Résumé

Beaucoup de programmes de visualisation, de planification de trajectoire, etc., utilisent intensivement des calculs de visibilité. Si ces calculs de visibilité ne constituent qu'une petite partie de ces programmes, ils sont en revanche responsables d'une grande partie du temps d'exécution de ces programmes : leur efficacité est donc cruciale.

Les algorithmes traditionnels de calculs de visibilité ont deux défauts : ils effectuent – inutilement – des calculs sur des objets non visibles et refont tous ces calculs à chaque nouvelle requête, même si les changements avec la requête précédente sont minimes.

Pour remédier à ces inconvénients dans le cadre de scènes polygonales bidimensionnelles, nous nous servons d'une structure de données – le complexe de visibilité – qui code toutes les relations de visibilité entre objets d'une scène. Après avoir montré comment construire de façon optimale le complexe de visibilité, nous montrons comment il permet d'utiliser la cohérence spatiale de la scène dans les calculs de polygones de visibilité. Nous montrons aussi comment il permet d'utiliser la cohérence temporelle dans le maintien d'une vue autour d'un point se déplaçant dans la scène.

Nous étudions ces algorithmes non seulement d'un point de vue théorique mais aussi d'un point de vue pratique. Nous avons programmé ces algorithmes et effectué des comparaisons expérimentales entre algorithmes. Nous nous sommes aussi intéressés aux problèmes de dégénérescences et d'imprécisions des calculs numériques qui se posent dès que les programmes sont exécutés sur des données « réelles ».

Mots clés : Géométrie algorithmique, visibilité, analyse d'algorithmes, comparaison expérimentale, traitement des cas dégénérés, traitement des imprécisions numériques.

Abstract

The core part of computer programs such as visualization softwares, rendering engines or robotics path planners lies in visibly computations. Although the corresponding algorithms are only a small part of the whole application, they are responsible for most of the running time and their efficiency is therefore crucial.

Traditional visibility computation algorithms have two drawbacks: They perform useless computations on invisible objects, and recompute everything from scratch for every single query, even if the changes undergone by two successive solutions are minor.

We show how to remedy these problems in the context of polygonal scenes in the plane by using the *visibility complex* data structure, which encodes all the visibility relations between objects of the scene. First, we present an optimal algorithm for computing the visibility complex. Second, we show how to use this data structure to compute visibility polygons and to maintain views around a moving point in the scene.

Both the theoretical complexity and the practical performances of our algorithms are presented. Finally, we explain how to handle the degeneracies and numerical imprecisions caused by real-world data sets.

Keywords: Computational geometry, visibility, algorithmic complexity, experimental analysis degeneracies, numerical imprecisions.

Table des matières

Liste des figures