

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.

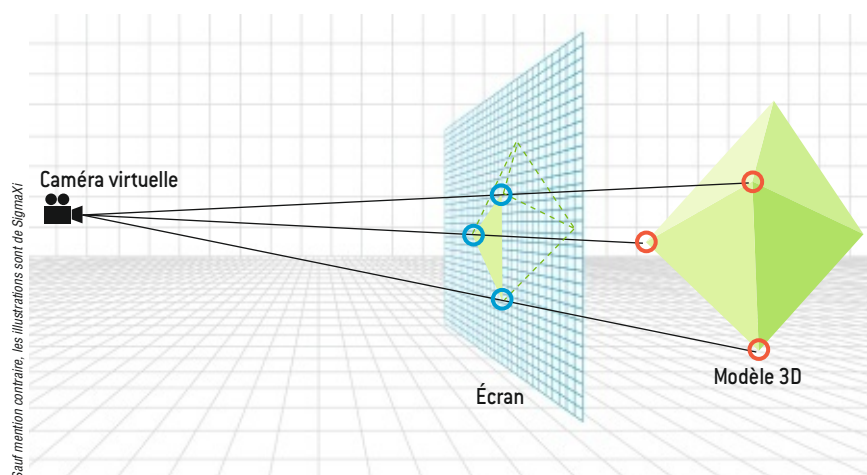
Le matériel programmable a également permis de trouver des astuces susceptibles de surmonter les limites inhérentes à la rasterisation. Par exemple, les chercheurs et les développeurs de jeux du monde entier,

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.

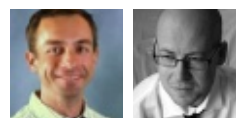
Conceptuellement, le lancer de rayons et la rasterisation ne sont pas très différents : tous deux déterminent ce qui est rendu visible le long d'un rayon. Le lancer de rayons diffère en ceci qu'il simule des rayons lumineux qui traversent une scène 3D, y compris les multiples nouveaux rayons nés de la 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.



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.

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