ВСТУП

Досягнення в технологічній галузі останніх років стали основою для полігонального моделювання, яке є одним з найбільш поширених способів представлення 3D оболонок. Проте ці моделі часто є дуже складними: нерідко поверхні містять декілька мільйонів полігонів. На сьогоднішній день багато компаній розробляють продукти з використанням систем автоматизованого проектування. Лазерні сканери, системи комп'ютерного зору і медичні пристрої візуалізації можуть створювати високо деталізовані поверхні. Моделі, отримані методом поверхневої реконструкції та екстраполяції, часто являють собою досить щільні сітки з рівномірним розподілом точок на поверхні. Програмні додатки в цій сфері варіюються від розподілених віртуальних середовищ та кінцевих елементів до спецефектів у комп’ютерних іграх та кіно, заснованих на полігональному моделюванні поверхонь. У всіх цих додатках існує відповідність між точністю, з якою відбувається моделювання поверхні, і кількістю часу, який потрібен для її обробки. Щоб досягти прийнятного часу роботи, часто доводиться замінювати складні моделі більш простими наближеннями. Модель, яка фіксує дуже дрібні деталі поверхні, може бути дійсно потрібна при створенні архівних наборів даних; це допомагає гарантувати, що додатки, які пізніше будуть обробляти модель, мають достатні і точні дані. Однак для багатьох додатків потрібно набагато менше деталей, ніж в повному наборі даних. Спрощення поверхні є цінним інструментом для підгонки великих наборів даних до потреб окремих додатків і для створення більш економічних моделей поверхонь. Наприклад, додаток, що намагається виміряти деяку властивість поверхні, скажімо, об’єм, може прийти до розумної оцінки, маючи в якості вхідних даних дуже просту модель. Отже, як наслідок, існує значний попит на ефективні алгоритми спрощення тріангуляційних моделей. Протягом останніх десятиліть було запропоновано чимало таких алгоритмів, і, загалом, можна сказати, що вони широко різняться за підходами, варіативністю вхідних даних, складністю реалізації та якістю отриманого результату. Тому, як правило, алгоритми, що є дуже простими в реалізації і швидкими в роботі, створюють спрощені сітки, які є візуально небажаними. Інші створюють більш приємні апроксимації, але є повільними або підходять для спрощення моделей лише певного типу. Отже, нашою метою було створення такого алгоритму спрощення, який мав би змогу спрощувати моделі будь-якого типу, при цьому був не дуже складним у реалізації, достатньо швидким у роботі, а також давав результати, візуально максимально наближені до оригіналу. На практиці ж, який алгоритм найкраще підходить для виконання певного спрощення, значною мірою залежить від характеристик вхідної моделі та бажаних атрибутів апроксимації.