

Zápočtová práce z předmětu ADS II

Konstrukce Voroného diagramu pomocí Fortunova algoritmu

DOKUMENTACE

Richard Jedlička

Zadání

Úkolem zápočtové práce bylo naimplementovat Fortunův algoritmus pro konstrukci Voroného diagramu ze zadané množiny bodů v rovině.

Popis programu

Technické detaily

Program je napsán v jazyce C++ s využitím knihovny SDL a OpenGL pro grafické zobrazení průběhu algoritmu a výsledného diagramu.

Program je odladěný a otestován na operačním systému Linux na platformě x80_64.

Kompilace

Pro snadnou kompilaci je využit systém CMake.

Kompilaci provedete tak, že v kořenové složce programu vytvoříte adresář **build**, přemístíte se do něj a spustíte příkaz **cmake . . /** výpis vás upozorní na případné problémy. Po bezchybném proběhnutí spustíte příkaz **make**. Po proběhnutí kompilace bude ve složce **bin** v kořenovém adresáři programu zkompileovaný binární spouštěcí soubor.

Ovládání programu

Program se spustí souborem **bin/voronoj**. Má volitelný parametr, a to cestu k textovému souboru obsahující seznam bodů, ze kterých bude konstruován diagram. Body jsou v souboru zadány pomocí souřadnic **x y**, každý bod na jednom řádku. Hodnoty souřadnic mohou být téměř libovolné (samozřejmě s ohledem na velikost datového typu **double**), ale do výpočtu budou zahrnuty pouze ty, které se vejdou do okna **640 x 480 pixelů**. Ukázkový vstupní soubor **bin/vstup** je přiložen k programu.

EDITAČNÍ REŽIM: Po spuštění programu se objeví okno s černým pozadím, případně i se vstupními body. To je editační režim, kdy je možné klikáním přidávat a ubírat body. Kliknutím **levým tlačítkem** myši se bod přidá na pozici kurzoru. Kliknutím **pravým tlačítkem** myši odstraní naposledy přidáný bod. Pro vstup do režimu pro výpočet Voroného diagramu stiskněte klávesu **enter**. Aktuální rozmístění bodů si můžete uložit klávesou **S**. Pro uložení se použije vstupní soubor, pokud byl zadán, jinak soubor s názvem **vstup**. Pro ukončení programu klávesou **escape**.

VÝPOČETNÍ REŽIM: Ve výpočetním režimu se barva pozadí změní na bílou. Nyní máte dvě možnosti. Buď výpočet krokovat po jednotlivých událostech a sledovat postupnou konstrukci diagramu, nebo si nechat vykreslit celý diagram hned. Pro krokování použijte klávesu **mezerník** a pro vykreslení celého diagramu klávesu **enter**. Během výpočtu si můžete všimnout červených postupně se objevujících bodů. Ty značí kružnicové události. Zobrazování kružnicových událostí lze vypnout (případně zapnout) klávesou **C**. Během vykreslování se diagram průběžně ukládá do souboru s názvem **vystup**. Hrany se v průběhu ukládají celé i s konci mimo obrazovku, na konci se ale oříznou na viditelnou velikost. Pro návrat do editačního režimu stiskněte klávesu **escape**.

Popis implementace

Jak již bylo zmíněno v zadání, byl použit **Fortunův algoritmus**. Je to jeden z rodiny tzv. zametacích algoritmů. Funguje na principu, že z jedné strany plochy s body se pohybuje zametací přímka a „zametá“ jednotlivé události. Přímka se samozřejmě nepohybuje spojitě, ale diskrétně po skocích (po událostech). Události jsou dvou typů: místní a kružnicové. Místní událost zanáčí, že je na řadě další vstupní bod. Připomeňme, jak Fortunův algoritmus konstruuje Voroného diagram. Abychom určili hranice mezi jednotlivými množinami bodů spadajícími ke konkrétním místům (vstupní body) využijeme definice paraboly. Parabola je množina bodů stejně vzdálených jak od ohniska tak od řídící přímky. Za řídící přímku v našem případě považujeme přímku zametací a

ohniska jsou vstupní body. Průniky parabol, které „vznikají“ kolem ohnisek nám „kreslí“ hrany Voroného diagramu. Tedy místní událost zpracujeme tak, že si vytvoříme novou parabolu a tu vhodně zařadíme mezi již vzniklé paraboly a poznamenáme si vznik nových hran. Tím nám vzniká tzv. pobřežní linie, seznam částí parabola a průsečíků mezi nimi. Někdy se může stát, že se některé paraboly rozevřou natolik, že pohltnou jednu část některé paraboly. Tuto událost je potřeba nějak zpracovat a od toho tu jsou právě kružnicové události. Kružnicová událost nastává, pokud se najdou některé tři oblouky jejich tři ohniska leží na společné kružnici. Střed této kružnice nám udává místo, kde k pohlcení dojde a nejnižší bod kružnice zas udává polohu zametací přímky, kdy dojde k pohlcení. Pohlcení proběhne tak, že sbíhající se hrany se spojí a vytvoří vrchol diagramu. Odtud dále pokračuje jedna či více hran. Místní události mohou některé kružnicové události invalidovat (kvůli nim nenastanou), taková událost se nebude zpracovávat, jen se přeskočí.

Pro ukládání všech potřebných informací potřebujeme vhodné datové struktury. Na ukládání událostí je využita halda (třída `EventCalender`) třídící podle y-nové souřadnice události. Pobřežní linie je reprezentovaná binárním stromem (třída `Coastline`), kde listy zastupují jednotlivé části parabol a vnitřní vrcholy jsou jejich průniky. Na samotný diagram je využit vektor (proměnlivé pole) hran diagramu (třída `VoronoiDiagram`). Hrany jsou určeny svými koncovými body. Body jsou buďto explicitně zadané (s pevnou konečnou hodnotou – hrcholy diagramu), nebo implicitně (průsečíky – drží informace o ohniscích a způsobu jak vypočítat aktuální polohu průsečíku). Celý algoritmus zastřešuje třída `FortuneAlgorithm`.

Algoritmus by se tedy dal popsat takto:

Dokud `EventCalender` není prázdný dělej:

- ◆ vyber nejbližší událost
- ◆ pokud není validní, vybírej další, dokud nějakou nenajdeš nebo nebude prázdný `EventCalender`
- ◆ pokud držíš validní událost, zpracuj ji
 - pokud je událost místní
 - přidej novou parabolu do `Coastline`
 - vytvoř novou hranu diagramu
 - přepočítej kružnicové události
 - pokud je událost kružnicová
 - odstraň část paraboly z `Coastline`
 - vytvoř vrchol diagramu a ukonči sbíhající se hrany
 - vytvoř nové hrany
 - přepočítej kružnicové události

Celý program i s grafickými prvky obstarává třída `GUI`.

Co se týče časové složitosti, tak operace s kalendářem událostí zaberou $O(\log n)$, taktéž i práce s pobřežní linií (binární strom). Přidání hrany do diagramu v čase konstantním, takže zpracování jedné události zabere maximálně $O(\log n)$. Kružnicových událostí je určitě méně než místních. Takže celková časová složitost je řádově **$O(n \log n)$** .

Prostorová složitost je **$O(n)$** .