

Des jeux vidéo plus réalistes



Henrik Wann Jensen et Tomas Akenine-Möller

Grâce à l'évolution conjointe des algorithmes et des composants matériels, l'informatique graphique crée des images de plus en plus proches de la réalité.

Depuis l'émergence, au début des années 1960, de l'informatique graphique à trois dimensions, ou 3D, les spécialistes du domaine rêvent de créer des univers virtuels dits photoréalistes, impossibles à distinguer du monde réel. Créateurs, architectes, éclairagistes, passionnés de jeux vidéo et du graphisme scientifique souhaitent que les cartes vidéo puissent afficher des scènes 3D interactives.

Les concepteurs et fabricants de matériels informatiques et d'algorithmes ont fait des progrès spectaculaires en la matière ; il suffit de regarder un adolescent s'adonnant aux jeux les plus récents pour s'en convaincre. Il reste néanmoins des défis de taille, à commencer par la tâche de calcul nécessaire pour produire de telles images. Des progrès ont été réalisés à l'aide d'algorithmes ingénieux, et les générations successives des processeurs graphiques ont permis d'organiser et réaliser les calculs lourds nécessaires à la synthèse d'images.

Pourtant, afficher des images photoréalistes en temps réel reste difficile. La plupart des ordinateurs de bureau n'utilisent

qu'une minuscule fraction de leurs cycles de calcul pour effectuer les tâches de routine. Rien à voir avec l'écrasante tâche qui incombe aux applications 3D les plus avancées : le rendu d'une seule image peut demander plusieurs heures de calcul aux machines les plus rapides, afin par exemple de suivre le chemin de milliards de photons ou les diffusions de la lumière sous la surface d'objets translucides.

La quête du temps réel

Or les joueurs vidéo les plus performants réclament aujourd'hui que les scènes 3D s'affichent à raison de 60 images par seconde au moins. Cette quête du temps réel est motivée non seulement par le désir de voir des scènes d'un bon réalisme, mais aussi par le besoin d'interagir avec elles. La norme télévisuelle de 29,97 images par seconde est suffisante pour un spectateur passif ; les applications en temps réel, tels les jeux et les simulateurs de vol, doivent fonctionner à la

vitesse des réflexes humains. Il s'agit de photoréalisme en temps réel.

En tant que participants à cette entreprise de création de graphismes à rendu réaliste (l'un d'entre nous se consacre plus particulièrement à la recherche d'une plus grande vitesse, l'autre à l'élaboration d'un meilleur réalisme), nous passerons en revue les différents types de calculs requis, les dispositifs qui ont été proposés pour les alléger, et l'univers parallèle du développement de matériels capables de mener ces calculs.

Les matériels et les logiciels évoluent de conserve depuis plusieurs décennies. Le secteur du matériel, porté principalement par la soif de vitesse des joueurs vidéo, a produit des cartes graphiques d'une extraordinaire puissance. Les calculs ont été transférés des unités centrales (CPU, *Central Processing Units*) des ordinateurs aux processeurs graphiques (GPU, *Graphics Processing Units*) des cartes vidéo commerciales. Le bond des prouesses de calcul entraîne alors la création d'algorithmes capables d'accroître le réalisme.



L'ESSENTIEL

✓ La puissance des cartes graphiques actuelles permet d'afficher des images de synthèse proches de la réalité à partir d'environnements virtuels en trois dimensions.

✓ Les jeux vidéo doivent manipuler de façon interactive un volume de données colossal et ne peuvent pour l'heure restituer des images parfaitement réalistes.

✓ Le rendu fluide et réaliste de scènes 3D demande des algorithmes qui réduisent le temps de calcul tout en créant les effets visuels typiques du monde réel.

Pour représenter la complexité visuelle des environnements virtuels 3D, les applications actuelles de graphisme en temps réel, les jeux par exemple, convertissent les descriptions des scènes en millions d'unités de construction nommées primitives géométriques : des points, des lignes et des polygones – souvent des triangles –, assemblés pour former des surfaces polygonales, c'est-à-dire des surfaces constituées de polygones.

Les premiers jeux (comme *3D Monster Maze*, dont on voit une image à côté du titre) représentaient les scènes à l'aide de quelques centaines de polygones ; même si à l'époque, déjà, interagir avec ces environnements 3D pouvait être prenant, l'attrait des jeux ne devait pas grand-chose au réalisme.

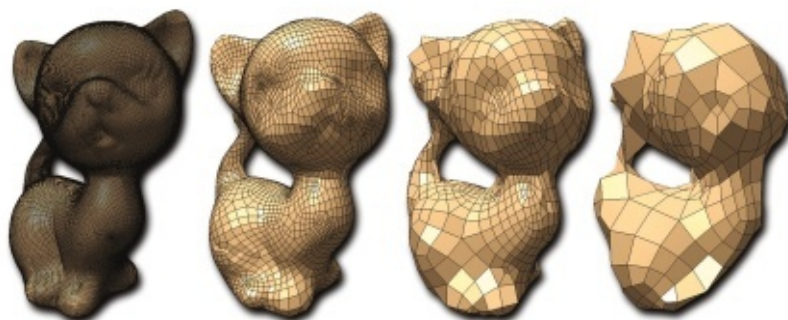
Construire une scène

Pour produire des images plus réalistes, une première étape a consisté à accroître le nombre de polygones sur les modèles 3D des scènes. Il faut bien choisir l'équilibre entre le plus grand réalisme ainsi rendu possible et le temps de calcul alors nécessaire. Le concepteur doit toujours se demander jusqu'à quel point pousser le souci du détail. Aujourd'hui, les processeurs graphiques sont capables d'afficher des scènes 3D composées de plusieurs millions de polygones à un rythme de 60 images par seconde, le point de référence des joueurs vidéo.

Si la tendance se poursuit, nous pouvons nous attendre à brève échéance à

1. LA DEMANDE DES USAGERS, couplée à l'amélioration des algorithmes et des composants matériels, rapproche l'informatique graphique de l'un de ses objectifs ultimes : l'animation au rendu réaliste. À la pointe de la technologie, le film d'animation 3D *Ratatouille*, des studios *Pixar*, a été produit par environ 850 ordinateurs équipés de près de 3 200 processeurs. Le temps moyen de calcul de chaque image était d'environ six heures, la projection se déroulant à 25 images par seconde. Chaque image était précalculée, mais il en va autrement pour les jeux vidéo, où une excellente qualité visuelle doit être conciliée avec la fluidité de l'animation, qui demande 60 images par seconde, calculées en temps réel. Les choses avancent en ce sens, comme on peut le voir avec le jeu interactif *Mirror's Edge* de la Société *Electronic Arts*, datant de 2008 (*ci-contre*). Ce jeu éblouissant parvient à afficher plus de 60 images par seconde sur des ordinateurs de bureau.

EA Digital Illusions Creative Entertainment



J. Damiens et al., Actes de SIGGRAPH Asie, vol. 27, pp. 1-9, 2008

2. POUR VISUALISER ET ANIMER DES MODÈLES 3D, et afficher rapidement les images résultantes, il faut combiner optimisation et approximation. Les algorithmes optimisent les calculs nécessaires, et des éléments de la scène 3D tels que les ombres, les réflexions et même la géométrie peuvent n'être qu'approximatifs, la précision étant sacrifiée au profit de la vitesse. Ainsi, avec la technique de rastérisation, qui crée un rendu de polygones correspondant au modèle 3D, le nombre de polygones peut être plus ou moins grand. Les deux modèles de chat du milieu permettraient un rendu beaucoup plus rapide que celui de gauche. Plus la distance entre le spectateur et le chat augmente, moins le nombre de polygones à utiliser est élevé, avec une perte de qualité réduite.

l'arrivée de matériels qui manipuleront des centaines de millions, voire des milliards, de polygones à une vitesse suffisante. L'un des fondateurs des studios d'animation *Pixar*, producteurs d'animations 3D allant de *Toy Story* à *Ratatouille*, affirmait que 80 millions de polygones seraient nécessaires pour le rendu réaliste d'une scène donnée. Il semble donc que le matériel sera bientôt au rendez-vous du photoréalisme en temps réel. Toutefois, ce dernier est loin de se résumer au nombre de polygones mis en œuvre.

Avec le matériel informatique et les logiciels actuels, le moyen le plus rapide d'afficher une scène 3D sur un ordinateur, c'est-à-dire de convertir les données géométriques 3D des objets – leurs coordonnées dans l'espace à trois dimensions – en une image formée de pixels, est la rastérisation, ou rendu de polygones, la technique utilisée par les jeux vidéo actuels.

À partir de la position de l'observateur, l'algorithme traite la scène 3D en détectant ses parties géométriques visibles et ses parties cachées (y compris les parties postérieures des objets faisant face à l'observateur). Ce qui n'est pas visible est écarté pour accélérer le calcul, puis l'algorithme détermine quels sont les sommets géométriques les plus proches de l'observateur. Les polygones formés par ces sommets sont projetés sur un écran virtuel constitué de pixels (voir la figure 3). La couleur de chaque pixel est alors définie par la couleur et les propriétés de surface (aspérités, éclat, etc.) assignées au polygone correspondant, ainsi que par l'éclairage de la scène. On fait

enfin disparaître de l'image les angles entre polygones contigus en prenant la moyenne des valeurs de leurs couleurs.

Ce sont des instructions logicielles nommées *shaders* (nuanceurs) qui déterminent la couleur et les propriétés de surface d'un polygone. En général, l'information sur la surface est fournie grâce à des « cartes de texture », images numériques plaquées sur l'objet 3D. Le placage de textures est un art en lui-même. Dans les chaînes de production des studios 3D, les artistes se spécialisent pour reproduire, par exemple, non seulement la couleur d'une orange, mais aussi la rugosité de sa surface et le brillant de la cire.

Ombres et lumière

Le placage de textures en relief, conçu par l'un des pionniers de l'infographie, l'Américain James Blinn, à la fin des années 1970, a représenté l'une des premières percées en la matière. Ce dernier a apporté de nombreuses contributions capitales à la production d'images convaincantes à partir de données 3D. Le placage de relief rend les détails de la microstructure de la surface sans alourdir la géométrie globale. Il donne pour instruction à la carte graphique de manipuler l'éclairage local comme si la surface était bosselée, les bosses étant définies par les zones claires et sombres de la carte de texture.

Un environnement 3D est éclairé par des sources lumineuses virtuelles placées dans la scène. Les techniques de rastérisation utilisent quelques équations simples

pour calculer quelle quantité de lumière issue d'une source lumineuse donnée arrive sur chaque polygone, et quelle fraction est réfléchie vers l'observateur.

Les premiers rendus de scènes 3D avaient une apparence irréaliste caractéristique, parce qu'il leur manquait un aspect essentiel du réalisme visuel : les ombres. La rastérisation a résolu ce problème en utilisant une technique qui procède à un affichage en plusieurs phases, ou passes. Par exemple, avec l'algorithme de placage d'ombres, une première image est construite du point de vue de la source lumineuse pour déterminer les polygones éclairés. Lors d'une deuxième passe, du point de vue de la « caméra », c'est-à-dire de l'observateur – différent de celui de la source lumineuse –, le calcul de couleur réalisé pour chaque polygone interroge la carte des ombres pour indiquer si celui-ci est visible depuis la source lumineuse ou s'il est dans l'ombre. La couleur du polygone est alors estimée pour rendre compte de l'ombre.

La rastérisation présente toutefois une faiblesse indiscutable pour reproduire les réflexions et les réfractions de la lumière. Les réfractions sont les déviations de la lumière quand elle traverse un milieu transparent tel que l'eau ou le verre ; comme les ombres, elles contribuent beaucoup au réalisme. Pour diverses raisons, ces deux types d'effets optiques ne peuvent être calculés au moyen de la technique de « peinture de polygones », la stratégie centrale de la rastérisation. Plusieurs propositions scientifiques ont tenté de contourner ce problème afin de donner l'impression d'effets miroirs ou de transparence, mais la méthode reste fondamentalement inadaptée à ces effets.

Il existe d'autres effets de lumière que la rastérisation ne parvient pas à reproduire. Dans les scènes réelles, les couleurs tendent à s'étaler lorsque des surfaces diffuses sont éclairées par des réflexions multiples. Ainsi, dans une pièce blanche garnie d'un tapis rouge, le tapis projette une subtile lueur rouge sur les murs blancs. Ce phénomène ne peut être simulé par rastérisation. Il en est de même des caustiques, concentrations lumineuses produites sur des surfaces par des objets transparents ou qui réfléchissent la lumière, comme celles que l'on voit au fond d'une piscine ou tout près d'un verre de vin posé sur la table.

Autre phénomène que la rastérisation n'arrive pas à simuler, la diffusion de la lumière sous les surfaces est un développement récent particulièrement important

du photoréalisme. Quand la lumière traverse la surface d'un matériau réel, elle est diffusée, en partie vers l'intérieur, et en partie vers l'extérieur, selon un effet de translucidité. La qualité visuelle distinctive de la diffusion sous la surface des objets entre en jeu dans l'apparence, entre autres, de la peau humaine, et la difficulté à la reproduire avec exactitude explique le rendu peu convaincant des visages.

Aujourd'hui, pour diverses raisons, dont les limites de la rasterisation dans le traitement des effets de lumière complexes, beaucoup de spécialistes pensent que cette technique ne pourra pas concrétiser l'affichage photoréaliste des scènes 3D en temps réel. La solution est à rechercher en partie du côté du matériel. L'une des technologies clefs du graphisme en temps réel repose sur des processeurs graphiques spécialisés capables de traiter et d'afficher rapidement quantités d'objets géométriques. Ces processeurs sont performants grâce à un haut degré de parallélisme : la tâche consistant à afficher une scène est divisée en tâches plus petites, traitées en parallèle par des unités de calcul spécialisées.

Programmer les processeurs

Pour cette raison, le graphisme peut être perçu comme un « trou noir » de puissance de calcul : plus vous injectez de la puissance pour répondre au problème, plus les consommateurs et les développeurs en demandent pour que leurs ordinateurs affichent des images toujours plus complexes. Les concepteurs des architectures matérielles, des processeurs centraux comme des processeurs graphiques, ont ces demandes du marché à l'esprit.

Depuis quelques années, on sait programmer les processeurs graphiques. Il y a dix ans, il s'agissait essentiellement d'unités à fonction fixe dont seuls certains paramètres étaient ajustables pour permettre quelques types de calculs différents. Cette rigidité limitait fortement les types d'effets graphiques possibles. Avec la flexibilité nouvelle des processeurs graphiques programmables, un programmeur peut choisir des modèles d'éclairage complexes pour tirer le maximum du matériel.

Le matériel programmable a également permis de trouver des astuces susceptibles de surmonter les limites inhérentes à la rasterisation. Par exemple, les chercheurs et les développeurs de jeux du monde entier,

en quête de scénarios de jeux toujours plus réalistes, ont développé des algorithmes approximatifs à plusieurs passes qui imitent l'étalement de couleur, les caustiques et la diffusion de la lumière sous les surfaces en implémentant la rasterisation dans des processeurs graphiques. Cependant, cette solution commence à se heurter à un mur. Ses résultats sont attrayants, pour ne pas dire époustouflants, et pourtant, pour les standards du rendu réaliste, ils ne sont pas convaincants.

En fait, afin d'obtenir un véritable photoréalisme, il faut changer la façon dont le traitement en temps réel gère la géométrie des scènes 3D et leur éclairage. La course est lancée pour développer du matériel capable de prendre en charge de tels algorithmes. L'un d'eux est le lancer de rayons.

Conceptuellement, le lancer de rayons et la rasterisation ne sont pas très différents : tous deux déterminent ce qui est rendu visible le long d'un rayon. Le lancer de rayons diffère en ceci qu'il simule des rayons lumineux qui traversent une scène 3D, y compris les multiples nouveaux rayons nés de la réflexion de lumière par les éléments géométriques de la scène. En faisant remonter les rayons lumineux jusqu'à une source lumineuse, on peut rendre compte de façon assez naturelle des phénomènes physiques associés à la lumière, non seulement de la réflexion et la réfraction, mais également d'effets particuliers tels que les diffusions lumineuses multiples et les caustiques.

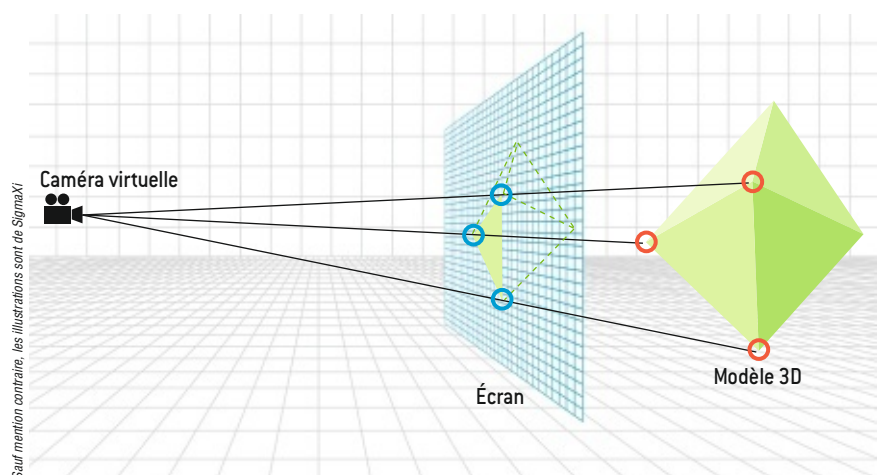
Le lancer de rayons est un algorithme élégant, relativement simple à programmer.

LES AUTEURS



Henrik WANN JENSEN est professeur à l'Université de Californie à San Diego. Tomas AKENINE-MÖLLER est professeur d'informatique à l'Université de Lund, en Suède, et collabore avec Intel.

© American Scientist



3. LES ALGORITHMES DE RASTERISATION affichent un modèle 3D d'une scène en projetant, depuis la caméra virtuelle qui détermine la prise de vue, des rayons sur les sommets géométriques (en rouge) de la scène. Ces sommets définissent des polygones qui sont alors associés aux pixels d'un écran virtuel (en bleu). L'apparence de surface assignée aux polygones du modèle est alors calculée et projetée sur l'écran pixellisé pour créer une image.