7. ¿Qué es el factor de atenuación y cómo se modela?
8. ¿Qué es la temperatura de un color?
9. ¿Qué es un diagrama goniométrico y para qué se utiliza?
10. Enumere tres tipos de fuentes de iluminación en función de la dirección de los rayos de luz que emanan de ella.
11. Indique en qué se diferencia una luz blanda de una luz dura. Discuta el uso de estos tipos de luz en un render de ventana norte.

REFERENCIAS

- [1] D. Hearn and M. P. Baker, *Computer Graphics, C Version*, segunda ed. Prentice Hall, 1997.
- [2] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [3] D. Hearn, M. P. Baker *et al.*, *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.
- [4] V. Wislar, *Illuminated pixels: the why, what, and how of digital lighting*. Cengage Learning, 2013.
- [5] J. Birn and J. Bim, "Digital lighting and rendering new riders," 2006.
- [6] R. Yot, *Light for visual artists: understanding & using light in art & design*. Laurence King Publishing Limited, 2011.
- [7] F. M. Sánchez, *Manual práctico de iluminación*. Antonio Madrid Vicente, 2005.
- [8] G. García-Gaite, "Iluminación y seguridad laboral," *Madrid: Fundación Mapfre*, 2003.

Generación de sombras

ÍNDICE

I.	Fundamentos	2
I-A.	Tipos de sombras	2
I-B.	Usos	2
II.	Características de las sombras	3
III.	Algoritmos para generar sombras	4
III-A.	Depth map	4
III-B.	Raytraced	4
III-C.	Ventajas e inconvenientes	5
	Referencias	6

Las sombras son un integrante esencial de las escenas fotorealistas. Parece difícil imaginarse una escena en la que haya fuentes de iluminación que proyecten luz sobre los objetos sin que éstos no arrojen ninguna sombra. Sin embargo, hasta hace unas pocas décadas las sombras se consideraban una sofisticación inalcanzable para muchas aplicaciones de informática gráfica como por ejemplo, los videojuegos.

Integrar sombras en una escena sintética supone multiplicar considerablemente el número de cálculos a realizar. No es un problema incluir sombras en aplicaciones en las que las imágenes se generan asumiendo que puede pasar un tiempo largo desde se comienza el proceso de renderizado hasta que termina como por ejemplo en generación de efectos especiales para cine. Sin embargo, cuando es necesario generar imágenes en tiempo real, es posible que, aun utilizando hardware específico como el que tienen las consolas de videojuegos, haya que renunciar a la calidad de las sombras para poder emplearlas.

En este documento, primero mostramos los tipos de sombra y sus usos y características para después, describir los métodos que se emplean en informática gráfica para la generación de sombras sintéticas. La presentación de estos métodos de generación de sombras nos ayudará a entender por qué la generación de sombras sintéticas es un proceso tan costoso computacionalmente hablando.

Figura 1: Ejemplos de tipos de sombra. Arriba de izquierda a derecha sombra primaria, secundaria y terciaria. En la segunda fila sombra falsa y de contacto. La sombra primaria es la que la esfera arroja sobre sí misma. La sombra secundaria es la que la esfera arroja sobre el suelo. Las banderas y personas de la tercera escena arrojan sombras terciarias. El perfil de la modelo se puede ver porque hay una falsa sombra provocada por un foco que se proyecta contra la pantalla posterior. Las botas generan una sombra de contacto en el suelo.

I. FUNDAMENTOS

I-A. Tipos de sombras

Distinguimos los siguientes tipos de sombras: primarias, secundarias, terciarias, sombras falsas y de contacto. A continuación, definimos cada una de ellas y mencionamos sus usos.

Las **sombras primarias** o sombras de forma (en inglés *form shadow*) son las que caen sobre el objeto como consecuencia de su propia forma. Las sombras primarias son un efecto natural de un objeto iluminado y ayudan a revelar su forma. Cuando este tipo de sombras son demasiado prominentes pueden distraer; por ejemplo, cuando la sombra de la nariz oscurece el rostro.

Las **sombras secundarias** o sombras proyectadas (en inglés *cast shadow*) son aquellas producidas por el objeto en los objetos circundantes. Estas sombras son muy importantes porque tienden a integrar el objeto con el resto de los objetos de la escena en la que se ubica. A partir de las dimensiones y posición de la sombra podemos determinar la proximidad o lejanía con respecto a otros objetos circundantes.

Las **sombras terciarias** son aquellas proyectadas sobre un objeto y producidas por otros objetos cercanos; por ejemplo, la sombra de las hojas de un árbol sobre una persona. Las distinguimos de las sombras secundarias porque el objeto que proyecta la sombra no aparece en la escena. Estas sombras ayudan a conectar el objeto con el entorno que lo rodea.

La **sombra de contacto** es aquella que se crea en las zonas en las que hay contacto entre objetos. Este tipo de sombra está relacionada con la iluminación ambiental global de la escena. En este sentido, mientras que las sombras primarias, secundarias y terciarias dependen fuertemente de la presencia de fuente de luz, en el caso de las sombras de contacto no es necesario. Contribuyen a evitar que los objetos den la sensación de flotar sobre los planos en los que se apoyan.

La **sombra falsa** es aquella que aparece porque una zona concreta de la escena que no está iluminada. Se ilumina sólo parte de un escenario de manera que las zonas no iluminadas están en sombra. En realidad, no es una sombra sino una zona no afectada por la fuente de luz.

En la figura 1 se muestran ejemplos de estos tipos de luz.

I-B. Usos

Las sombras son un componente que aporta información fundamental para entender la escena. Entre las funciones más importantes de las sombras están la definición de las relaciones

Figura 2: Uso de las sombras para resolver relaciones espaciales.

Figura 3: Uso de las sombras para revelar detalles de los objetos que no son visibles por la cámara.

espaciales, la revelación de ángulos alternativos, la mejora de la composición y la visualización del contexto.

Las sombras permiten **resolver las relaciones espaciales** porque sin ellas, se pierde la referencia de proximidad y lejanía cuando no tenemos claro el tamaño de los objetos representados. En la figura 2 vemos como sin sombras es imposible saber qué objetos están más cerca y cuales están más lejos, cuales más arriba y cuales más abajo. La sombra sobre el suelo y sobre la pared resuelve ambigüedades que se pueden dar, en este caso, cuando los objetos tienen la misma forma, pero distinto tamaño. En la misma figura vemos como una escena inverosímil (un animal en un vagón de metro) se vuelve más creíble cuando las sombras se proyectan en la escena.

Las sombras permiten **revelar ángulos alternativos** que quedan ocultos debido a la posición de la cámara. En la figura 3 vemos cómo una misma figura como es el rectángulo cobra sentido cuando vemos lo que se esconde detrás de un objeto gracias a las sombras. En la imagen del busto, las sombras nos permiten ver partes de la cabeza que no se ven de forma directa por quedar en la parte de atrás. Por ejemplo, el detalle del corte de pelo de que se muestra en la figura 3 no podría verse si no fuera por la sombra proyectada. En la misma figura, vemos que la sombra proyectada sobre una superficie permite distinguir entre una superficie con surcos o una superficie con rayas.

Las sombras, principalmente las terciarias, aunque no sólo, permiten **visualizar el contexto**. Las sombras proyectadas por elementos que no se ven en la escena de forma directa permiten enriquecer la composición a la vez que aportan una información que puede ser importante para entender la escena (ver figura 3).

Una sombra tiene por sí, un **valor comunicativo** que puede ayudar a mejorar la composición, equilibrando la imagen final y añadiendo contraste. En la figura 4, vemos que la sombra superior permite equilibrar la composición aportando una masa en la parte superior de la escena, mejorando también el contraste en la zona de la embocadura de la jarra.

En el mundo real todas las luces que no son blandas proyectan sombras. En los escenarios 3D, generalmente podemos elegir si las luces que ponemos en la escena proyectan o no proyectan sombra. También puedes elegir qué objetos proyectan sombras. Cuando las sombras que proyecta una determinada luz no sirven para mejorar la composición conviene quitarlas. En los estudios de fotografía, televisión o cine esto se hace incluyendo luces de relleno cuya misión

Figura 4: Uso de las sombras para mejorar la composición de la escena. En la parte izquierda la sombra añade contraste y equilibrio. En la parte central y derecha se eliminan sombras para limpiar la escena.

Figura 5: Factores que afectan en el tamaño de la sombra y en su longitud.

es eliminar sombras innecesarias (luces sucias). En el mundo de la informática gráfica hacer esto puede ser muy sencillo porque el hecho de que una luz proyecte o no sombras es una opción de configuración de la fuente de luz. Hay que tener en cuenta que cuando se usa una luz que no proyecta sombra, hay que intentar que no se note su presencia (por ejemplo evitando que aparezcan reflejos o brillos reveladores), porque de lo contrario, la sensación de realismo caerá.

II. CARACTERÍSTICAS DE LAS SOMBRAS

Las características fundamentales de las sombras son tu tamaño, longitud, nitidez, color y densidad. A continuación describimos cada uno de estos términos detallando sus dependencias.

El **tamaño de la sombra** siempre es mayor que el tamaño del objeto. El tamaño aumenta a medida que se reduce la distancia lámpara/objeto. Además, el tamaño de la sombra aumenta con la distancia objeto/fondo. Se cumple la relación:

$$\frac{\text{Tamaño de la sombra}}{\text{Tamaño del objeto}} = \frac{\text{Distancia fuente/sombra}}{\text{Distancia fuente/objeto}} \quad (1)$$

En la figura 5 (parte superior) vemos, en la parte 1, que al acercar la fuente de iluminación desde la posición A hasta la posición C, el tamaño de la sombra aumenta. En la parte 2, vemos que al alejar el objeto de la fuente de iluminación desde la posición A a la posición C, el tamaño de la sombra disminuye.

La **longitud de la sombra** disminuye con la elevación vertical de la fuente de luz. A mayor elevación de la fuente de luz, menor sombra. Por otro lado, la inclinación de la superficie en la que se proyecta la sombra también hace que la sombra pueda tener una longitud diferente como puede verse en la figura 5 (parte inferior).

La **nitidez de la sombra** nos permite distinguir una zona de umbra y otra de penumbra. En la zona de umbra la sombra es más densa que en la de penumbra. Cuanto menor sea la penumbra, mayor nitidez de la sombra. Una sombra es dura cuando no tiene penumbra. Una sombra es suave cuando apenas tiene umbra. La figura 6 muestra los factores que influyen en la nitidez de las sombras. La nitidez disminuye:

1. Con el área de la fuente de luz.

Figura 6: Factores que afectan en la formación de umbra y penumbra en las sombras de los objetos iluminados.

Figura 7: Cálculo de la proyección de las sombras.

2. Al separar el objeto del fondo sobre el que se proyecta la luz.
3. Al disminuir la distancia entre objeto a iluminar y la fuente de iluminación.
4. Cuando el objeto es pequeño con respecto a la fuente de luz.

El tamaño de la fuente de luz afecta pues al tamaño y a la nitidez de la sombra. La figura 6 ilustra la razón por la cual cuanto mayor es el área de la fuente de luz mayor es la penumbra. Ningún punto del área de la fuente de luz es visible desde las zonas de umbra. En cambio, en las zonas de penumbra sí que son visibles una parte de los puntos de la superficie de la fuente de iluminación. Cuantos más puntos de la fuente de luz son visibles, menos densa será la sombra en la zona concreta de penumbra. El mismo esquema explica por qué hay más penumbra cuando la distancia entre pantalla y objeto o distancia entre fuente y objeto disminuyen.

La última característica de las sombras que vamos a mencionar es el **color**. Aunque percibamos las sombras en blanco y negro, con tonalidad de gris, éstas pueden y suelen tener color. El color de una sombra viene determinado por el contexto. A cielo abierto, por ejemplo, las sombras tienen un tono azul. En el interior de un bosque, las sombras tienen un tono verde debido al color reflejado por la masa vegetal.

Para **calcular la proyección de la sombra** en la escena, hay que tener en cuenta la posición de fuente de luz, incluyendo su altura, y la posición de los planos de proyección. La figura 7 ilustra cómo hacerlo cuando la fuente de luz no es visible en la escena, cuando está detrás del objeto y cuando está delante. Cuando está detrás, se lanzan proyectores desde un punto imaginario donde se supone que está la fuente de luz que pasen por los puntos frontera del objeto (1). Luego se trazan la vertical que, pasando por la fuente de luz intersecta en el horizonte (2). Desde el punto de intersección con el horizonte se vuelven a lanzar proyectores a los puntos de la base del objeto para calcular la longitud de la misma (3). En caso de que la sombra esté delante del objeto, primero se establece el punto de desvanecimiento de la sombra en el horizonte (1) y se traza una vertical (2). Este punto determina la posición (excepto la altura) de la fuente de luz. En esa vertical se establece un punto que está en relación con la altura de la fuente de luz. Cuando más lejos de la horizontal esté este punto, más alta está la fuente de luz. Desde este punto, se lanzan proyectores que pasan por los puntos frontera del objeto y que determinan la longitud de la sombra.

Figura 8: Esquema de la operativa de los métodos de generación de sombras de mapa de profundidad (a la izquierda) y de trazado de rayos (a la derecha).

III. ALGORITMOS PARA GENERAR SOMBRAS

En informática gráfica se recurre principalmente a uno de estos dos algoritmos a la hora de generar sombras: mapa de profundidad (*depth map shadows* o *shadow map* en inglés) por lado y trazado de rayos (*raytraced shadows* en inglés) por otro. En el método de *shadow map* se precalcula dónde van a caer las sombras de cada fuente de luz y se hace un mapa de distancias para cada una de ellas. En el método de trazado de rayos, se proyectan haces de luz desde la cámara y se comprueba si en sucesivos rebotes inciden o no sobre una fuente de luz; si no lo hacen el punto está en sombra. La figura 8 esquematiza ambos procesos.

Si el método del mapa de profundidad es rápido y eficiente, el método de trazado de rayos necesita gran cantidad de tiempo de cálculo; si el primero ofrece calidad baja y se suele retocar para evitar artefactos, el segundo es mucho más preciso a cualquier resolución. A continuación detallamos cada uno de los métodos y desglosamos las ventajas e inconvenientes de cada uno de ellos.

III-A. *Depth map*

El *depth map* es una matriz de números que almacena distancias. Se hace un *depth map* para cada fuente de luz que proyecta sombras. Para cada dirección en la que hay un rayo de luz, el mapa guarda la distancia al objeto más cercano que proyecta sombra. La figura 9 (parte superior) muestra cómo para una fuente de luz dada, se crea un *Depth Map* en el que se almacena la distancia al objeto más cercano. Los puntos a una distancia mayor estarán en sombra.

En este algoritmo, la cámara juega también un papel importante. En la figura 9 (parte inferior) vemos dos escenarios para destacar por qué algunas zonas de la escena aparecen en sombra y otras no. Dado un punto P de la escena que sea visible desde la cámara, le corresponde un pixel único en el *shadow map*. Se comprueba el valor de Z de dicho punto desde la fuente de luz (en la figura es Z_B). Si el valor almacenado en el *shadow map* para el punto P tiene un valor Z menor que Z_B , entonces P está en sombra porque eso significa que hay un objeto interpuesto entre el punto B y la fuente de luz. En figura 9, en la parte izquierda, el punto P está en sombra porque el valor almacenado en el *shadow map* es menor que el correspondiente del punto P ($Z_B > Z_A$); en la parte derecha, el punto P no está en sombra porque le corresponde un valor de Z en el *shadow map* que es precisamente Z_B por ser el punto más cercano a la fuente de luz en la dirección que une el punto con dicha fuente.

Este método de generación de sombras tiene un problema relacionado con el **tamaño de memoria** necesaria para poder implementarlo. El mapa de profundidad (*Shadow Map* o *Depth*

Figura 9: Funcionamiento del método de generación de sombras de mapa de profundidad.

Figura 10: Efecto del tamaño de la zona iluminada en relación con la resolución del mapa de profundidad. Al aumentar la cobertura de la fuente de luz, si no se aumenta el tamaño del **Shadow Map** la sombra aparece pixelada.

Map) es una matriz cuadrada que almacena las distancias a los puntos más cercanos a la escena desde la fuente de luz. Cada distancia es un número real que suele ocupar 4 bytes. Si el número de fuentes es elevado y la resolución del mapa de profundidad también, las necesidades de memoria crecen y puede ser un problema. Si un mapa de una resolución de 128 pixeles necesita 0.06MB, uno de 4096 pixels necesitará 64MB ($4096 \times 4096 \times 4 / 1024 / 1024 = 64\text{MB}$).

Otro condicionante de esta técnica, es que es importante acotar el volumen de actuación de la fuente de luz para que la resolución del mapa sea efectiva (*depth map framing*). Una fuente de luz que actúe en un volumen de la escena elevado necesitará un mapa de profundidad de mayor resolución para conseguir unas sombras de una calidad comparable a las obtenidas con una fuente de iluminación que actúe sobre un volumen menor (ver imagen 10). Algunos tipos de fuentes de luz, de hecho, no pueden tener un mapa de profundidad porque no acotan el volumen de la escena en el actúan. Tal es el caso de las fuentes de luz puntual. En una fuente de luz puntual, a diferencia de lo que pasa en una fuente de luz tipo *spotlight* no podemos ajustar el mapa a la cobertura de la luz porque una fuente de luz puntual cubre todo el espacio. Cuanto mayor cobertura tenga la luz, mayores necesidades de tamaño para el *Shadow Map*.

III-B. *Raytraced*

El método de trazado de rayos no es sólo un método para generar sombras, sino que es un método de representación de escenas 3D que genera las sombras como consecuencia de la representación de escenas con realismo. La idea básica es lanzar rayos de visión desde la posición de la cámara pasando por cada pixel de la ventana de visión para calcular el color con el que representar estos pixels; el conjunto de colores de los pixels es la imagen resultante. Para cada rayo de visión se calcula el punto de la superficie del primer objeto de la escena con el que se cruza. Si desde ese punto de la escena no hay visión directa de una fuente de luz, entonces ese punto está en sombra.

En realidad, no es tan sencillo porque se rastrean también los reflejos y refracciones de cada rayo de visión de forma recursiva. Puede haber cientos de iteraciones y en cada intersección se calculan los efectos de las distintas fuentes de iluminación, de manera que este método consigue tener en



Figura 11: Generación de sombras de objetos transparentes en el método de mapa de profundidad (izquierda) y en el método de trazado de rayos (derecha) (imágenes propiedad de Autodesk 3DS).

Figura 12: Sombras con zona de penumbra generadas empleando el método del mapa de profundidad (izquierda) y de trazado de rayos (derecha).

cuenta los efectos conjuntos de los distintos elementos que configuran la escena. Al calcularse los efectos de múltiples fuentes de luz y de múltiples rebotes de la luz en otros objetos, las sombras generan zonas de umbra y penumbra de forma natural.

III-C. Ventajas e inconvenientes

A la hora de trabajar con **objetos transparentes** los dos algoritmos ofrecen resultados muy diferentes. En sombras tipo *shadow map* se generan sombras sólidas. El algoritmo nos permite estimar si un punto está en sombra o no, pero nada dice de su grado de transparencia. La transparencia sólo se puede simular poniendo una textura sobre la sombra: una imagen se simula los colores de la transparencia real. En cambio, en sombras de tipo *raytraced shadow* las sombras de las transparencias se generan de forma natural porque para cada rayo se calculan los efectos de los rayos refractados de forma recursiva. La figura 11 ilustra las diferencias.

Con respecto a las **zonas de sombra**, la umbra y penumbra, en sombras tipo *shadow map*, un filtro suaviza la sombra en los contornos. Hay un posproceso que calcula las texturas de las sombras para conseguir representar la penumbra. En las sombras tipo *raytraced*, hay que usar luces tipo área o volumen y distribuir diversos emisores a lo largo del mismo.

Siempre se puede conseguir sombras suaves cuando hay varios emisores de luz. La sombra emitida por cada emisor quedará iluminada parcialmente por la luz de los otros. En las luces tipo volumen o área se distribuyen emisores a lo largo de una forma geométrica y el resultado son penumbras de gran realismo.

Es fácil distinguir si una sombra ha sido generada empleando el método de trazado de rayos o el método del mapa de profundidad. En las sombras de mapa de profundidad, la penumbra tiene el mismo ancho a lo largo de todo el contorno, mientras que en las sombras de tipo trazado de rayos la zona de penumbra crece con la distancia (ver figura 12). En el método de mapa de profundidad la penumbra se genera emborronando



Figura 13: Empleo de desplazamientos artificiales de la sombra (*bias*) para evitar que se vean zonas de penumbra por detrás del objeto (imagen propiedad de Autodesk 3DS)

Figura 14: Aspecto granulado en las sombras generadas empleando trazado de rayos.

el contorno de la sombra combinando el color de la sombra con los colores de los pixels vecinos. Esta técnica puede generar efectos indeseables como que la sombra se expanda por detrás del objeto que la produce. Para evitar estos efectos se permite incluir un *bias* o desplazamiento de la sombra. La figura 13 ilustra este fenómeno.

En las sombras tipo *raytraced* la sombra puede aparecer con un **aspecto granulado**, como con ruido (imagen de la izquierda en figura 14). Es la consecuencia de los múltiples rayos emitidos en el proceso de construcción que tienen un procesamiento independiente. Para mitigar este efecto se puede aumentar el número de rayos pero esto es costoso en tiempo de ejecución. Como alternativa se puede hacer uso de soluciones tipo *anti-aliasing* para corregirlo. En estos casos, básicamente se filtran los puntos de las sombras considerando los puntos de color de las zonas vecinas para recalcular cada punto de las sombras.

MÁS INFORMACIÓN

Los libros básicos de referencia sobre informática gráfica, no suelen incluir un capítulo específico sobre cálculo de sombras. Por ejemplo, en [1] no aparece un capítulo reservado a este asunto, sino que estos temas aparecen integrados dentro de los capítulos de rendering (capítulo 15). En [2] estamos en una situación parecida donde sólo el capítulo 14 dedicado a modelos de renderizado de superficie trata el asunto de las sombras. Quien sí que aborda este asunto con profundidad es [3] que dedica el capítulo 10 a detallar algoritmos de trazado de sombras.

Donde puede encontrarse más información sobre cuestiones de iluminación es en otra serie de libros más cercanos al uso de programas informáticos por parte de creativos. [4] dedica el capítulo dos a hablar de aspectos relativos a las sombras. [5] dedica el capítulo tres a desarrollar los temas tratados en el presente artículo (de hecho, buena parte de los ejemplos que mostramos salen de este texto).

Es muy recomendable la lectura del libro completo [6] en cuestiones relativas a la iluminación de escenas y también a las sombras. Con un tono muy divulgativo profundiza en asuntos relativos a iluminación que deben tenerse en cuenta para avanzar en este dominio con cierta solidez.

EJERCICIOS Y PROBLEMAS

1. Indique la diferencia entre *form shadow* y *cast shadow* y los efectos de cada una de ellas.
2. Nombre cuatro tipos diferentes de sombra y localícelas en la figura (falta la figura).
3. Indique tres funciones de las sombras en las imágenes 3D.
4. Desde el punto de vista de la nitidez de la sombra, ¿cuáles serían las zonas distinguibles?
5. ¿De qué depende la nitidez de una sombra?
6. ¿De qué depende el tamaño de una sombra?
7. ¿Qué relación existe entre el tamaño de una sombra y la distancia entre la lámpara y el objeto?
8. Indique una situación en la que pueda interesar incluir una luz que no proyecte sombra en un render 3D.
9. Normalmente pensamos que las sombras son grises no tienen color o que son grises, sin embargo, hay situaciones en las que las sombras tienen color. ¿En cuáles?
10. ¿Qué relación existe entre el tamaño de la zona de umbra y el tamaño de la fuente de luz?
11. Explique en tres frases, en qué consiste el método *depth map* de generación de sombras.
12. Explique en tres frases, en qué consiste el método *trazado de rayos* de generación de sombras.
13. Especule sobre el método que se ha empleado para generar las sombras de la imagen de la figura (falta la figura).
14. Indique cómo acometen el problema de las transparencias los distintos métodos de generación de sombras que conoce.
15. Indique la estrategia general que se sigue en informática gráfica para tener zonas de umbra y de penumbra.
16. En la técnica de generación de sombras llamada *Depth map*, ¿Por qué es importante acotar la zona de actuación de la fuente de iluminación?
17. Indique las capacidades de los algoritmos de generación de sombras para producir transparencias.
18. En la técnica de generación de sombras llamada *Depth map*, ¿Por qué puede ser importante añadir un BIAS a las sombras generadas?
19. En la figura 19 (izq) se representa un render tipo ventana sur. Haga una discusión sobre el tipo de luces y de sombras que debería utilizar para conseguirlo sin utilizar técnicas de iluminación global.

Figura 15: Imágenes para plantear ejercicios sobre sombras.

20. A la vista de la imagen 19 (der) haga una discusión sobre la relación entre el tamaño de la fuente de luz y la nitidez y composición de la sombra.
21. ¿Qué es un z-buffer y para qué sirve?

REFERENCIAS

- [1] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [2] D. Hearn and M. P. Baker, *Computer Graphics, C Version*, segunda ed. Prentice Hall, 1997.
- [3] A. Watt, *3D Computer Graphics*, tercera ed. Addison-Wesley, 2000.
- [4] V. Wislar, *Illuminated pixels: the why, what, and how of digital lighting*. Cengage Learning, 2013.
- [5] J. Birn and J. Birn, "Digital lighting and rendering new riders," 2006.
- [6] R. Yot, *Light for visual artists: understanding & using light in art & design*. Laurence King Publishing Limited, 2011.

Materiales: sombreado de superficies, reflexión y refracción

ÍNDICE

I. Representación de alambres	2	Los objetos 3D que representamos en informática gráfica internamente son geometrías formadas por mallas de puntos que se visualizan con superficie suavizadas, con colores, con reflejos e incluso transparencias. En este tema veremos alguno de los fundamentos necesarios para poder representar las mallas de poligonales como objetos sólidos de diferentes materiales. Se presentan los métodos de forma incremental, primero presentando la representación de alambre, luego la eliminación de superficies ocultas, continuamos con los modelos de iluminación que asignan color difuso y reflexión a las superficies. Después mostramos los métodos que permiten suavizar las aristas frontera entre las superficies para finalizar con algunas consideraciones sobre reflexión y refracción en los materiales.
II. Supresión de superficies ocultas	2	
III. Iluminación y color de superficies	3	
III-A. Modelo de iluminación ambiental . . .	3	
III-B. Reflexión difusa	4	
III-C. Superficies coloreadas	5	
III-D. Reflexión especular	5	
III-E. Modelos que emplean propiedades físicas	6	
IV. Sombreado de superficies	6	
IV-A. Sombreado de intensidad constante . . .	7	
IV-B. Sombreado de Gouraud	7	Los objetos 3D suelen estar modelados con poligonales que no son más que conjuntos de aristas ordenadas. ¿Cómo es que no vemos dichas aristas cuando renderizamos? ¿Cómo se consigue el aspecto suavizado de las superficies? Además, cuando representamos los objetos, sólo vemos las superficies que están más próximas, y no mostramos las superficies que quedan ocultas. ¿Cómo conseguimos hacer que desaparezcan? En ocasiones vemos que unos objetos reflejan otros y que algunos incluyen transparencias. ¿Cómo es posible cuando todo lo que representamos son poligonales? La figura 1 representa esta evolución que, de forma incremental, aumenta el realismo de la representación.
IV-C. Sombreado de Phong	7	
V. Reflexiones	8	
VI. Transparencias	9	
VII. Más información	11	
Referencias	12	

En este capítulo daremos respuesta a estas preguntas. Primero presentamos el modelo de alambres como representación más sencilla del modelo de alambres. Mostramos a continuación, cómo eliminar superficies ocultas para después presentar los métodos que permiten dar color a cada superficie y suavizas las fronteras entre superficies para eliminar las aristas. Finaliza el tema presentando algunas consideraciones sobre reflexión y refracción en superficies.

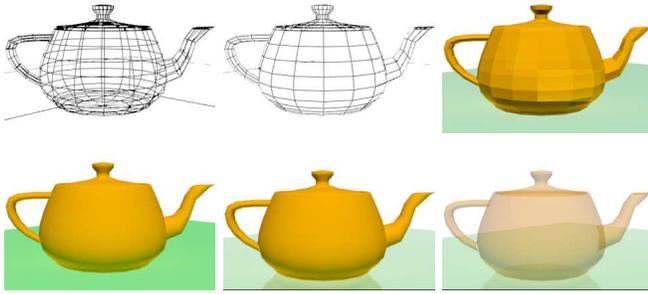


Figura 1: Ejemplos de rendering que emplean diferentes modos de sombreado. De izquierda a derecha y de arriba abajo vemos un modelo de alambres, el mismo modelo de alambres con supresión de superficies ocultas, el modelo con iluminación en las poligonales, el modelo con suavizado de poligonales, reflejos y por último transparencias.

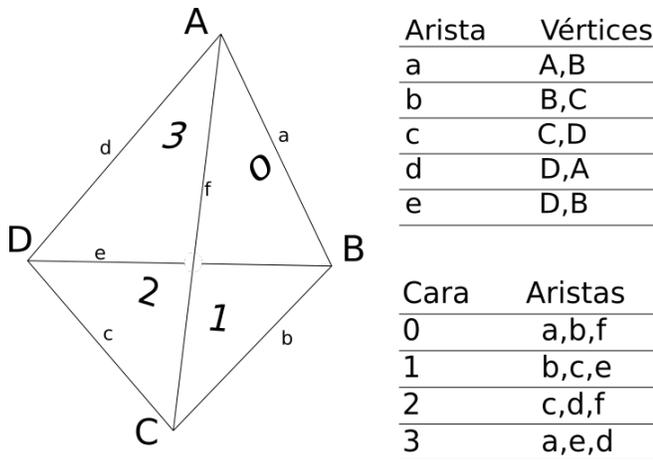


Figura 2: Las tablas de aristas completan las tablas de vértices (donde aparecerían las coordenadas de A, B, C y D) para guardar la información necesaria para representar internamente los objetos 3D. Un recorrido a la tabla de aristas puede servir para hacer la representación de modelo de alambres de los modelos 3D.

I. REPRESENTACIÓN DE ALAMBRES

Una representación con modelo de alambres (*wireframed*) es la representación 3D más sencilla. Consiste en trazar cada una de las aristas del modelo geométrico de los objetos 3D. Para ello, se acude a la lista de aristas de la representación interna del modelo 3D (ver figura 2) y se dibujan una a una las aristas. Para trazar las aristas se dispone de primitivas de dibujo básicas elementales que trazan segmentos (incluso polilíneas o secuencias de segmentos) a partir de los vértices extremos.

La representación de alambres es muy poco realista. Si ni siquiera se eliminan superficies ocultas el resultado final es muy difícil de interpretar. En la figura 3 se muestra un ejemplo en el que se superpone un render realista con un modelo de alambre. Este tipo de imágenes son bastante efectistas y se consiguen superponiendo varias imágenes en un editor gráfico. En la figura se han superpuesto tres imágenes: la imagen

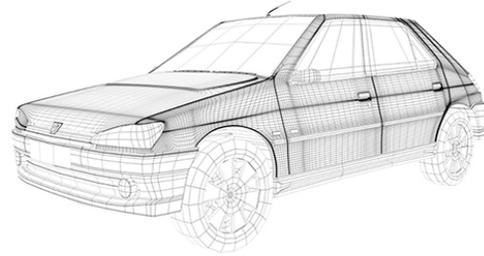


Figura 3: Imagen que explota el modelo de alambres (imagen propiedad de Autodesk Arnold).

realista, una versión con modelo de alambre y una versión en blanco y negro en el fondo. A las distintas capas se han aplicado transparencias.

II. SUPRESIÓN DE SUPERFICIES OCULTAS

Los algoritmos de supresión de superficies ocultas sirven para que en la ventana de visualización no aparezcan los objetos o las caras de los objetos que no serían visibles desde el punto de vista del observador. La figura 4 ilustra esta funcionalidad.

El primer proceso que se aplica para eliminar superficies ocultas en el de **eliminación de caras de atrás** o *back-face culling*. Cada superficie tiene un vector normal asociado \vec{N} (ver figura 5). Desde cada superficie puede trazarse un vector \vec{V} que apunte hacia la cámara. Si el ángulo que forman \vec{V} y \vec{N} es mayor que 90° o menor de -90° , entonces se considera que la superficie no debe representarse porque no es visible desde la cámara. La superficie mira hacia otro lado y no debe representarse. El ángulo que forman dos vectores se calcula con el producto escalar de manera que el $\cos(\alpha) = \frac{\vec{N} \cdot \vec{V}}{|\vec{N}| \cdot |\vec{V}|}$ debe ser ≥ 0 para que la superficie sea considerada visible desde la cámara.

Este método funciona cuando el objeto es sólido y no tiene aperturas. Si tiene aperturas puede ocurrir que las superficies interiores no se visualicen. Esto es así porque los vectores normales a las superficies son únicos para cada superficie, y en los objetos cerrados, estas normales apuntan hacia afuera. El resultado es que las caras interiores de un polígono cerrado no se representan porque el ángulo de sus vectores normales con el ángulo de visión tiene un coseno negativo. Este caso de superficies que desaparecen es más frecuente de lo que nos gustaría y aparece mucho cuando un modelo geométrico creado en un programa de modelado es importado a otro

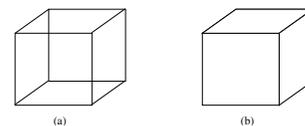


Figura 4: Representación de la figura antes de aplicar el algoritmo de supresión de superficies ocultas (a) y después de aplicar el algoritmo de supresión de superficies ocultas (b).

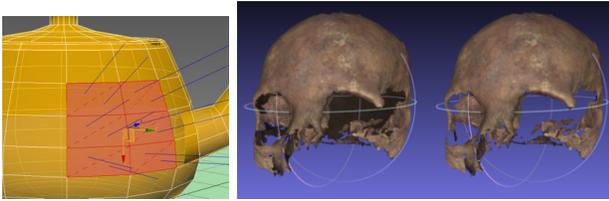


Figura 5: A la izquierda se muestran los vectores normales a las superficies del objeto. A la derecha los efectos de desaparición de caras en el back-face culling en objetos abiertos (la imagen de la derecha es propiedad de C. Rochini CC BY-SA vía Wikipedia).

programa diferente. Las superficies no desaparecen, simplemente es que su vector normal mira hacia el lado inesperado. Para resolver este inconveniente hay dos soluciones: por un lado aplicar un volteo del vector normal (operación *flip*) o indicar al motor de render que visualice ambas caras de cada superficie.

Cuando los objetos se solapan desde el punto de vista del observador, algunas de las superficies pueden quedar ocultas. No es tan sencillo como encontrar las superficies que quedan detrás de otras porque las superficies pueden aparecer particionadas (ver figura 4).

Distinguimos dos tipos básicos de algoritmos:

- De **objeto-espacio** donde se comparan los objetos entre sí para saber qué superficies y líneas no son visibles y
- De **imagen-espacio** donde la visibilidad se decide punto a punto en cada pixel del plano de visualización.

La mayoría de estos algoritmos incluyen técnicas de ordenación de superficies y hacen uso de las propiedades de coherencia de imagen, aprovechando la línea de rastreo para que la supresión de líneas ocultas se realice lo más rápido posible.

Algunos métodos de rendering solucionan este problema de forma de implícita como es el caso del método de *trazado de rayos*, donde por la propia naturaleza del método, las superficies ocultas no se representan nunca, aunque puedan influir en la iluminación global de la escena. A continuación comentamos brevemente alguno de estos métodos.

El **algoritmo de buffer de profundidad** es el método imagen-espacio más extendido. El buffer de profundidad, o z-buffer, es una matriz que almacena las coordenadas z del punto del objeto que se ve en cada pixel. En cada punto (x, y) del plano de visión, solamente es visible la superficie con z mínima, la que está más cerca.

La figura 6 ilustra el algoritmo. Si llamamos Z al z-buffer y FB (del inglés *frame buffer*) a una matriz que almacena el color que corresponde a cada pixel, el algoritmo se detalla como:

- **Paso 1** Inicializar el z-buffer.

$Z(i, j) = 1$; $FB(i, j) = \text{Mínimo o color de fondo.}$

Hacemos $Z(i, j) = 1$ porque asumimos que hemos realizado la transformación de volumen canónico por lo que los valores de z están entre 0 y 1, 0 el valor más cercano y 1 el valor más lejano.

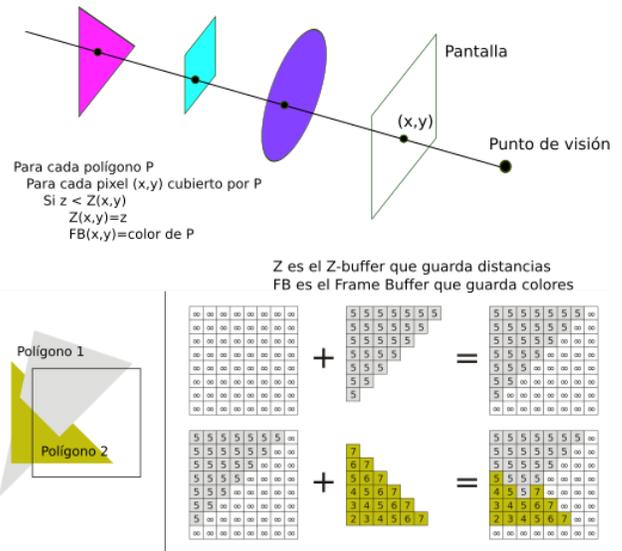


Figura 6: Esquema explicativo del algoritmo del buffer de profundidad (imagen de V. Marie dominio público vía Wikipedia).

- **Paso 2** Para cada superficie, comparar los valores de z con los almacenados en el buffer.

- Calcular $z(x, y), \forall (x, y) \in \text{Superficie}$
- Si $z(x, y) < Z(i, j)$ correspondiente, entonces $Z(i, j) = z(x, y)$
 $FB(i, j) = \text{color de superficie en } z(x, y)$

Después del algoritmo, Z contiene los valores de z para las superficies visibles; $FB(i, j)$ suele ser la memoria gráfica o que representa la imagen en la pantalla.

Este algoritmo presenta la ventaja de ser fácil de implementar y que no exige ordenación de las superficies. La principal desventaja es la necesidad de almacenamiento en memoria de $Z(i, j)$. El tiempo de procesamiento depende del número de superficies.

III. ILUMINACIÓN Y COLOR DE SUPERFICIES

Un modelo de iluminación sirve para asignar un valor de iluminación o color a cada punto de cada superficie que compone la escena 3D que queremos representar. Para describir un modelo de iluminación necesitamos expresar una **ecuación de iluminación**. La ecuación de iluminación se expresa en términos de una serie de variables asociadas con los puntos que queremos iluminar. El proceso de evaluación de la ecuación de iluminación en uno o más puntos de un objeto dado se denomina en inglés **lighting**.

En esta sección veremos en detalle los modelos más empleados en de informática gráfica para este fin. Analizaremos cada uno de ellos, describiendo las peculiaridades que presentan y sus ventajas e inconvenientes.

III-A. Modelo de iluminación ambiental

Este es el modelo más simple. Se visualiza cada objeto empleando la iluminación que es intrínseca a él. Se trata de un modelo sin sombras, donde no hay fuentes de luz y donde cada objeto ofrece unas características de iluminación

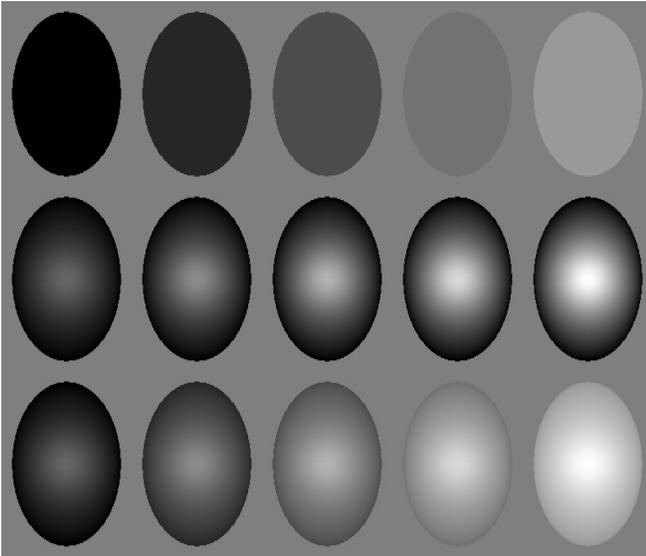


Figura 7: Comparación de variaciones de reflexión difusa y ambiental: La fila superior representa el ovoide iluminado sólo con luz ambiental, $k_a = 0, 0,15, 0,30, 0,45, 0,60$; en la fila intermedia sólo hay reflexión difusa sin emplear iluminación ambiental, $k_d = 0,4, 0,55, 0,70, 0,85, 1,00$; y en la fila inferior fijamos $I_a = I_p = 1$, $k_d = 0,4$ y variamos $k_a = 0, 0,15, 0,4, 0,55, 0,70$ de siempre de izquierda a derecha.

independientes del entorno en el que se sitúa. Los efectos de este modelo son insatisfactorios en la mayoría de aplicaciones ya que los objetos aparecen carentes de toda sensación de realidad.

El modelo se denomina de iluminación ambiental y la ecuación que lo describe es:

$$I = k_i, \quad (1)$$

donde I es la intensidad resultante en los puntos de la superficie y el coeficiente k_i es la intensidad o color intrínseco del objeto o superficie i .

Consideramos en este modelo que existe una fuente de luz suave, no direccional, que es producto de las múltiples reflexiones de la luz en los múltiples objetos del entorno. Esta es la **luz ambiental**, que como es igual en todos los objetos de la escena, el modelo se transforma en

$$I = I_a \cdot k_a, \quad (2)$$

Donde I_a es la intensidad de la luz ambiental, supuesta constante para todos los objetos. k_a se denomina **coeficiente de reflexión ambiental** y es la cantidad de luz ambiental reflejada por un objeto. k_a es un valor cuyo rango está entre 0 y 1, de manera que puede interpretarse como la cantidad de toda la iluminación de la escena I_a que refleja el objeto. El coeficiente de reflexión ambiental es una propiedad del material del objeto, mientras que la intensidad de la luz ambiental es una propiedad de la escena. La figura 7, muestra el efecto de cambiar k_a . Esta figura muestra que, si sólo se usa el modelo de iluminación ambiental, las figuras aparecen

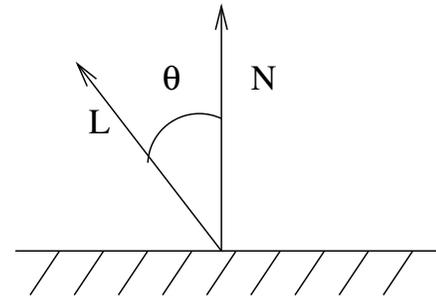


Figura 8: Factores presentes en el modelo de reflexión difusa

planas porque el color es el mismo en todos los puntos de la superficie.

III-B. Reflexión difusa

Los objetos iluminados con el modelo de luz ambiental tienen la misma iluminación a lo largo de toda la superficie. Pero cuando iluminamos un objeto con una fuente de luz apreciamos que la iluminación del objeto varía a lo largo de su superficie en función del ángulo de la normal a la superficie con respecto a la posición de la fuente de luz. Para representar este hecho introducimos los modelos de reflexión difusa.

Las superficies mate tienen reflexión difusa o reflexión Lambert. Estas superficies aparecen igualmente brillantes desde todos los ángulos de visión, porque la luz se refleja con igual intensidad en todas las direcciones. Para una superficie dada, su brillantez sólo depende del ángulo que forman la dirección de incidencia de la fuente de luz \vec{L} y del ángulo que forma \vec{L} con la normal de la superficie \vec{N} (ver figura 8)

Cuanto menor sea el ángulo de incidencia con respecto a la normal, mayor será la iluminación. Para modelar este hecho, se describe la ecuación:

$$I = I_p k_d \cos \theta = I_p k_d (\vec{N} \cdot \vec{L}). \quad (3)$$

Donde I_p es la intensidad de la fuente de luz; k_d es una constante entre 0 y 1 llamada **coeficiente de reflexión difuso** cuyo valor depende del material y que representa la proporción de la luz incidente que el objeto puede reflejar. El ángulo θ debe estar entre 0 y 90 grados. \vec{N} y \vec{L} han de estar normalizados (valores entre 0 y 1).

Si la fuente de luz está lo suficientemente alejada, el ángulo que forma la luz incidente con la superficie es el mismo en cada punto de la misma. En este caso la luz se denomina **fuentes de luz direccional**.

Para conseguir mayor realismo se conjugan los modelos de reflexión difusa y ambiental en un solo modelo:

$$I = I_a k_a + I_p k_d (\vec{N} \cdot \vec{L}) \quad (4)$$

En la figura 7 se muestra el efecto de la variación del factor de reflexión difusa. Incrementando k_d se consiguen objetos más claros porque reflejan mayor cantidad de la iluminación que les llega de la fuente de luz I_p . En el punto central la iluminación es mayor porque el ángulo que forman los vectores \vec{L} y \vec{N} es cero ya que la fuente de iluminación está en la posición del punto de visión y los vectores normales emanan

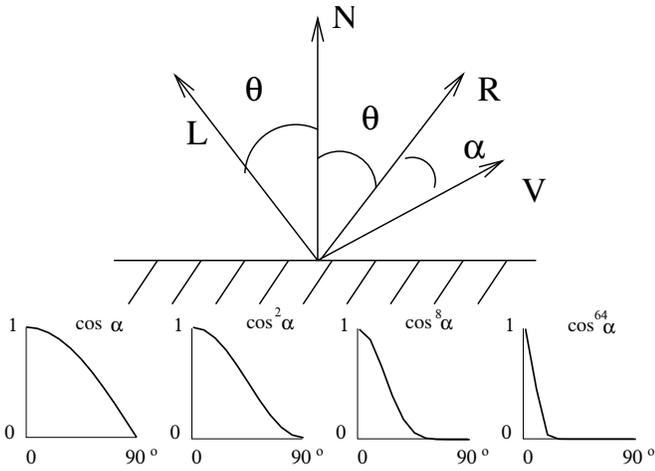


Figura 9: Factores presentes en el modelo de reflexión especular. Diferentes valores de $\cos^n \alpha$ empleados en el modelo de Phong.

desde el centro de la figura. En los contornos de las figuras, la iluminación es cero cuando no hay iluminación ambiental porque el ángulo que forman \vec{L} y \vec{N} es cero o próximo a cero.

Emplear un modelo de iluminación que haga depender la iluminación de la superficie del ángulo que forman \vec{L} y \vec{N} es fundamental para dar sensación de volumen a los objetos y evitar que parezcan planos.

III-C. Superficies coloreadas

Hasta aquí hemos discutido el caso de superficies sin color que queremos representar con luz acromática. Para tratar el caso de superficies coloreadas, debemos tratar por separado cada componente de color. Representamos el color difuso de un objeto con un valor O_d para cada componente de color. Si empleamos el modelo RGB, tendremos la tupla (O_{dR}, O_{dG}, O_{dB}) , que define las componentes difusas para un objeto determinado. La fuente de iluminación, si proyecta luz cromática, tendrá también sus respectivas componentes primarias de iluminación (I_{pR}, I_{pG}, I_{pB}) que se reflejan en proporciones de $(k_d O_{dR}, k_d O_{dG}, k_d O_{dB})$ respectivamente. Las componentes RGB del color de la superficie se calcularían empleando las ecuaciones:

$$\begin{aligned} I_R &= I_{aR} k_a O_{dR} + I_{pR} k_d O_{dR} (\vec{N} \cdot \vec{L}) \\ I_G &= I_{aG} k_a O_{dG} + I_{pG} k_d O_{dG} (\vec{N} \cdot \vec{L}) \\ I_B &= I_{aB} k_a O_{dB} + I_{pB} k_d O_{dB} (\vec{N} \cdot \vec{L}) \end{aligned} \quad (5)$$

El uso de los coeficientes que escalan cada componente permite controlar la cantidad de luz ambiente o difusa sin alterar las proporciones de sus componentes. Podemos generalizar el modelo especificando, no una ecuación para cada componente de color, sino una ecuación genérica para cada longitud de onda. Si añadimos el factor de atenuación de la luz emitida, que tenga en cuenta la distancia entre el objeto y la fuente de iluminación, la ecuación quedaría como:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} k_d O_{d\lambda} (\vec{N} \cdot \vec{L}) \quad (6)$$

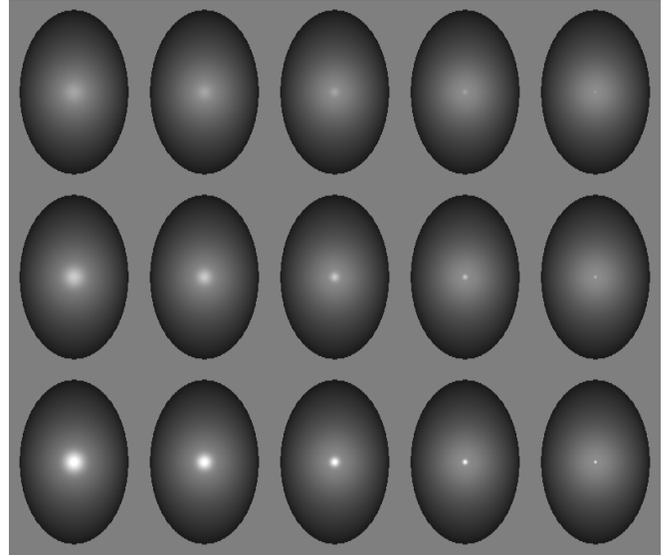


Figura 10: Comparación de variaciones en el modelo de Phong: para todas las ovoides $k_a = 0,1$, $I_a = I_p = 1,0$, $k_d = 0,45$; de izquierda a derecha, $n = 3,0, 7,0, 19,0, 67, 259$; de arriba a abajo, $k_s = 0,1, 0,25, 0,5$

III-D. Reflexión especular

La reflexión especular puede observarse en los objetos con brillo. Cuando iluminamos una bola de billar nueva, observamos que en ella aparece el punto de luz del objeto que la ilumina. Dicho brillo resulta no ser del color de la bola sino del color de la luz con que se ilumina. Si se mueve el punto de visión observamos que el reflejo también se mueve. Esto es así porque la componente especular no se refleja en la misma cantidad en todas las direcciones y depende del punto de visión. Recoger estos factores supone crear un nuevo modelo de iluminación.

El modelo empleado para resolver la reflexión especular es el **modelo de Phong**. En este modelo entran en juego los cuatro factores que aparecen en la figura 9: la orientación de la superficie \vec{N} , la dirección de incidencia del rayo de luz \vec{L} , la dirección del rayo reflejado \vec{R} y la dirección del punto de observación \vec{V} .

El modelo supone que la máxima reflexión especular ocurre cuando α es cero, y que su valor cae rápidamente cuando α crece. Además, hace depender la rapidez de caída del material del objeto empleado. La rapidez de caída viene dada por el factor $\cos^n \alpha$, donde n es el **exponente de reflexión especular** que depende del material empleado. La figura 10 muestra la evolución del factor con n .

La cantidad de luz incidente que se refleja especularmente depende del ángulo de incidencia θ . Si k_s es la fracción de la luz reflejada especularmente el modelo queda:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} \cos \theta + k_s \cos^n \alpha]. \quad (7)$$

Normalmente k_s se considera constante y se hace depender del material. En este caso hablamos del **coeficiente de reflexión especular**, cuyo rango varía entre 0 y 1. El valor

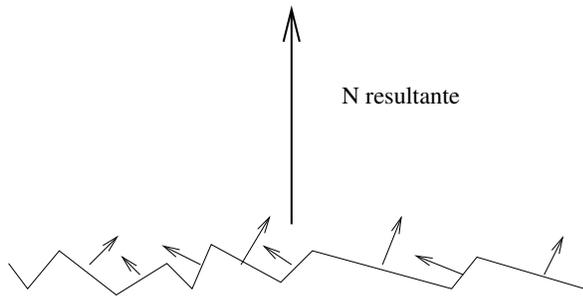


Figura 11: Modelo de micro facetas

de k_s se selecciona empíricamente para producir resultados satisfactorios. Si normalizamos vectores, la ecuación queda:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} (\bar{N} \cdot \bar{L}) + k_s (\bar{R} \cdot \bar{V})^n]. \quad (8)$$

El modelo entiende que la reflexión especular no dependerá en absoluto del color del objeto considerado. Para incluir el hecho de que el color o material de la superficie pueda influir en la reflexión especular, modificamos el modelo:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} (\bar{N} \cdot \bar{L}) + k_s O_{s\lambda} (\bar{R} \cdot \bar{V})^n]. \quad (9)$$

donde $O_{s\lambda}$ es el color especular del objeto. La figura 10 muestra un conjunto de esferas iluminadas empleando este modelo.

III-E. Modelos que emplean propiedades físicas

Los modelos que hemos descrito hasta ahora intentan aproximarse lo más posible a la realidad. Para ello no hacen uso de ninguna propiedad física sino más bien del sentido común. Desde luego no podemos afirmar que los modelos no sean buenos o que no sean útiles, pero sí que no se fundamentan en propiedades físicas de la luz y de la materia por otro lado conocidas y estudiadas desde mucho tiempo atrás.

Existen un grupo de modelos que se apoyan en propiedades físicas de la iluminación para formular los valores de iluminación en los puntos de la escena. El más famoso de todos ellos es el de **Cook Torrance**. No vamos a entrar en profundidad en el estudio de este algoritmo; simplemente comentaremos algunas generalidades sobre el mismo, introduciendo al lector en la problemática de los modelos basados en propiedades físicas.

Este modelo considera que las fuentes de luz emiten en una dirección un flujo de intensidad radiante. Este flujo se transforma en una energía luminosa sobre la superficie del objeto incidido. Esta energía o **irradiancia** se hace depender de la intensidad emitida por la fuente, y del ángulo de incidencia de la luz con respecto a la normal.

Pero la irradiancia no es la luz que nosotros vemos. De toda la irradiancia parte de ella se transforma en una intensidad reflejada que es la que nos llega a nosotros. La relación entre

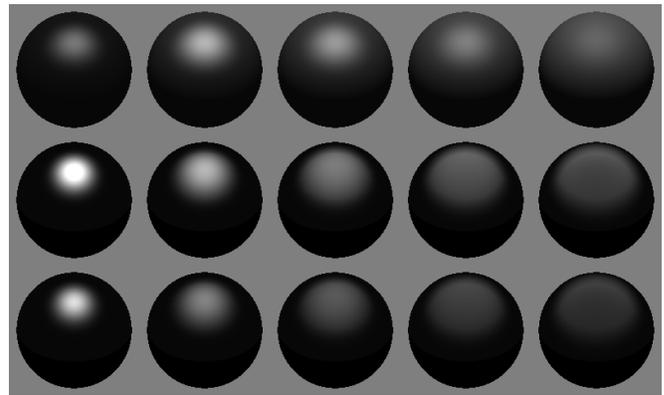


Figura 12: Comparación del modelo de Cook-Torrance y el de Phong. Las esferas de la fila superior se han representado empleando el modelo de Phong haciendo $I_a=0.2$, $k_a=0.15$, $k_d=.1, .15, .2, .25, .3$, $k_s=.7, .55, .4, .25, .1$, $I_p=1$, $n=3$. La fila intermedia se ha realizado empleando el modelo de Cook-Torrance anulando el término de Fresnel y haciendo $I_a=0.2$, $k_a=0.15$, $R_d=.7$, $d=0$, $s=1$, $G=0.3, 0.4, 0.5, 0.6$ y 0.7 . La fila de abajo emplea el modelo de Cook-Torrance y emplea un factor $F=0.7$.

la irradiancia E_i y la intensidad reflejada I_r viene dada por la **reflectividad bidireccional** ρ :

$$\rho = \frac{I_r}{E_i} \quad (10)$$

Además el factor de reflectividad bidireccional se descompone a su vez en un componente difuso y otro especular:

$$\rho = k_d \rho_d + k_s \rho_s \quad (11)$$

A la hora de calcular la componente especular, se emplea un modelo de la superficie conocido como de **microfacetas**. Según este modelo la superficie está compuesta por otras micro superficies cuya orientación es aleatoria (figura 11).

El coeficiente ρ_s viene dado por la expresión:

$$\rho_s = \frac{F_\lambda}{\pi} \frac{DG}{(\bar{N} \cdot \bar{V})(\bar{N} \cdot \bar{L})} \quad (12)$$

donde D es la distribución de probabilidad de las micro facetas; G es un factor que mide el sombreado entre micro facetas; F_λ es el factor de Fresnel que hace depender ρ_s del valor de la longitud de onda de la luz incidente y del ángulo de incidencia de la luz; $(\bar{N} \cdot \bar{V})$ porque es proporcional al área que el observador puede ver y $(\bar{N} \cdot \bar{L})$ porque es proporcional al área que la fuente de luz ve.

Empleando este modelo se consiguen efectos muy próximos a la realidad. Consiguen eliminar la apariencia plástica de los objetos iluminados con el modelo de Phong. La contrapartida es el coste computacional.

IV. SOMBREADO DE SUPERFICIES

En la sección anterior hemos visto cómo podemos asignar un valor de iluminación a cada punto de las superficies de la escena. Los objetos tridimensionales suelen estar formados por poligonales.

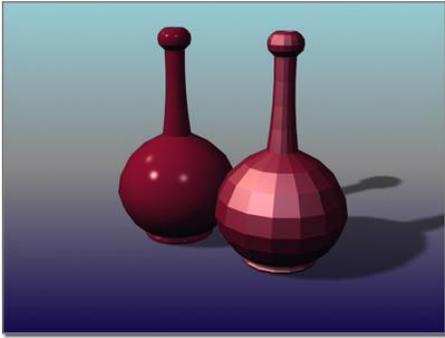


Figura 13: El sombreado constante puede ser suficientemente bueno si el tamaño de los poligonales es lo suficientemente pequeño (imagen propiedad de Autodesk 3DS).

Con lo expuesto, deberíamos calcular el valor de intensidad luminosa en cada punto de cada polígono, lo cual es tremendamente costoso en tiempo de computación. En este apartado veremos cómo esto no es necesario, sino que basta con aplicar la fórmula del modelo de iluminación una vez para cada polígono y después aplicar procedimientos de interpolación para asignar una intensidad al resto de puntos. Se consigue con esto reducir significativamente el tiempo de computación a la vez que podemos producir ilusión de suavizado en las aristas que unen dos poligonales.

IV-A. Sombreado de intensidad constante

En este método, también conocido como *sombreado plano*, se calcula un valor de intensidad para cada polígono. Todos los puntos del polígono se representan con el mismo valor de intensidad. Es el peor de los casos porque al representar cada polígono con un valor de intensidad diferente las aristas que separan los poligonales se notarán claramente en la escena.

Para un objeto 3D formado por poligonales, el método de sombreado plano ofrece una representación de calidad suficiente sólo si se cumple estas condiciones:

1. El objeto no se aproxima con ninguna superficie curva.
2. Las superficies se representan sin texturas.
3. Las fuentes de luz están lo suficientemente lejos ($\vec{N} \cdot \vec{L} = \text{cte}$).
4. El punto de visión también ($\vec{V} \cdot \vec{R} = \text{cte}$).

Si existe una superficie curva, se puede aproximar con un conjunto de poligonales, cada una con intensidad constante. Si la fuente o el punto de visión no están lo suficientemente alejados, puede reducirse el tamaño de las poligonales y calcular la intensidad de cada faceta en el centro del polígono (ver figura 13).

La ventaja principal de este método de representación es la rapidez. La contrapartida es que si la orientación de los planos adyacentes cambia bruscamente, se producen cambios no deseables de intensidad.

IV-B. Sombreado de Gouraud

En este método también se ajustan las superficies por poligonales, pero se interpola linealmente la intensidad en cada plano de forma que en las fronteras no haya cambio brusco.

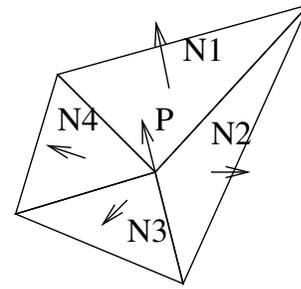


Figura 14: Cálculo del vector P_v en el sombreado de Gouraud. El vector que representa la normal en un vértice P_v se calcula como la media de los vectores normales en las superficies que comparten dicho vértice.

Para ello se realizan los siguientes cálculos:

1. Se determina el vector normal unitario promedio en cada vértice del polígono (\vec{P}_v).
2. Se aplica un modelo de iluminación en cada vértice para calcular la intensidad del vértice.
3. Se interpolan de manera lineal las intensidades de vértice sobre la superficie del polígono.

El vector normal unitario promedio en cada vértice se obtiene al promediar las normales de la superficie de todos los poligonales que tienen ese vértice en común.

$$\vec{P}_v = \frac{\sum_{k=1}^n \vec{N}_k}{|\sum_{k=1}^n \vec{N}_k|} \quad (13)$$

Donde n es el número de poligonales coincidentes en el vértice v y \vec{N}_k son los vectores normales de cada uno de los poligonales. Con los vectores \vec{P}_v de cada polígono aplicamos un modelo de iluminación y obtenemos un valor de intensidad en cada vértice. La intensidad en los puntos de cada arista se interpola de forma lineal entre las intensidades de los vértices, como se muestra en la figura 15.

$$I_4 = I_1 - (I_1 - I_2) \frac{y_4 - y_2}{y_1 - y_2} \quad (14)$$

Para calcular la intensidad en cualquier punto, interpolamos a través de una línea de rastreo en la dirección horizontal, entre los cortes con las aristas.

$$I_p = I_5 - (I_5 - I_4) \frac{x_p - x_4}{x_5 - x_4} \quad (15)$$

Este algoritmo puede ser fácilmente integrado con el algoritmo de superficies ocultas para rellenar poligonales visibles a lo largo de una línea de rastreo. Se pueden aplicar los métodos de reducción de operaciones empleando métodos ya conocidos.

El método presenta deficiencias al representar puntos de luz sobre las superficies. En ocasiones se representan con formas anormales o con bandas brillantes y oscuras por interpolar la intensidad.

IV-C. Sombreado de Phong

Este método en lugar de interpolar la iluminación interpola los vectores normales y luego aplica sobre ellos el cálculo de la iluminación.

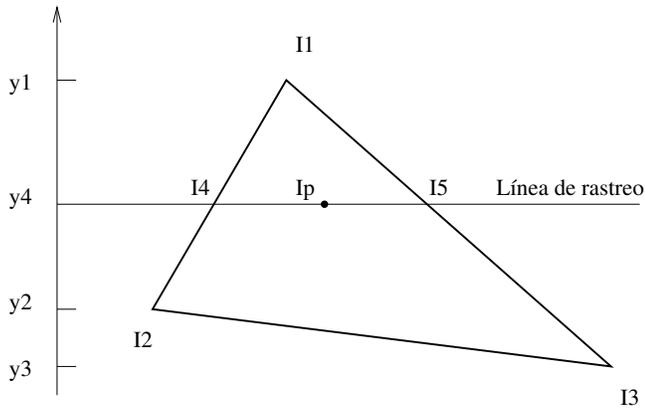


Figura 15: Interpolación de intensidades a lo largo de la línea de rastreo. El valor I_4 se obtiene interpolando los valores de I_1 y de I_2 en y , el valor I_p se obtiene interpolando en x los valores de I_4 e I_5 .

Para representar una superficie poligonal se llevan a cabo los siguientes pasos:

- Se determina el vector normal unitario promedio en cada vértice.
- Se interpola de manera lineal las normales de los vértices en la superficie del polígono.
- Se aplica un modelo de iluminación a lo largo de cada línea de rastreo para calcular las intensidades de los pixels que se proyectan para los puntos de la superficie empleando los vectores interpolados.

En la figura 15 se ilustra el modo de proceder a la hora de interpolar vectores normales. Los cálculos se realizan empleando la expresión:

$$N = N_1 - (N_1 - N_2) \frac{y - y_2}{N_1 - N_2} \quad (16)$$

Mediante este método se producen modelos más realistas, pero el tiempo de cálculo se incrementa considerablemente.

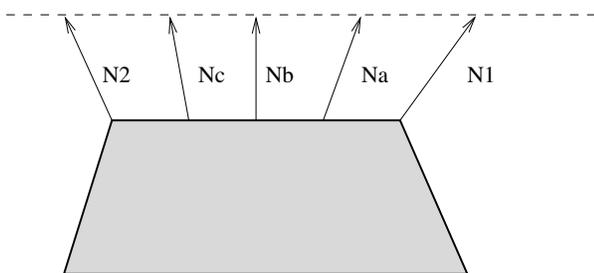


Figura 16: Cálculo del vector \vec{N} en el sombreado de Phong. Dados los vectores \vec{N}_1 y \vec{N}_2 en los vértices de un polígono, los vectores de los vectores \vec{N} en las aristas se obtienen por interpolación de los de los vértices. Con los vectores de las aristas se pueden obtener los vectores en los puntos de los polígonos empleando el procedimiento descrito en la figura 15.

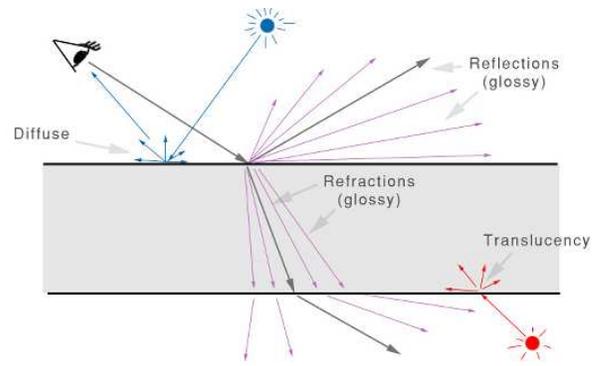


Figura 17: Diferentes efectos de interacción de la luz con una superficie (imagen propiedad de Autodesk 3DS).

V. REFLEXIONES

Con lo que hemos visto hasta ahora, el aspecto final de una superficie está determinado por su componente difusa, responsable del color soporte, y por su componente especular responsable de los **destellos** provocados los reflejos de las fuentes de luz (en inglés *highlights*).

Sin embargo, cuando la luz ilumina un objeto, la luz interactúa con la superficie de múltiples maneras:

1. Parte de la luz es **absorbida** por el material para dar sensación de color siendo la componente difusa del color del material;
2. Parte es **transmitida** a través de él dependiendo de su grado de transparencia u opacidad;
3. Parte es **reflejada** de forma especular;
4. además, la luz puede ser **dispersada** (en inglés *scattered*) tanto al ser reflejada como al ser o al ser transmitida.

La figura 17 ilustra estas componentes. Un modelo de materiales de superficie que pretenda ser realista que considere estas **cinco componentes** y, además, considerar que la magnitud de cada una de las componentes, en su conjunto, debe cumplir el principio de conservación de la energía, de manera que el total de luz incidente debe ser igual al total de luz absorbida, transmitida y reflejada. De esta manera, la intensidad del haz incidente I_0 debe ser igual a la intensidad de los haces transmitidos I_T , absorbidos I_A y reflejados I_R , de manera que $(I_0 = I_T + I_A + I_R)$. De forma alternativa esta ecuación se expresa como $T + A + R = 1$ siendo T, A y R la transmitancia (I_T/I_0) la absorbancia o extinción I_A/I_0 y la reflectancia o reflectividad (I_R/I_0)[1].

Los materiales capaces de transmitir luz con relativamente poca absorbancia y reflexión se denominan **transparentes**. Los materiales son **translúcidos** si dejan pasar la luz parcialmente, la luz se dispersa en su interior de manera que los objetos que están detrás no se ven nítidamente. Los materiales que no dejan pasar la luz son **opacos**. En este apartado vamos a hablar de las componentes reflejada tanto de forma especular como dispersada. En el siguiente apartado hablamos de la componente transmitida y parcialmente dispersada durante la transmisión.

Relacionado con la cantidad de luz que se refleja tanto de forma especular como de forma dispersada está el concepto de

glossiness cuya traducción al español podríamos relacionarla con el grado de pulido del objeto. Un valor de *glossiness* elevado implica que los reflejos estarán bien matizados, mientras que un valor bajo implicará que los destellos se emborronan. El grado de *glossiness* depende de la topografía de la superficie. El modelo de microfacetas explica por qué no toda la luz que incide en una superficie se refleja en una misma dirección. En la sección III-E presentamos un modelo de reflexión que utiliza estas una serie de parámetros para modelar la rugosidad de la superficie y el efecto de dicha rugosidad en la distribución de la luz reflejada. Esta rugosidad es responsable de la dispersión en la luz reflejada.

El material sustrato influye en el color del material al absorber luz y también en el nivel de reflexión especular que se observa en el mismo. Así por ejemplo, un objeto no metálico, bien pulido o cubierto por una capa o revestimiento adecuado aparecerá con brillos y reflejos bien definidos. El color del objeto dependerá de las propiedades de color difuso y el color de los reflejos se programa de forma independientemente pudiendo establecerse un color particular, una rugosidad (inverso de *glossiness*) y un índice de refracción (IOR). Cuando la radiación lumínica pasa de un medio a otro que tiene un índice de refracción diferente, parte de la luz se difunde en la intercara de los dos medios, incluso cuando sean transparentes. La reflectancia R mide la proporción de la luz incidente que se refleja y es función del IOR de ambos medios y del ángulo de incidencia como veremos en la siguiente sección.

Cuando un objeto es **metálico** la opacidad funciona de forma diferente. Los objetos metálicos no tienen color difuso sino un color base. La radiación incidente es absorbida totalmente en una fina capa exterior. La mayor parte de la radiación absorbida se vuelve a emitir en forma de luz visible de la misma longitud de onda y aparece como luz reflejada. El color percibido está determinado por la distribución de longitudes de onda reflejada y no absorbida, que son generalmente iguales a los de la luz incidentes. La plata y el aluminio tienen un comportamiento reflector total. El cobre y el oro tienen color porque algunas longitudes de onda son absorbidas. En algunos paquetes informáticos se puede determinar si un objeto es metálico o no estableciendo una propiedad llamada *metalness*.

Para representar cómo se distribuye la luz que incide en un determinado material se emplea la **BRDF** (Bidirectional Reflectance Distribution Function). Cada material tiene una función BRDF que describe cómo se refleja la luz teniendo en cuenta desde qué punto es iluminada y desde qué punto es observada. La función BRDF se mide empíricamente para cada material empleando dispositivos específicos que proyectan un haz de luz sobre un material desde diferentes ángulos de incidencia y miden la cantidad de luz reflejada también en los ángulos de reflejo (figura 18). El resultado es una representación muy precisa de la forma en la que se refleja la luz en el material. La función BRDF mide la luz reflejada, como ya hemos indicado, la luz no reflejada es absorbida, transmitida o dispersada.

Si la reflexión no sufre cambios a medida que la superficie rota sobre su normal el material es **isotrópico**. Si la reflexión cambia, entonces estamos hablando de materiales **anisotrópico**. La anisotropía se produce porque las microfacetas se



Figura 18: La figura en la parte superior muestra un dispositivo empleado para medir la función BRDF de un material. Se emite un rayo de luz desde todas las direcciones posibles y se mide la cantidad de luz reflejada (imagen propiedad de E. Erin CC BY-SA vía Wikipedia).

organizan en surcos o estrías (figura 19). Esto afecta a cómo se reflejan los rayos de luz de manera que los brillos se extienden en una dirección dada. El aluminio cepillado o el terciopelo son buenos ejemplos de este tipo de materiales.

VI. TRANSPARENCIAS

Cuando la luz incide en un material con cierto grado de transparencia, parte de la luz se refracta, atravesando el nuevo material sobre el que incide y cambiando su dirección. El **índice de refracción** (IOR) permite medir el cambio de dirección del haz de luz al cambiar de medio. La **ley de Snell** establece la expresión:

$$n_i \sin(\theta_i) = n_t \sin(\theta_t) \quad (17)$$

donde n_i y n_t son los índices de refracción (IOR) del medio desde el que incide la luz y en el que se transmite la luz respectivamente. θ_i y θ_t son los ángulos de incidencia y de transmisión respectivamente. La figura 20 muestra estos valores sobre un ejemplo. También muestra como superado un cierto umbral ($\theta_t = 90$) sólo hay reflexión, no hay refracción. En la figura 21 se muestran los efectos de representar un objeto con diferentes valores de IOR. Cuando el IOR es el mismo en los

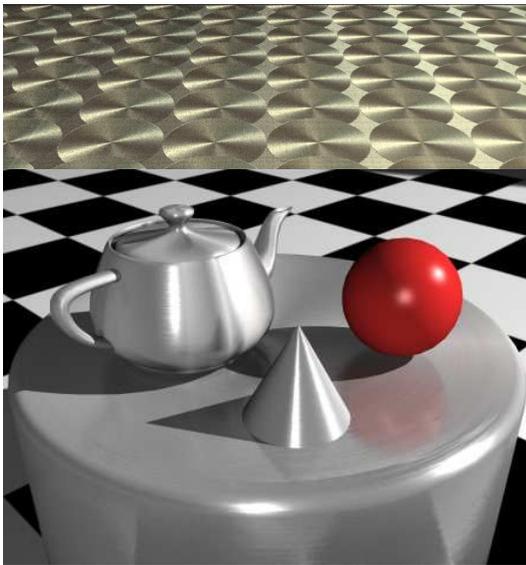


Figura 19: Ejemplo de superficies con reflejos anisotrópicos (imagen propiedad de Autodesk 3DS).

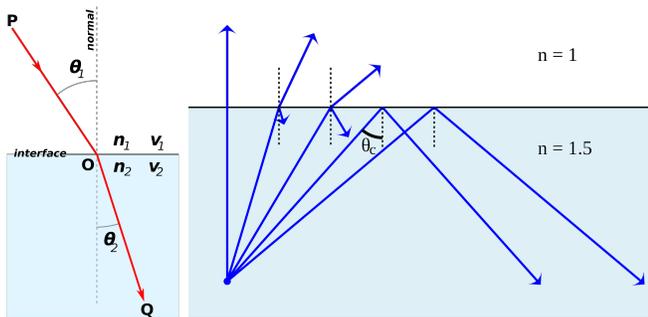


Figura 20: Refracción y refracción de un haz de luz al cambiar de medio (imágenes propiedad de O. Alexandrov y L. Havalund CC BY-SA vía Wikipedia).

dos medios, la luz no se desvía y el efecto es que el material atravesado parece invisible.

En la sección anterior hemos dicho que la cantidad de luz reflejada también depende del IOR. La expresión de indica la fracción de luz polarizada que se refleja viene dada por la **ecuación de Fresnel**:

$$r_p = \frac{n_t \cos(\theta_i) - n_i \cos(\theta_t)}{n_t \cos(\theta_i) + n_i \cos(\theta_t)} \quad ; \quad R_p = r_p^2 \quad (18)$$

La fracción de luz reflejada es R_p y la fracción de luz transmitida polarizada es $T_p = 1 - R_p$. La ecuación se completa con una componente R_s ortogonal a R_p con la que se podría representar cualquier tipo de luz independiente de su grado de polarización.

Relacionado con la refracción está el concepto de **translucencia**. La translucencia modela la dispersión de la luz que atraviesa un objeto (ver figura 22 para un ejemplo). Suele usarse para modelar cortinas, mamparas de luminarias o cristales rugosos. Como los objetos transparentes, un objeto translucido deja pasar la luz, pero a diferencia de los objetos transparentes, la luz se dispersa de manera que lo que hay



Medium	Refractive index
Vacuum	1
Air	1.00
Water	1.33
Alcohol	1.36
Sugar solution (80%)	1.49
Perspex	1.50
Glass	1.50-1.70
Diamond	2.42

Figura 21: Efectos del IOR en la imagen transmitida. Izquierda: IOR=1.0, centro: IOR=1.2 y derecha IOR=1.5 (imágenes propiedad de Autodesk 3DS).



Figura 22: Efecto de la translucencia. El cristal de la derecha tiene translucencia verde (imagen propiedad de Autodesk 3DS).

detrás no puede verse claramente. Para representar la translucencia con precisión puede emplearse una función BTDF (*bidirectional transmited distribution function*) que se obtiene forma similar a como lo hace la BRDF. Con estas funciones podemos contralar la parte de la luz reflejada (BRDF) y la parte de la luz transmitida que es dispersada. La figura 23 muestra la relación de estos términos.

La cantidad de luz no reflejada puede ser absorbida en caso de superficies opacas, generando color. De igual manera, puede haber objetos en los que no haya ni refracción ni dispersión cuando son atravesados por la luz pero sí absorción y transmisión como es el caso de las lentes de color. En los objetos tipo mampara, no hay refracción y sí hay absorción y

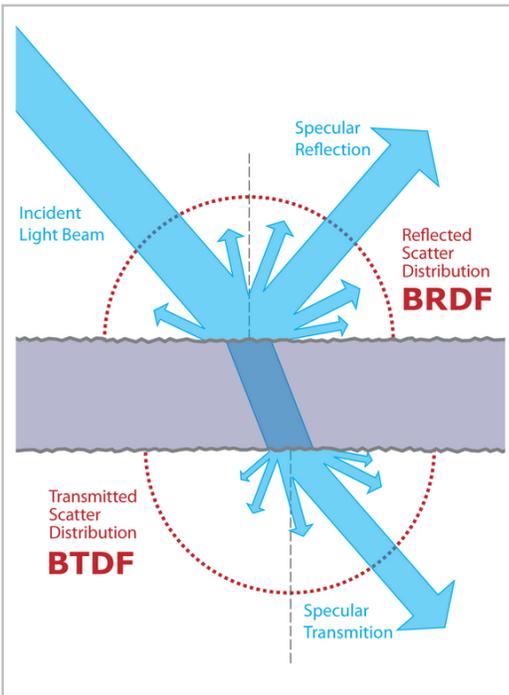


Figura 23: La figura muestra el contraste en entre la BRDF y la BTDF (imagen propiedad de Autodesk 3DS).

dispersión.

VII. MÁS INFORMACIÓN

Los libros básicos de referencia sobre informática gráfica encontramos capítulos que dan cobertura a los puntos tratados en este tema:

Con respecto a la eliminación de superficies ocultas, [2] lo trata en profundidad en el capítulo 36. En [3] se reserva el capítulo 13.

Con respecto a cuestiones relativas a la reflexión y refracción, [2] dedica el tema 26 a tratar cuestiones relativas a la luz y el capítulo 27 a cuestiones relativas a sombreado. En [3] se reserva el capítulo 14 a tratar temas relativas a modelos de iluminación y de presentación de superficie.

Watt [4], [5] dedica los primeros capítulos a estos temas.

Se pueden encontrar tutoriales de distintos productos de modelado y rendering 3D donde se explican en profundidad estos conceptos. Por ejemplo, en este tutorial hay una buena explicación del concepto de *metalness* <https://www.chaosgroup.com/blog/understanding-metalness>

EJERCICIOS Y CUESTIONES

1. Describe el papel de las componentes de iluminación ambiental en el modelo de iluminación difusa de superficies.
2. Describa el método de sombreado de *Gouraud*.
3. Indique en qué consiste la técnica de sombreado plano.
4. Describa el efecto del exponente de reflexión especular en el modelo de iluminación de *Phong*.
5. Describa el efecto del coeficiente de reflexión especular en el modelo de *Phong*.

6. En la literatura de informática gráfica nos encontramos con el modelo de iluminación de *Phong* y con el modelo de sombreado de *Phong*. Indique (sin explicarlos) a qué hace referencia cada uno.
7. Enumere las etapas del algoritmo de sombreado de *Phong*.
8. Enumere tres técnicas de sombreado.
9. ¿Para qué sirven las técnicas de supresión de superficies ocultas? Ilústrelas con un ejemplo.
10. Usted quiere hacer el rendering de una cacerola de acero inoxidable. Sabe que tiene que asignar un valor a los coeficientes de reflexión ambiental k_a , difusa k_d y especular k_s a la superficie de dicha cacerola. Describa el modelo de intensidad que tiene en cuenta esos tres parámetros e indique qué valores de k_a , k_d , k_s emplearía para el caso de la cacerola.
11. Escriba las ecuaciones del modelo de reflexiones difusas para superficies coloreadas indicando el significado de cada uno de los términos.
12. Detalle la fórmula del modelo de iluminación de *Phong* para superficies coloreadas: comente sus elementos.
13. Indique a qué se refiere el producto $(\mathbf{R} \cdot \mathbf{V})$ en el modelo de reflexión especular.
14. Cada uno de los 4 conjuntos esfera-cilindro que se representa en la figura 24 (parte superior) ha sido renderizados utilizando el modelo de *Phong*. Las diferencias entre los conjuntos de deben a que en uno de los conjuntos se ha alterado k_a , en otro de los conjuntos k_s y en otro k_s y n . Identifique estos tres casos y justifique su respuesta.

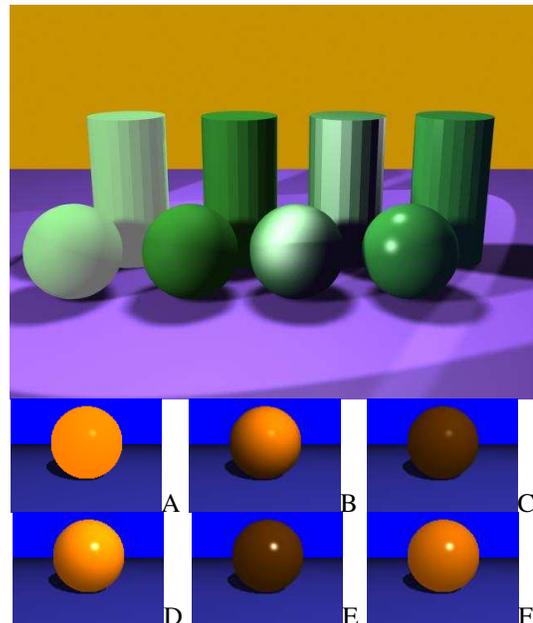


Figura 24: Figura con ejemplos de modelo de iluminación.

15. La esfera que se renderiza en la figura 24 (parte inferior) ha sido generada variando los parámetros k_a , k_d y k_s de un determinado modelo de iluminación. Describa el dicho modelo. Indique qué valores de los que se listan a continuación se corresponden con cada una de las imagen: $(k_a = 0,2, k_d = 0,2, k_s = 0,2)$ $(k_a = 0,2, k_d =$

$1, k_s = 0,2)(k_a = 1, k_d = 0,2, k_s = ,2)(k_a = 0,5, k_d = 0,5, k_s = 1)(k_a = 0,5, k_d = 1, k_s = 1)$

16. El objeto que se muestra en las imágenes de la figura 25 (parte superior) ha sido modelado variando los parámetros de realismo k_a, k_d, k_s, I_p (obsérvese que hay dos fuentes de luz) del modelo de iluminación de *Phong*. La diferencia entre las dos escenas se consigue alterando sólo uno de dichos parámetros. Indique qué parámetro varía justificando la respuesta.

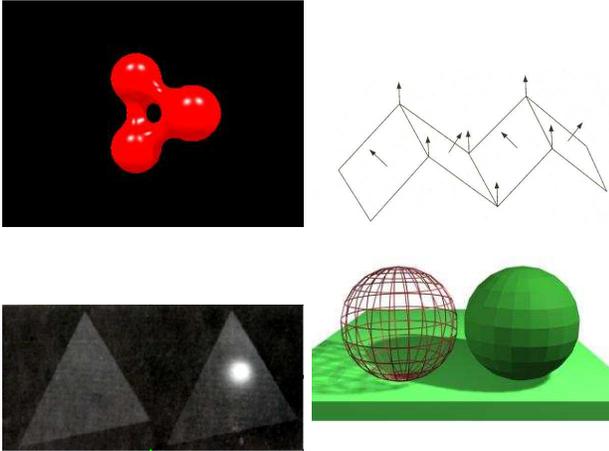


Figura 25: Figuras relacionadas con los ejercicios del tema de sombreado de superficies.

17. ¿Cómo se vería la imagen de la figura 25 (parte inferior derecha) empleando el sombreado de *Gouraud*?
18. En la figura 25 (parte inferior centro) se representan dos versiones de una poligonal renderizada empleando el modelo de sombreado de *Phong* y el de *Gouraud* respectivamente. En la versión renderizada con el sombreado de *Gouraud* no aparece el brillo interior ¿por qué?
19. Describa el modelo de sombreado que se emplea en el rendering de las esferas de la figura 25 (parte inferior izquierda). Con este mismo sombreado, ¿Qué haría para mejorar el aspecto de la esfera de la derecha?
20. Conocidas las fórmulas del modelo de iluminación de reflexión difusa y el interfaz de 3D Studio para manipular los materiales. Si modifica la propiedad *Diffuse* en el interfaz de 3D Studio, ¿Qué parámetros del modelo matemáticos se ven afectados?

REFERENCIAS

- [1] W. D. Callister, *Introducción a la ciencia e ingeniería de los materiales: Vol. I*. Reverté, 2020, vol. 2.
- [2] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [3] D. Hearn, M. P. Baker *et al.*, *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall., 2004.
- [4] A. Watt, *3D Computer Graphics*, tercera ed. Addison-Wesley, 2000.
- [5] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*. acm Press, Addison-Wesley, 1992.

Mapas y texturas sobre superficies

ÍNDICE

I.	Conceptos básicos	2
II.	Tipos de mapas	2
II-A.	Mapas tipo bitmap frente a mapas procedurales	2
II-B.	Mapas 2D frente a mapas 3D	2
III.	Proyección de mapas 2D	3
IV.	Mapas especiales	5
IV-A.	Mapas de abombamiento y desplazamiento	5
IV-B.	Mapas de opacidad	5
IV-C.	Mapas de entorno	6
	Referencias	7

Los modelos de iluminación emplean un mismo valor para todos los puntos de la superficie de cada polígono que forma los modelos 3D. Los modelos de sombreado de Gouraud y Phong eliminan las aristas de los polígonos mediante un suavizado de las transiciones. Sin embargo, rara vez consiguen generar un aspecto realista porque dan aspecto demasiado sintético a los objetos.

Imaginemos que queremos renderizar la pared de una habitación. Esta pared típicamente será una malla de poligonales, todos ellos coplanares. Si nos apoyamos en las poligonales podríamos pensar en asignar un color para las poligonales que representen la puerta, otro color para los poligonales que representen un cuadro en la pared etc... Empleando este método podríamos dividir a su vez la puerta en nuevas poligonales para ir asignando colores diferentes a los detalles de la puerta y seguir iterativamente hasta alcanzar un aspecto realista. Este enfoque no es práctico y no se suele seguir. Mucho más práctico es buscar imágenes de una puerta real y proyectarlas en la zona de la pared correspondiente. Este último método es el de proyección de texturas, al que dedicamos el presente tema.

En la imagen de la figura 1 se han aplicado diferentes texturas sobre las superficies de las geometrías. Así, en las paredes de la casa aparecen texturas de piedra que se consiguen proyectando imágenes de materiales reales. La puerta simula un material de madera que se puede conseguir con una función que simula las betas de la madera.

Primero presentamos una serie de conceptos básicos de creación de materiales con mapas y texturas. Después presentamos los distintos tipos de proyección de mapas y sus fundamentos matemáticos. Finalizamos el tema describiendo algunos mapas especiales como son los de abombamiento, desplazamiento, mapas de opacidad y mapas de entorno.



Figura 1: Ejemplo de escena en la que se han aplicado texturas sobre las geometrías para conseguir un acabado más realista (imagen propiedad de S. Moheimani vía Autodesk).

I. CONCEPTOS BÁSICOS

En el tema de sombreado de superficies, cada superficie tiene asociados una serie de atributos que condiciona su aspecto. Entre dichos atributos están el coeficiente de reflexión difusa, el color difuso, los coeficientes de reflexión, la transparencia etc... Con lo visto hasta ahora, todos los puntos de una misma superficie tienen el mismo valor para cada atributo.

El uso de texturas implica la modulación o el cambio de algún atributo de la superficie de manera que diferentes puntos de una misma superficie pueden tener diferentes valores para los mismos atributos. Al proceso que permite controlar el cambio de atributos en la superficie se le llama **texture mapping** o aplicación de la textura. A la función que permite cambiar los valores, o imagen que se proyecta se le llama: **texture map** o **mapa** o **textura**.

Las técnicas de aplicación de texturas pueden ser clasificadas de acuerdo a diferentes criterios, por ejemplo:

1. Según la **dimensionalidad** de las funciones de proyección distinguimos entre mapas 2D y 3D.
2. Teniendo en cuenta la **correspondencia** entre los puntos de las superficies y los puntos de la imagen o de la función podemos distinguir diversos tipos de proyecciones.
3. Si la función que define la textura es una tabla (una imagen digital grabada) o es un procedimiento o programa hablamos de **mapa bitmap** o **mapa procedural**.

Normalmente el atributo que se modula con el mapa es el color, pero puede ser también que cambien otros atributos como son los coeficientes de reflexión, transparencias etc...

En este tema, en la sección II desarrollamos los diferentes tipos de mapas teniendo en cuenta su dimensionalidad, distinguiendo entre mapas procedurales y mapas tipo bitmap; en la sección III distinguimos entre los distintos tipos proyección; en la sección IV detallamos mapas que se especializan en otros atributos que no son el color de la superficie.

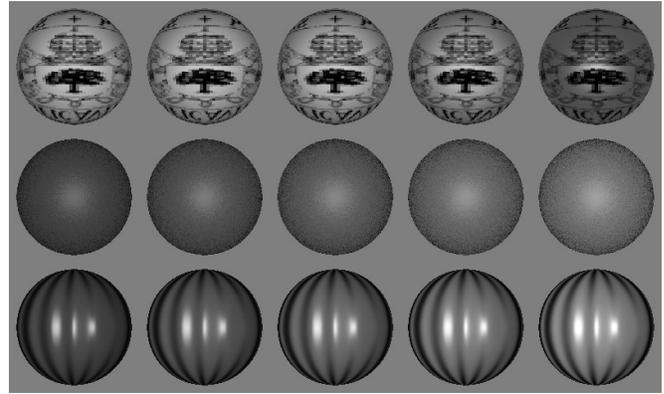


Figura 2: Representación de una esfera iluminada con distintas texturas. De izquierda a derecha variamos distintos coeficientes de iluminación

II. TIPOS DE MAPAS

II-A. Mapas tipo bitmap frente a mapas procedurales

En un **mapa tipo bitmap**, una imagen digital pregrabada (o pixmap o bitmap) se proyecta sobre los objetos de manera que los pixels de la imagen son empleado para cambiar los atributos de las superficies. Esta imagen digital puede ser una fotografía o una imagen sintética generada con algún programa informático. La idea intuitiva que ilustra este proceso es la de pensar que la imagen es una pegatina que se adhiere a la superficie. En la figura 2 se utiliza una imagen con el logo de la Universidad de Valladolid para cambiar el color de la superficie de las esferas.

En un **mapa procedural**, es un algoritmo o método automático el que aplica alguna fórmula conocida para cambiar sistemáticamente los valores algún atributo sobre las superficies de los objetos. En la fila central de la figura 2, las esferas tienen aspecto rugoso porque se ha usado un procedimiento que asigna valores aleatorios al color de los distintos puntos de la superficie. En la misma figura, en la fila inferior, se ha aplicado un procedimiento que cambia el vector normal en la superficie aplicando una función seno. Los procedimientos encargados de generar las texturas admiten parámetros de manera que para un mismo tipo de textura base, las opciones posibles pueden ser muchas. En el mismo ejemplo, podríamos tener un parámetro para establecer la amplitud del abombamiento y otro para la frecuencia a la que varían dichos abombamientos.

II-B. Mapas 2D frente a mapas 3D

En un **mapa 2D** la textura se codifica en dos dimensiones para luego ser proyectada sobre los objetos. El mejor ejemplo de este tipo de mapas son las texturas bitmap presentadas en la sección anterior. Una imagen bitmap no es más que una matriz de pixels de colores ordenada en un espacio bidimensional donde una de las dimensiones es la columna que ocupa el pixel (coordenada u) y la otra dimensión es la fila (coordenada v). Los bitmaps son útiles para simular diversos tipos de materiales empleando fotografías de dichos materiales. Las representaciones finales pueden ser muy realistas para simular

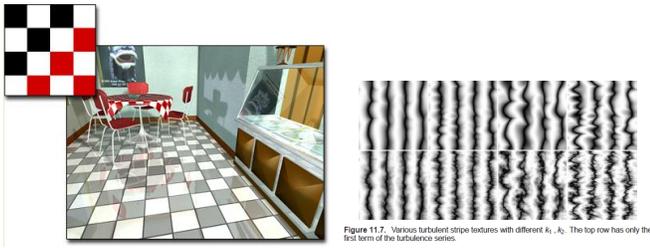


Figure 11.7. Various turbulent stripe textures with different k_x, k_y . The top row has only the first term of the turbulence series.

Figura 3: Ejemplos de uso del mapa tipo *checker* y una variación basada en perturbaciones (imágenes propiedad de Autodesk 3DS).

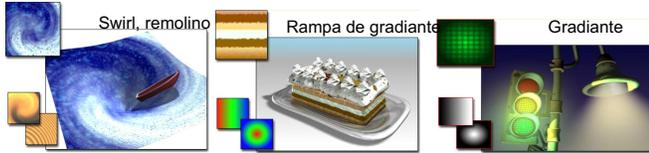


Figura 4: Ejemplos de mapas procedurales 2D (imágenes propiedad de Autodesk 3DS).

diversidad de materiales que van desde madera, paredes, piel, cuero... (ver figura 1).

Los mapas 2D también pueden ser procedurales, siendo un algoritmo el encargado de establecer valores para las dos coordenadas u y v de la textura. Un ejemplo de este tipo de texturas es el *checker* o tablero de ajedrez. Un tablero básico tendría como parámetros el tamaño de las filas y columnas y los dos colores a representar. Como se muestra en el algoritmo 1, aplicando una fórmula que juega con la periodicidad de las funciones trigonométricas, se puede conseguir los diferentes valores de la textura para cada coordenada u y v . La figura 3 muestra diversos usos de esta textura. El mismo procedimiento sirve para hacer patrones de tiras verticales y horizontales. Pueden aplicarse variaciones pseudoaleatorias a los patrones aplicando perturbaciones como también se muestra en la figura.

Algorithm 1 Algoritmo para generar una textura tipo tablero de ajedrez.

```

procedure CHECKER(colorA, colorB, scaleU, scaleV, u, v)
  patron  $\leftarrow$   $\sin(2\pi \cdot \text{scaleU} \cdot u) \cdot \sin(2\pi \cdot \text{scaleV} \cdot v)$ 
  return (patron > 0)?colorA : colorB
end procedure

```

Otros mapas procedurales 2D típicos son los gradientes, remolinos, o el uso de imágenes vectoriales. La figura 4 muestra ejemplos de este tipo de texturas.

Los **mapas 3D** son patrones generados mediante un procedimiento en tres dimensiones. Pueden trabajar directamente sobre los puntos 3D de los polígonos de los objetos por lo que es necesario hacer una proyección. Por ejemplo el mármol tiene una textura similar (no necesariamente igual) independientemente del corte que le demos. Cada vez que se hace un corte en el objeto se genera una nueva superficie cuyos puntos de coordenadas (x, y, z) se emplean como entradas del

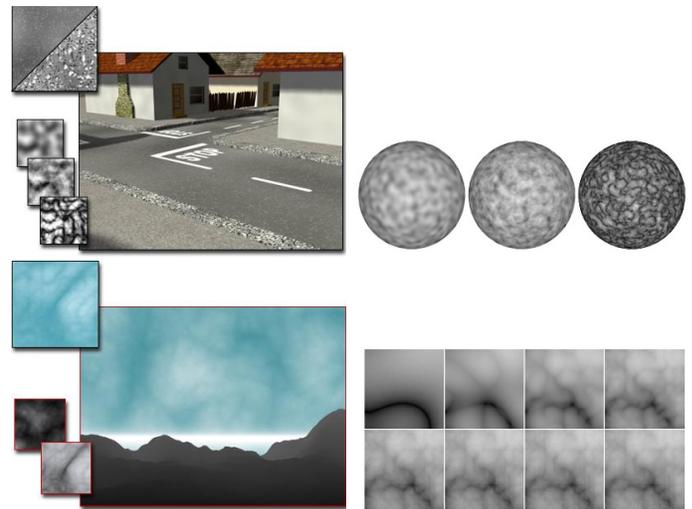


Figura 5: Diferentes tipos de mapas procedurales 3D. Abajo las texturas celular y smoke (imágenes propiedad de Autodesk 3DS).

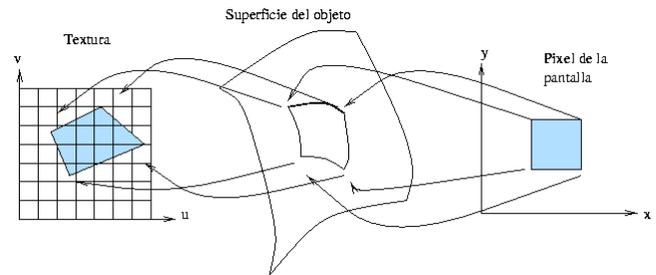


Figura 6: Proyección de mapas 2D sobre los objetos

procedimiento que les asigna un valor para los atributos de la textura, por ejemplo un color.

Algunos ejemplos de mapas 3D son el **ruido** que crea una perturbación aleatoria sobre la superficie basada en la interacción de dos colores o materiales. Existen ruidos especiales como son el **ruido Perlin** o la textura **celular**, que mejoran el aspecto granular del ruido blanco convencional. En el mapa celular la textura se organiza en granos que son muy útiles para simular mosaicos o suelos de piedras. El ruido Perlin se emplea para simular manchas similares a las manchas de humedad con variaciones como son las utilizadas para simular betas de mármol. Otro mapa 3D que también simula ruido es la textura de **humo**. El humo genera un patrón de turbulencia amorfo basado en fractales.

Otros mapas procedurales 3D pueden ser las manchas, el stucco, ondas, madera, hielo, etc... Cada uno de ellos tiene un procedimiento o programa asociado para generar los atributos de las superficies. La figura muestra ejemplos de este tipo de mapas 5.

III. PROYECCIÓN DE MAPAS 2D

A diferencia de los mapas 3D, los mapas 2D deben ser proyectados sobre los objetos 3D. La figura 6 ilustra el proceso. De un lado, sabemos que a cada pixel de la imagen

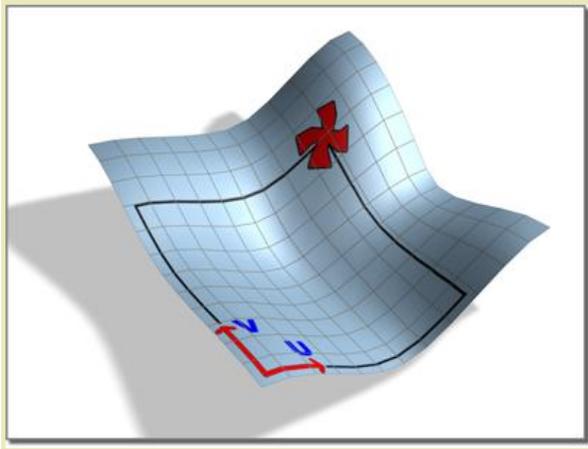


Figura 7: Coordenadas de proyección de textura (imagen propiedad de Autodesk 3DS)

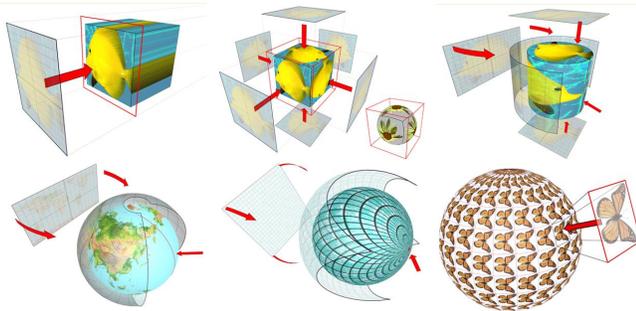


Figura 8: Tipos de proyección con gizmo. De izquierda a derecha y de arriba a abajo, proyección plana, de caja, cilíndrica, esférica, envolvente y de parches (imágenes propiedad de Autodesk 3DS).

en pantalla le corresponde una zona concreta de la escena 3D. La localización de esta zona se resuelve con la formulación desplegada en el tema de visualización 3D. De otro lado, y en el problema que toca ahora, debemos saber qué zona de la textura se corresponde con las superficies de las escenas. Las zonas de textura son las responsables del color en las superficies y por tanto, del color del pixel que se está analizando.

Por convenio, empleamos coordenadas llamadas u y v o **coordenadas del mapa** para referirnos a los puntos de la textura y x, y, z para referirnos a los puntos en la escena. Las coordenadas de la textura suelen estar normalizadas (valores entre 0 y 1) (ver figura 7). El problema de la proyección se resumen en encontrar un función $(u, v) = F(x, y, z)$ que asigna a cada punto (x, y, z) del modelo un punto (u, v) en la textura.

Existen diferentes **tipos de proyección** o mapeado que emplean una geometría exterior o **gizmo** sobre la cual colocan la textura 2D y desde la cual realizan la proyección. Este recurso facilita enormemente realizar la conversión entre coordenadas x, y, z de los objetos 3D y coordenadas u, v de las texturas. La figura 8 esquematiza las seis proyecciones más empleadas: plana, cilíndrica, de caja, esférica, por parches y envolvente.

En la **proyección cilíndrica** nos aprovechamos de la de-

Figura 9: Esquema de resolución de las proyecciones cilíndrica y esférica.

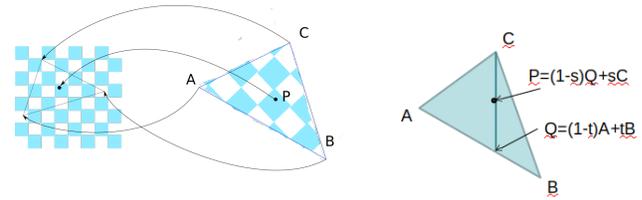


Figura 10: Cálculo de coordenadas internas a un triángulo empleando coordenadas baricéntricas.

finición paramétrica de los puntos de la superficie de un cilindro que utiliza la distancia a la base y ángulo rotación como parámetros. Si colocamos el centro de coordenadas en el centro de la base del cilindro, cada punto de la superficie del cilindro viene dado por la expresión:

$$(x, y, z) = (r \cos(\theta), r \sin(\theta), z) \quad (1)$$

donde $0 < \theta < 2\pi$, $0 < z < h$, r y h el radio y la altura del cilindro. Para asociar cada punto (u, v) del mapa con un punto del cilindro hacemos:

$$(u, v) = (\theta/2\pi, z/h) \quad u, v \in [0, 1] \quad (2)$$

El caso de la **proyección esférica** es más complicada, entre otras cosas porque hay indeterminaciones en los polos. La siguientes ecuaciones, tal y como aparecen en [1], resuelven el problema:

$$(x, y, z) = (r \cos \phi \sin \theta, r \sin \phi \sin \theta, r \cos \theta), \quad (3)$$

con $0 \leq \theta \leq \pi/2$ y $\pi/4 \leq \phi \leq \pi/2$. Podemos definir entonces (u, v) como:

$$(u, v) = \left(\frac{\phi}{\pi/2}, \frac{\pi/2 - \theta}{\pi/4} \right) \quad (4)$$

En la figura 9 se ilustra la resolución de estas ecuaciones.

No hay que perder de vista que los objetos 3D son un conjunto de poligonales, generalmente triángulos. Sea cual sea la proyección de puntos (x, y, z) en coordenadas de la textura (u, v) , con los métodos arriba descritos o con asignación directa, al final conseguimos tener una correspondencia que asigna a cada vértice \bar{A} , \bar{B} , \bar{C} de cada triángulo del objeto 3D un par de valores (u, v) que llamamos \bar{u}_A , \bar{u}_B y \bar{u}_C .

Para asignar un valor a cada **punto interior del triángulo**, podemos utilizar coordenadas baricéntricas de la forma $\bar{P} = \alpha \bar{A} + \beta \bar{B} + \gamma \bar{C}$. La correspondencia será $\bar{u}(\bar{P}) = \alpha \bar{u}_A + \beta \bar{u}_B + \gamma \bar{u}_C$. Como se muestra en la figura 10, $\bar{P} = (1-s)(1-t)\bar{A} + (1-s)t\bar{B} + s\bar{C}$ con s y t entre 0 y 1. Los puntos interiores cumplen que $\alpha + \beta + \gamma > 0$.

Aplicado tanto a mapas 3D como a mapas 2D está la técnica del **tiling** o mosaico. Esta técnica consiste en repetir sistemáticamente el mapa una o más direcciones cuando el tamaño de ésta sea menos que el tamaño del objeto. Para poder

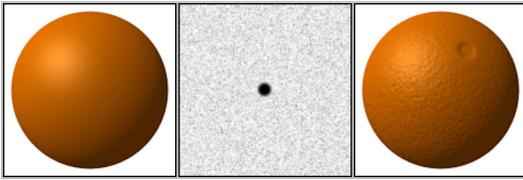


Figura 11: Mapas de abombamiento. A la izquierda una esfera sin mapa de abombamiento y a la derecha la misma esfera con el mapa. En el centro el mapa de abombamiento aplicado (imagen propiedad de Autodesk 3DS).

hacer tiling no sirve cualquier textura sino que debe haber cierta semejanza en los extremos de la imagen que van en contacto. Con esto se persigue evitar que se vea una frontera entre repeticiones de la imagen. Algunos programas de modelado permiten aplicar simetrías automáticas de la imagen que hacen coincidir los bordes de forma precisa: en realidad se secuencia la imagen con su imagen especular.

IV. MAPAS ESPECIALES

IV-A. Mapas de abombamiento y desplazamiento

Mediante los **mapas de abombamiento** podemos simular superficies abombadas, con diferencias de relieve. Para ello, perturbamos el valor de la normal de la superficie antes de utilizarlas en el modelo de iluminación.

La perturbación de la normal en la superficie se hace aplicando un mapa de abombamiento. El mapa de abombamiento es una matriz 2D que contiene los valores con los que se modifica el vector normal. En la fórmula del modelo de iluminación de Phong.

$$I_{\lambda} = I_{a\lambda}k_aO_{d\lambda} + f_{att}I_{p\lambda} [k_dO_{d\lambda}(\bar{N} \cdot \bar{L}) + k_s(\bar{V} \cdot \bar{R})^n]. \quad (5)$$

la desviación del vector normal \bar{N} provoca cambios en el valor de I que hacen que el observador vea la superficie como si se hubiera producido un hundimiento o un abultamiento.

En la figura 2, parte inferior, hemos modulado el valor del vector normal a la superficie en función del ángulo de rotación en el recorrido de la esfera. De este modo, al aplicar el modelo de iluminación de reflexión especular, conseguimos que las esferas aparezcan como abombadas. En la figura 11 mostramos el efecto de aplicar una imagen en escala de grises sobre una esfera. Los valores más oscuros del mapa aparecen hundidos en el objeto 3D. Como resultado, se consigue dar sensación de superficie detallada.

Un **mapa de desplazamiento** cambia la geometría de las superficies de los objetos utilizando la textura. El valor del mapa se emplea para cambiar las posiciones (x, y, z) de los puntos, realizando alteraciones en la posición que sí tienen efectos reales en la silueta de los objetos.

Los mapas de abombamiento ni arrojan sombras ni afectan a la silueta. Por contra en un mapa de desplazamiento, al cambiar las posiciones de los puntos de las superficies, consiguen efectos inalcanzables para los mapas de abombamiento. Ahora las sombras cambian porque hay desplazamientos de

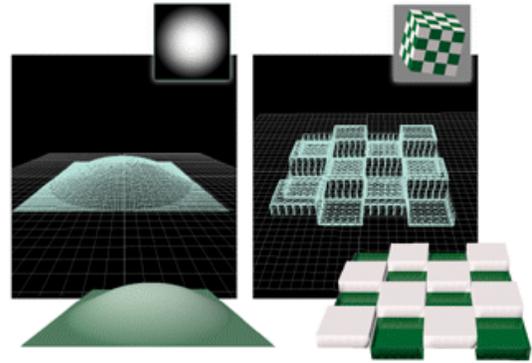


Figura 12: Mapas de desplazamiento. Arriba los mapas aplicados (un gradiente y un checker) abajo el resultado (imagen propiedad de Autodesk 3DS).

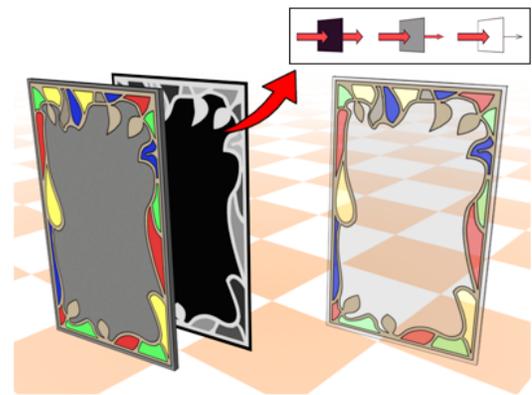


Figura 13: Mapas de opacidad. En blanco y negro los mapas de opacidad aplicados, a la derecha el resultado conseguido (imagen propiedad de Autodesk 3DS).

los puntos en la superficie, bien hacia arriba bien hacia abajo. La figura 12 muestra un ejemplo de uso.

IV-B. Mapas de opacidad

Los mapas de opacidad se utilizan para determinar si la superficie es opaca o transparente. El convenio suele ser utilizar el blanco para zonas opacas y el negro para zonas transparentes. En las zonas opacas se completa con otro mapa que da los detalles de la superficie. En la figura 13 se muestra el proceso de uso de los mapas de opacidad. En ambos casos, el modelo sobre el que se aplica la textura es simplemente un plano. Al aplicar el mapa de opacidad se delimitan las zonas del plano que serán visibles. La fotografía del caminante y del árbol se utilizan para dar colores al plano en las zonas opacas.

Este tipo de mapas pueden utilizar la propiedad de transparencia de la superficie, de manera que el valor del mapa indica el grado de transparencia. En este caso, pueden emplearse tonos de gris para establecer valores intermedios de opacidad. En otros casos, simplemente se trata de un valor binario que indica al motor de render si debe o no renderizar cada punto de la superficie del objeto.

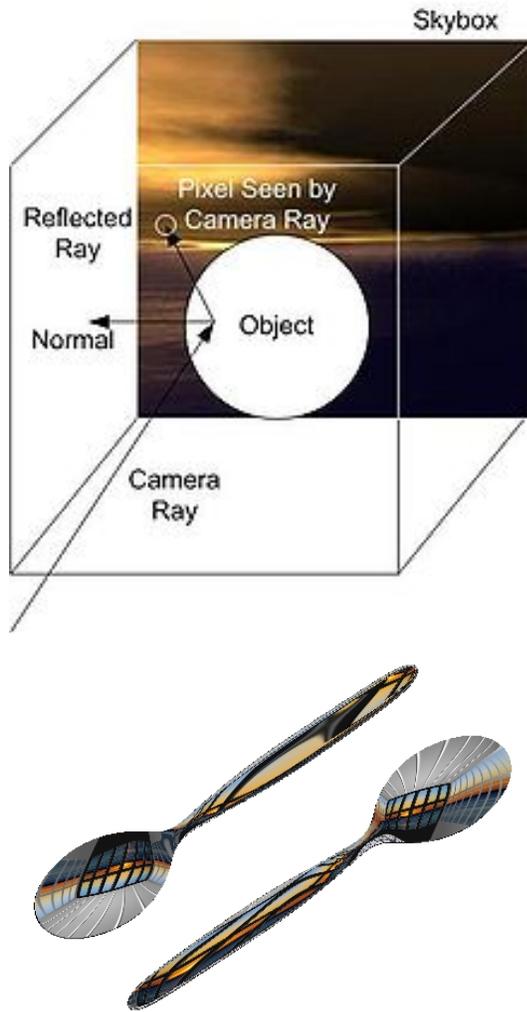


Figura 14: Mapas de entorno. Se ilustra el proceso de cálculo del punto del mapa que afecta al punto del objeto (imagen propiedad de T. Topher CC BY-SA vía Wikipedia).

IV-C. Mapas de entorno

Los mapas de entorno se emplean para poner fondos en las escenas y hacer que estos fondos sean reflejados por los objetos. Los objetos, al reflejar el contexto en el que están integrados, incrementan la sensación de realismo. El mapa, por ejemplo una foto de un paisaje (ver figura 14), se coloca sobre una envoltura que suele ser una esfera, un cilindro o las caras de un cubo. En lugar de hacer una proyección del mapa sobre los objetos, como hemos hecho en el caso de las texturas convencionales, aquí se utiliza para calcular los reflejos de los objetos que componen la escena. Si en el caso de los mapas de abombamiento era el vector \vec{N} de la ecuación 5 el que era modificado, aquí es el vector \vec{V} el que toma protagonismo. Los vectores \vec{V} inciden con algún de la superficie envoltura para establecer los colores reflejados.

La calidad de los mapas de entorno depende fuertemente de la resolución de la imagen utilizada. Hay que tener en cuenta que la superficie sobre la que se proyectan estos mapas envuelve toda la escena, por lo tanto, es de un tamaño mucho



Figura 15: Mapas de entorno elaborado para realizar la proyección con precisión (imagen propiedad de S. Majboroda CC0 vía polyhaven.com).

mayor. No usar una imagen de resolución elevada tendrá la consecuencia de generar unos reflejos pobres.

La textura empleada como mapa de entorno puede ser usada como imagen de fondo o no. En caso de ser utilizada como imagen de fondo, puede ser interesante utilizar una variación de la misma imagen porque el fondo de la imagen y el mapa de entorno no tienen por qué coincidir. Los reflejos son dependientes del punto de vista del observador, pero la envoltura es la misma independientemente de la posición de la cámara, por lo que el efecto en las animaciones puede ser inesperado sobre todo cuando la imagen de fondo no se desplaza de forma consistente.

Al analizar rayos reflejados los mapas de entornos también se conocen como mapas de reflexión. Los mapas de reflexión se integran muy bien en el modelo de trazado de rayos en el que los rayos reflejados son seguidos hasta que terminan en una fuente de iluminación o en algún punto del mapa de entorno (ver figura 15).

La realización de un mapa de entorno que tenga en cuenta la proyección es un proceso complejo que incluye la realización de imágenes panorámicas. En estas imágenes panorámicas no sólo es complicado fusionar diferentes imágenes de manera que haya continuidad espacial, sino que también es complejo ajustar las condiciones de iluminación de las distintas fotografías tomadas.

MÁS INFORMACIÓN

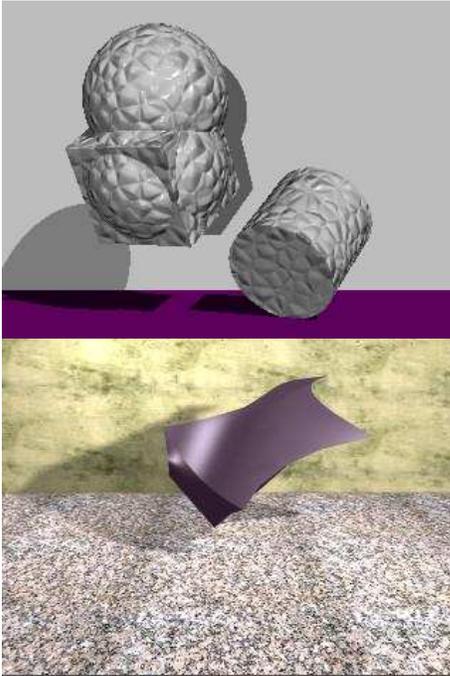
En <https://sketchfab.com/blogs/community/how-to-create-your-own-hdr-environment-maps/> se muestra cómo construir el mapa de entorno.

Los libros básicos de referencia sobre informática gráfica encontramos capítulos que dan cobertura a los puntos tratados en este tema: [2] dedica el tema 26 a tratar cuestiones relativas a la luz y el capítulo 27 a cuestiones relativas a sombreado. En [3] se reserva el capítulo 14 a tratar temas relativos a modelos de iluminación y de presentación de superficie. Watt [4], [1] dedica los primeros capítulos a estos temas.

CUESTIONES Y PROBLEMAS

1. En la aplicación de texturas, ¿a qué hacen referencia los ejes de coordenadas (u,v)?

2. Detalle el principio básico para la formación de texturas de tipo abombamiento. Indique la relación de esta técnica con los modelos de iluminación.
3. Existen diferentes técnicas para mejorar el realismo en escenas 3D. Enumere las que se observan en la figura (parte de la izquierda). De todas las técnicas para hacer texturas que conoce, en la figura se aplica sólo una de ellas. Diga cual es y explique sus fundamentos.



4. ¿Cuál de los parámetros del modelo de iluminación de Phong (ver la ecuación) se modifica para conseguir el efecto de texturas de tipo abombamiento?
5. La figura (parte de la derecha) ha sido generada en un software tipo 3D Studio empleando tres objeto tipo caja. Comente las transformaciones geométricas que ha sido necesario aplicar sobre ellos para llegar a la composición final. (no es necesario comentar nada sobre efectos de realismo).
6. Nombre dos criterios de clasificación de las técnicas de *texture mapping*.
7. ¿Qué es un mapa procedural?
8. Explique brevemente el papel de los GIZMOS que emplea 3D Studio en el modificador UVW Map en aplicación de texturas.

REFERENCIAS

- [1] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*. acm Press, Addison-Wesley, 1992.
- [2] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [3] D. Hearn, M. P. Baker *et al.*, *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall,, 2004.
- [4] A. Watt, *3D Computer Graphics*, tercera ed. Addison-Wesley, 2000.

Iluminación global y rendering

ÍNDICE

I.	Definiciones básicas	2
II.	Rendering con iluminación local	2
III.	Técnicas fundamentales de iluminación global	3
III-A.	Trazado de rayos	3
III-B.	Radiosity	5
III-C.	Photon mapping	5
IV.	Técnicas auxiliares	6
IV-A.	Final gathering	6
IV-B.	Oclusión ambiental	6
V.	Antialiasing	7
	Referencias	7

Los elementos fundamentales de una escena 3D son los objetos geométricos que la componen, las luces que iluminan el conjunto y una cámara desde la cual se visualiza el conjunto. Para, con estos elementos, generar una imagen sintética se debe aplicar un procedimiento de rendering que muestre cómo se ven los objetos geométricos desde la cámara teniendo en cuenta la iluminación establecida.

En este tema, primero presentamos la técnica básica de rendering empleando iluminación local, para luego mostrar los algoritmos de iluminación global más empleados, que son raytracing, radiosity y photon mapping. A continuación explicamos algunas técnicas de apoyo en el proceso de generación de iluminación global como son final gathering, oclusión ambiental y caustics. Finalizamos presentando el problema del aliasing como un aspecto fundamental relacionado con la iluminación global y el rendering.

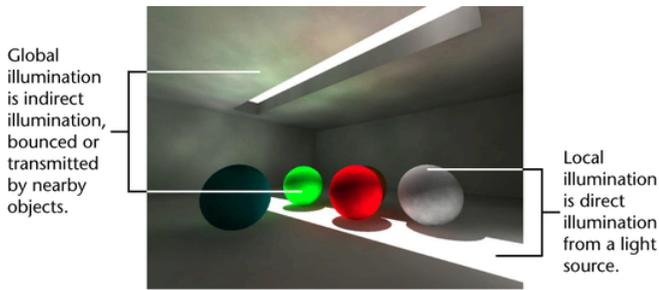


Figura 1: Iluminación global frente a iluminación local (imagen propiedad de Autodesk Maya Creative).

I. DEFINICIONES BÁSICAS

El **rendering** es el proceso de generación de imágenes digitales sintéticas a partir de escenas 3D que incluyen los objetos geométricos que la componen, las fuentes de iluminación y la cámara desde la cual se visualiza. En este tema se explican cuestiones relativas a dicho proceso poniendo especial atención en cuestiones relativas a la iluminación global, esto es, a la iluminación de los puntos de la escena en los que no incide directamente la luz proveniente de ninguna fuente de luz sino del reflejo de la luz en los objetos de la escena.

Los algoritmos de rendering que emplean **iluminación global**, simulan los reflejos de la luz entre las distintas superficies. El término contrasta con iluminación local. En **iluminación local** las superficies de la escena se renderizan empleando sólo las fuentes de iluminación que proyectan luz directamente sobre ellas, sin reflejos.

Se considera que la luz está compuesta por múltiples partículas discretas llamadas fotones. Los fotones hacen recorridos que empiezan en las fuentes de luz hasta que se encuentran con las superficies de la escena. Algunos fotones son absorbidos y otros rebotan trazando un nuevo recorrido. El hecho de que algunos fotones de determinadas longitudes de onda reboten y que otros sean absorbidos es lo que determina el color de la superficie.

Las superficies lisas (o pulidas) reflejan los fotones en una única dirección, en un ángulo igual al ángulo de incidencia. Son las superficies especulares. La mayoría de las superficies tienen componente especular y difusa. La manera en la que los fotones rebotan dependen de lo pulida que esté la superficie. Las superficies rugosas también reflejan fotones aunque en direcciones dispares.

La iluminación final de una escena está determinada por la interacción de billones de fotones con las superficies. En un punto dado de una superficie, es posible que lleguen fotones directamente de una fuente de luz (**iluminación directa**) o si no, indirectamente de rebotes de otras superficies (**iluminación indirecta**) A la retina del observador, llegan una porción del total de fotones. Su estímulo, hace que se forme la imagen que es percibida por el cerebro. En informática gráfica sustituimos la retina por un plano de visión. El objetivo de un algoritmo de iluminación global es recrear este proceso de la forma más realista y rápida posible.

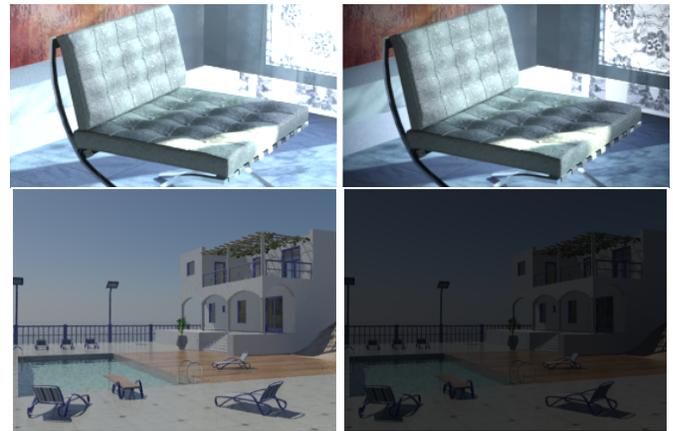


Figura 2: Efectos de la exposición a la iluminación global (imágenes propiedad de Autodesk 3DS).

II. RENDERING CON ILUMINACIÓN LOCAL

Hacer el seguimiento de todos los posibles rebotes de todos los posibles rayos de luz que recorren la escena no es factible empleando los ordenadores que tenemos hoy en día. Debemos asumir que es necesario elegir un subconjunto de la totalidad de rayos posibles. En el extremo más sencillo consideramos sólo el rayo que sale de cada pixel del plano de visión. Esta técnica, conocida **ray casting**, calcula el punto de intersección del rayo con la superficie más cercana de la escena 3D. Una vez identificado el punto visible desde el pixel, se calcula el color en dicho punto aplicando un modelo de iluminación para la superficie visible que tenga en cuenta las fuentes de luz que inciden directamente sobre dicho punto.

Cuando el cálculo del color en cada pixel se hace de manera sistemática recorriendo la imagen de izquierda a derecha y de arriba a abajo se dice que hacemos **rasterization** (rastreo) de la imagen. El algoritmo 1 detalla el procedimiento.

Algorithm 1 Algoritmo básico de rastering empleando iluminación local.

```

for cada pixel (x, y) do
  R ← rayo que sale de la cámara y pasa por (x, y)
  for cada triángulo T do
    P ← intersección entre el rayo R y T
    if P es el punto más cercano then
      for cada fuente de iluminación L do
        color en P += iluminación en P de L
      end for
    end if
  end for
end for

```

Nótese la enorme complejidad (gran cantidad de cálculos) del algoritmo cuando el número de triángulos de la escena es elevado. Tal y como está definido, en cada paso es necesario comprobar la posición de cada triángulo de la escena, lo cual puede suponer un número de cálculos tremendo cuando el número de objetos es elevado. En implementaciones prácticas se recurre a simplificaciones como puede ser reducir el número

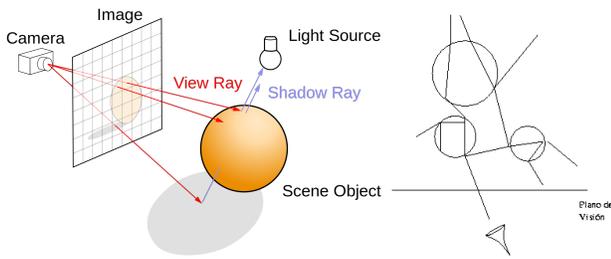


Figura 3: Esquema de funcionamiento del algoritmo de trazado de rayos (la imagen de la izquierda es propiedad de C. Henrik CC BY-SA vía Wikipedia).

de triángulos teniendo en cuenta sólo los que están dentro del volumen de visión. Otra forma de simplificar el número de cálculos consiste en aprovechar la consistencia de las escenas 3D en el sentido de que si en un punto (x, y) se visualiza el triángulo T , la probabilidad de que en el siguiente punto $(x, y + 1)$ se visualice el mismo triángulo es elevada y la probabilidad de que el color sea similar también es elevada. Un motor de render que utiliza estas propiedades es el conocido como **scanline rendering**. En este método, cada línea horizontal de la pantalla se trata individualmente, para lo cual se identifican primero todos los triángulos que pueden afectar a dicha línea. A la hora de procesar la siguiente línea se tiene en cuenta que muchos de los triángulos coinciden, de manera que se reutilizan muchos cálculos.

III. TÉCNICAS FUNDAMENTALES DE ILUMINACIÓN GLOBAL

En este apartado presentamos las tres técnicas fundamentales en las que se apoya la iluminación global. Cada una de ellas hace referencia a cada uno de los tres agentes principales del render: los rayos de visión, los rayos de iluminación y las superficies de los objetos. Primero presentamos el algoritmo de trazado de rayos que permite hacer el seguimiento de los rayos de luz desde el plano de visión; después el algoritmo de radiosity, que considera el papel de los elementos de las distintas superficies en el render; y finalmente el método de foton mapping que traza el recorrido de los fotones emitidos desde las fuentes de iluminación.

III-A. Trazado de rayos

El algoritmo de trazado de rayos asume que, aunque hay billones de fotones viajando por la escena, los que primero importan son los que llegan al ojo del observador. Comparte en este sentido el principio del algoritmo de *ray casting* explicado en la sección anterior. La diferencia fundamental es que ahora se considera que el rayo proyectado, una vez que incide con el primer punto de la superficie, genera nuevos rayos (los reflejados y refractados) que se pueden seguir para calcular nuevos efectos del contexto en dicho punto.

El algoritmo funciona trazando rayos, no desde las fuentes de iluminación, sino desde atrás; de manera que desde cada pixel de la imagen (del plano de visión) se lanzan rayos a los modelos 3D. Así, sólo se hacen cálculos para la información estrictamente necesaria para construir la imagen.



Figura 4: Ejemplo de escenas generadas con un algoritmo de trazado de rayos (imágenes creadas por G. Tran y por T. Nakai vía Wikipedia y Autodesk respectivamente).

Para cada pixel de la imagen se emplea el siguiente procedimiento:

- Se lanza un rayo desde el ojo del observador (centro de proyección) a través de cada pixel en el plano de proyección hasta que intersecta con una superficie. Sabemos cómo refleja en la superficie porque conocemos el material, pero no sabemos la cantidad de luz que llega a este punto.
- Para saber la cantidad de iluminación, trazamos un rayo desde el punto de intersección hacia cada fuente de luz (shadow ray). Si el rayo hacia la fuente de luz no es bloqueado por ningún otro objeto, hay que computar la contribución de esta luz al color de la superficie.
- Si la superficie intersectada es brillante o transparente, tenemos que determinar lo que se ve desde la superficie. Los pasos anteriores se repiten en el rayo reflejado (y en caso de transparencia en el rayo transmitido) hasta que llega a otra superficie. El color en la intersección subsecuente se calcula y se tiene en cuenta a la hora de calcular el color de punto original.
- Si la segunda superficie es también reflectiva o transparente, el proceso se repite hasta un máximo de interacciones o hasta que ya no hay intersecciones.

La figura 3 ilustra el proceso. Los rayos emanan desde la

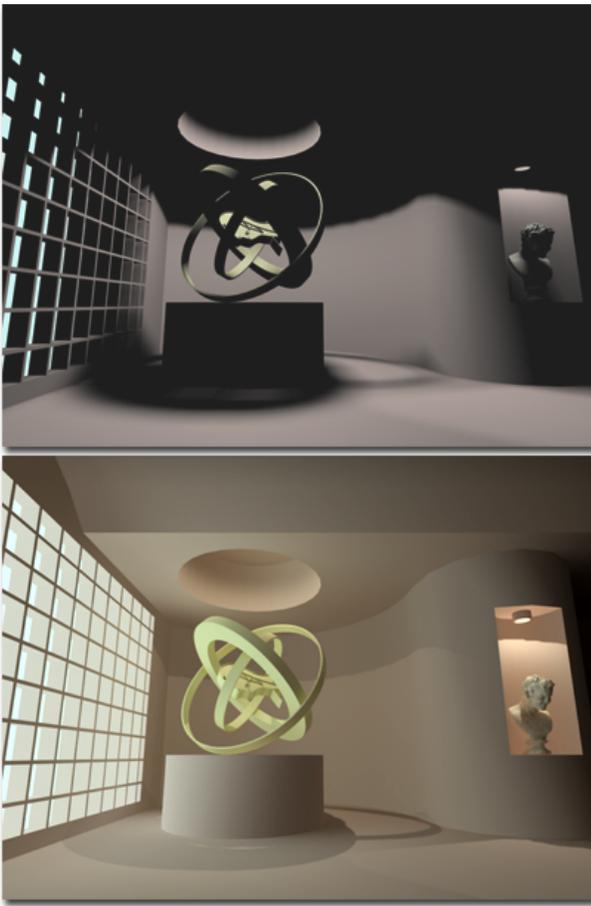


Figura 5: Ejemplo de escena generada con y sin radiosidad (arriba) (imágenes propiedad de Autodesk 3DS).

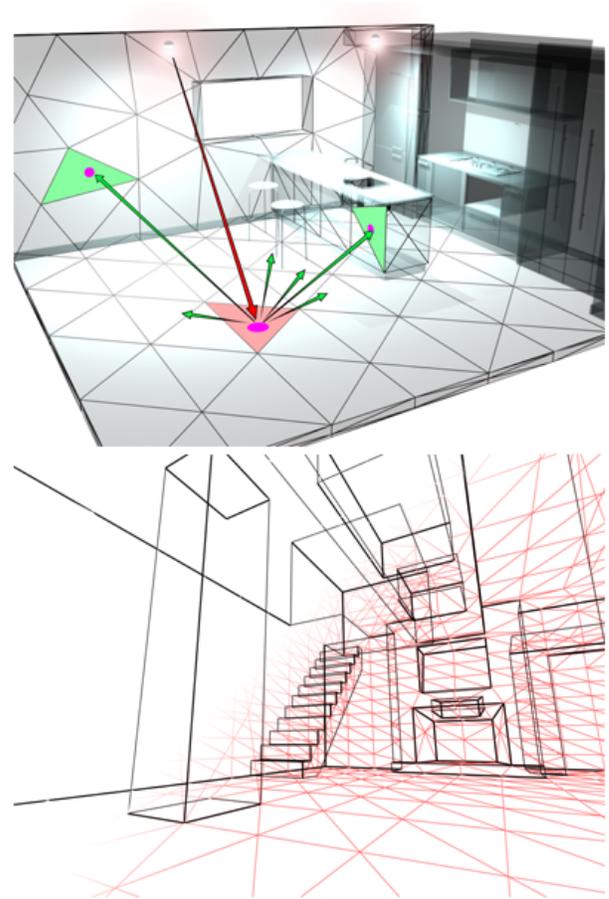


Figura 6: Ejemplo de red de radiosidad (imágenes propiedad de Autodesk 3DS).

cámara y se traza el recorrido que realizan por la escena. En las múltiples intersecciones se calcula la contribución de los múltiples reflejos a la hora de calcular el color.

El método de trazado de rayos se ha impuesto a la hora de generar escenas fotorrealistas (ver ejemplo en figura 4) porque presenta importantes ventajas a la hora de hacer los renders. Entre las ventajas de este método citamos las siguientes:

- Incluye transparencias y reflejos de forma natural. Al calcular en cada punto la contribución del rayo reflejado y reflectado, estamos computando de forma indirecta la contribución de la transparencia y de la refracción. Se calcula la transparencia porque habrá contribución del rayo refractado sólo cuando el material incluya cierto grado de transparencia. Se calcula el reflejo porque el seguimiento del rayo reflejado se hace teniendo en cuenta la función BRDF del material, de manera que la contribución del rayo reflejado será mayor en función del glossiness de la superficie.
- Incluye eliminación de superficies ocultas. Las superficies ocultas son aquellas en las que no incide ningún rayo proyectado de forma directa desde el observador o de manera indirecta cuando las superficies que están delante son transparentes.
- Sombreado por iluminación directa. En cada punto de

incidencia de un rayo de luz se calcula la ecuación de iluminación teniendo en cuenta las fuentes de luz visibles desde dicho punto.

- Sombreado por iluminación ambiental. En cada punto visible se calcula la iluminación teniendo en cuenta múltiples rebotes del rayo de luz en otros puntos de la escena, por lo tanto se tiene en cuenta iluminación indirecta. Los puntos de intersección de los rayos reflejados y refractados funcionan de alguna manera como fuentes de iluminación que contribuyen con energía luminosa transmitida a calcular el color.
- Cálculo de sombras. Un punto estará en sombra cuando no hay ninguna fuente de iluminación visible desde él, o lo que es lo mismo, cuando el rayo reflejado no llega a ninguna fuente de luz. La penumbra se genera de forma natural al ser la zona de la sombra donde incide una menor cantidad de rayos de la fuente de luz.

La desventaja es su alto coste computacional. Si un algoritmo de *ray casting* básico sin optimizar debe realizar $n \times m \times T \times L$ cálculos para renderizar una imagen de $n \times m$ pixels de una escena con T triángulos y L fuentes de iluminación; el algoritmo de trazado de rayos que considere que cada triángulo puede tener reflexión y refracción deberá hacer un número de cálculos del orden $n \times m \times T \times L \times 2^k$ cálculos, siendo k

el número de rebotes. Esto si en cada punto de contacto se traza solo un rayo reflejado y otro transmitido. Si se trazan más rayos, el número sería mucho mayor.

III-B. Radiosity

La técnica de Radiosity, difiere fundamentalmente de la técnica de ray-tracing. En lugar de seguir los rayos a partir de los pixels de la imagen, en el método de radiosity se comienza con la luz emitida desde las fuentes de iluminación y se analizan las superficies donde ésta incide. Lo importante en este método, de hecho, son las superficies de la escena y no el punto de observación. En lugar de determinar el color de cada pixel en la pantalla, radiosity calcula la intensidad para cada superficie en el entorno analizando la interacción entre dichas superficies.

Para ello, se dividen las superficies en una malla de superficies más pequeñas llamadas elementos o parches. El algoritmo de radiosidad calcula la cantidad de luz distribuida desde cada elemento de la malla al resto de los elementos de la malla. Las fuentes de iluminación son vistas también como superficies que emiten luz. Los valores finales de radiosidad o iluminación transmitida entre parches se almacenan para cada elemento de la malla.

La luz indirecta se transmite entre las superficies a través de la reflexión difusa de cada superficie y es almacenada en los vértices de las mallas. El cálculo de la **malla de radiosidad** es un proceso iterativo que se realiza subdividiendo progresivamente las superficies. El algoritmo se detiene cuando la radiosidad emitida desde cada punto del parche es la misma. En este momento el parche es considerado un *elemento*.

La información de sombreado de cada parche se almacena en sus vértices. Se añaden más vértices cuando hacen falta más detalles: por ejemplo, las fronteras de las sombras. Para renderizar se emplea esta información y se interpola a lo largo de los puntos del parche.

Como inconvenientes de este método, primero hay que tener en cuenta que se trata de un algoritmo muy costoso. Tanto la división iterativa en parches como el cálculo de cada valor de radiosidad entre parches, suponen procesos muy costosos en tiempo de ejecución. También hay que tener en cuenta que el método sólo se emplea con superficies difusas, no admite reflexiones especulares ni transparencias. Por último, el método está pensado para interiores, espacios cerrados donde desde cada elemento se vean otros elementos en todas las direcciones.

La principal ventaja es que, al ser un método que se apoya más en el principio físico de distribución de la luz entre superficies, se obtienen escenas de un gran realismo. En animaciones, además, se puede reutilizar la red de radiosidad siempre que no haya cambios en la escena aparte del movimiento de la cámara. Lo cambios en la cámara no afectan en la red de radiosidad por lo que se pueden realizar animaciones tipo recorridos virtuales de forma muy rápida. La figura 6 muestra un ejemplo de red de radiosidad y una imagen generada con este método.



Figura 7: Ejemplo de escena que emplea photon mapping.

III-C. Photon mapping

En el método de photon mapping, para calcular la iluminación indirecta, se hace un seguimiento del camino que siguen los fotones al salir de las fuentes de la iluminación. A diferencia del método de trazado de rayos, aquí se trazan los caminos de los fotones emitidos desde cada fuente de luz y no el camino de la luz desde el punto de vista del observador. Los fotones que emanan desde las fuentes de iluminación se trazan a través de la escena, siendo reflejados o transmitidos por los objetos.

En un ideal, si encontramos todos los fotones que llegan a un punto P de la escena, podríamos utilizarlos para calcular la iluminación en dicho punto. Sin embargo, este enfoque no es viable porque no es factible hacer el seguimiento de todos los fotones que pueden salir de una fuente de luz, se debe seleccionar un subconjunto. Acertar con los fotones que inciden en el punto P sería pura coincidencia. Lo que se hace es hacer el seguimiento de los fotones y cuando golpean una superficie difusa, se registra dicho impacto en un mapa de fotones. El mapa de fotones es una textura a aplicar en las superficies. Esta textura registra los impactos de los fotones en dicha superficie, bien provenientes de alguna fuente de iluminación, bien provenientes de rebotes en otras superficies.

La generación de mapas de fotones es un proceso costoso. Para mejorar el rendimiento, se puede especificar explícitamente:

- Qué fuentes de luz emiten fotones para el cálculo de iluminación indirecta.
- Qué objetos pueden generar brillos cáusticos (descritos abajo) o participar en la iluminación global con rebotes y refracciones.
- Qué objetos pueden recibir brillos cáusticos o iluminación global.

Además, para cada fuente de luz podemos parametrizar el número de fotones que emiten e incluso su tamaño.

La emisión de fotones se hace desde las fuentes de luz en una dirección dada que es alterada mediante a factores aleatorios para simular la distribución de energía lumínica desde dicha fuente. A la fuente se le asigna una potencia lumínica que es proporcional a la cantidad de fotones emitidos por unidad de tiempo. Cuando los fotones inciden con una superficie puede ocurrir que sean absorbidos contribuyendo al color en el mapa de fotones y sean reflejados o refractados continuando distribuyendo luz en la escena. La proporción de luz absorbida y reflejada depende del material pudiendo incluir también factores aleatorios.



Figura 8: Ejemplo de escenas que incluyen brillos cáusticos (imágenes propiedad de H. Ottertedt CC BY-SA vía Wikipedia y de Arnold).

Este método se suele utilizar en conjunción con el método de trazado de rayos limitando el número de rebotes. Además, suele ser necesario aplicar filtros a los mapas de fotones para que desaparezca el aspecto granulado indeseable de los mismos. Los granos no son otra cosa que los impactos de los fotones en las superficies (ver figura 7).

Los **brillos cáusticos** son un recurso visual que consiste en un efecto luminoso de concentración y dispersión de la cantidad de energía luminosa que se proyecta sobre la escena y que tienen su origen en los objetos que reflejan y/o refractan (ver figura 8). No son especialmente costosas de computar pero deben incluirse fuentes de luz que emitan fotones.

IV. TÉCNICAS AUXILIARES

Los algoritmos de trazado de rayos, photon mapping y radiosity son los métodos fundamentales empleados para renderizar teniendo en cuenta la iluminación ambiental. No hay que entenderlos como técnicas exclusivas sino que un mismo motor de render puede utilizar técnicas que se inspiren en más de uno de estos métodos básico. Además, es muy común que se añadan otros recursos y técnicas de gran utilidad como son el *final gathering* y la oclusión ambiental entre otros.

IV-A. Final gathering

La técnica de final gathering se utiliza junto al photon mapping y trazado de rayos para suavizar resultados. Esta técnica recoge la idea básica del algoritmo de radiosity de tener en cuenta la iluminación de las superficies vecinas. Para cada punto de la superficie que es renderizada se mira alrededor, en todas las direcciones, para identificar las superficies que son visibles desde ese punto. Cuando se mira alrededor, se tiene en cuenta el color e iluminación de las superficies cercanas y los fotones vecinos en el photon map. El color de las superficies vecinas se utiliza para corregir el color de la superficie analizada.

El efecto más característico de la iluminación global es del **color bleeding**: el color de un objeto se proyecta en sus vecinos. En la figura 9 se ilustra el análisis de superficies vecinas a la hora de asignar el color a la superficie. La misma superficie tiene tonalidades diferentes en los puntos A y B por el hecho de estar cerca de objetos de diferente color. La parte inferior de la figura muestra el efecto del *color bleeding*, donde el color de las paredes está contaminado por el color de las paredes vecinas.

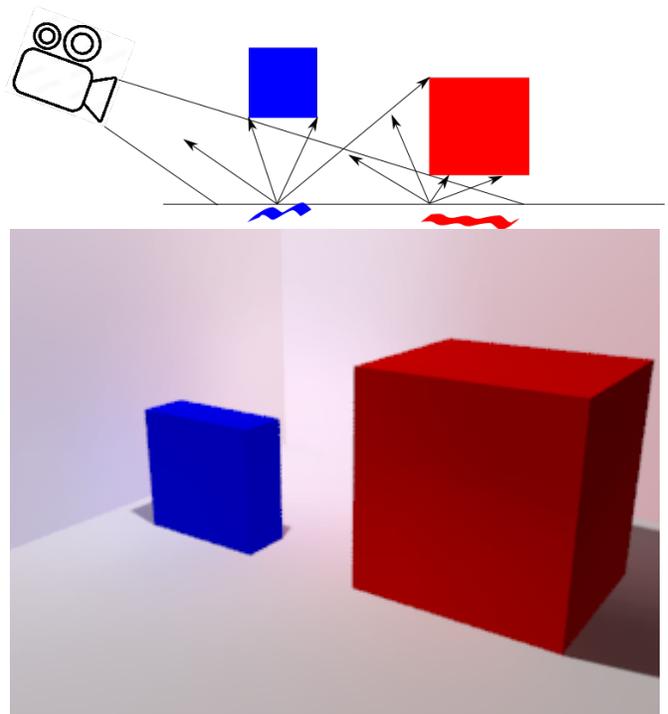


Figura 9: Ejemplo de escenas que emplean final gathering.

En cada punto de intersección del rayo de visión se trazan rayos en un hemisferio determinado por la normal a la superficie. Los rayos trazados inciden con objetos (o con el ambiente) que contribuirán con su color a la componente difusa del punto emisor. Es un cálculo de la radianza que el punto emisor recibe desde superficies vecinas. Un parámetro que debe ser establecido cuando se utiliza photon mapping es el número de rayos en torno al punto emisor que se analizan.

IV-B. Oclusión ambiental

Esta técnica se emplea para simular las sombras de contacto que son aquellas sombras que aparecen en las zonas de contacto de dos objetos. Cuando hay objetos en contacto, estos se ven menos expuestos a la iluminación ambiental, por lo que aparecen sombras en las zonas de proximidad. Para conseguir este efecto es necesario tener un render específico o bien añadir mapas o texturas a los objetos que tengan en cuenta este efecto.

Para conseguir el efecto de oclusión ambiental, se proyectan rayos en un hemisferio alrededor de la normal a la superficie. La cantidad de oclusión ambiental depende del número de rayos que contactan con otros objetos de la escena. El resultado es que se oscurecen las superficies en función de la proximidad de otras superficies. Al considerar un hemisferio alrededor de los puntos, también hay un efecto sobre las zonas cóncavas, que aparecerán más oscuras que las superficies convexas.

Como no depende de luces ni de materiales es una forma muy rápida de renderizar. Si el objeto no forma parte de una animación el mapa de oclusión se puede precalcular una única vez y reutilizar el valor. La figura 10 muestra un ejemplo de uso. El resultado es que las esquinas aparecen oscurecidas y que las concavidades y convexidades aparecen resaltadas.



Figura 10: Ejemplo de escenas que incluyen oclusión ambiental (imagen propiedad de Autodesk Arnold).

V. ANTIALIASING

Representar las imágenes del redering final en dispositivos de una resolución limitada trae consigo asociado el problema del **aliasing**. Las escenas 3D necesitarían resolución ilimitada para poder ser representadas ya que, al tratarse de modelos vectoriales, la calidad podría ser siempre superior a la que nos ofrecen los monitores. La consecuencia es que, aparece bordes y ruido indeseable en la imagen final. El antialiasing en un conjunto de técnicas empleadas en informática gráfica (en general en procesamiento de señal) para evitar que aparezcan estos artefactos en la escena final.

En rendering, las técnicas de antialiasing más utilizadas son el oversampling y el filtrado. El **oversampling** implica utilizar más datos de los que, en principio, se necesitan. En trazado de rayos por ejemplo, en lugar de proyectar un rayo por pixel, pueden lanzarse varios (cuatro u ocho por ejemplo) y usar la media del resultado. El **oversampling adaptativo** implica que se utilizan más rayos (o en general que se emplea más información) en las zonas en la que hace falta un nivel de detalle mayor. La figura 11 muestra el problema del aliasing de dientes de sierra y cómo se puede mitigar con la técnica de oversampling y oversampling adaptativo.

Por otro lado, el **filtrado** consiste en utilizar los valores de los pixels vecinos para calcular el valor de cada pixel. Se selecciona un área de pixels vecinos y se pondera empleando algún fitro para recalcular el valor de cada pixel.

Los motores de render suelen ofrecer parámetros para ajustar las opciones de oversampling y de filtrado. Una solución que siempre ayuda es **renderizar a más resolución**, esto es, aumentar el número de pixels de ancho y de alto de la imagen final. Subir la resolución de la imagen final suele resolver los problemas de aliasing, pero la contrapartida es que el coste computacional de generar la nueva imagen de tamaño mayor puede ser muy alto. La última opción es **retocar la imagen** resultante con algún editor gráfico como Gimp o Photoshop, que no es ninguna tontería cuando el render tarde demasiado tiempo en generarse.

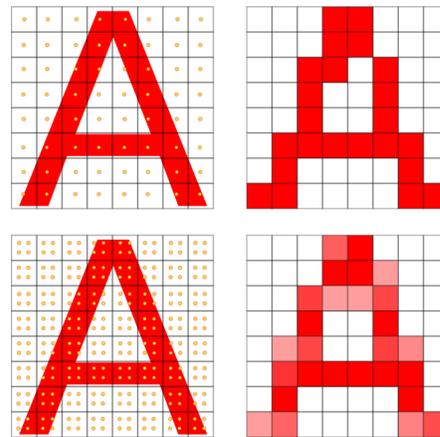


Figura 11: Soluciones de oversampling para reducir el aliasing de dientes de sierra.

MÁS INFORMACIÓN

Los libros de Watt son posiblemente los más completos para explicar las técnicas de iluminación global [1], [2]. En libro de Foley [3] se ha utilizado para extraer la información sobre *ray casting*. El algoritmo de trazado de rayos aparece descrito en otros muchos volúmenes relativos a informática gráfica como los capítulos 13 y 14 de [4] o el capítulo 9 de [5].

Gran parte de las imágenes que se presentan en este capítulo han sido obtenidas de [6], que es un excelente texto para aproximarse al rendering desde un punto de vista divulgativo. Algo similar ocurre con [7] cuyos capítulos 14-18 están dedicados al rendering.

EJERCICIOS Y CUESTIONES

1. Indique la diferencia entre los términos *iluminación global* e *iluminación local* en los algoritmos de *rendering*.
2. Explique en tres frases en qué consiste la técnica de trazado de rayos.
3. Indique si es verdadera o falsa la siguiente afirmación: En la técnica de trazado de rayos, los rayos emanan desde las fuentes de luz. Justifique su respuesta.
4. ¿A qué se refiere el término de *radiosity*. ¿Se trata de un método adecuado cuando se realizan animaciones?
5. Indique si es cierta o falsa la siguiente afirmación: En la técnica de *photon mapping*, los rayos emanan desde las fuentes de luz. Justifique su respuesta.
6. ¿En qué consiste la técnica de *final gathering*?
7. ¿Qué relación tiene la técnica conocida como *final gathering* con la simulación de brillos caústicos?
8. Indique en qué consiste la técnica de *oclusión ambiental* y para qué se utiliza.
9. Enumere dos técnicas para combatir el problema del *aliasing*.
10. ¿Qué son los brillos caústicos?

REFERENCIAS

- [1] A. Watt, *3D Computer Graphics*, tercera ed. Addison-Wesley, 2000.
- [2] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*. acm Press, Addison-Wesley, 1992.

- [3] J. D. Foley, F. D. Van, A. Van Dam, S. K. Feiner, J. F. Hughes, E. Angel, and J. Hughes, *Computer graphics: principles and practice*. Addison-Wesley Professional, 2014.
- [4] D. Hearn, M. P. Baker *et al.*, *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall., 2004.
- [5] P. Shirley and S. Marschner, *Fundamentals of Computer Graphics*, 3rd ed. Natick, MA, USA: A. K. Peters, Ltd., 2009.
- [6] J. Birn and J. Birn, "Digital lighting and rendering new riders," 2006.
- [7] V. Wissler *et al.*, *Illuminated pixels: the why, what, and how of digital lighting*. Nelson Education, 2013.