String art

Michal, Juli, Jáchym

1. prosince 2024

1 Co je to string-art?

String-art je výtvarná technika, při níž se snažíme napodobit nějaký obrázek pomocí k nití[1], které natahujeme mezi špendlíky rozmístěnými po obvodu kruhu. Protože string-art je činnost veskrze praktická, máme k dispozici pouze konečně mnoho špendlíků (n).

Náplň naší konfery ovšem nespočívala tolik v natahování nití mezi špedlíky[2], ale spíše v hledání algoritmu, který určí, mezi kterými dvojicemi špendlíků má vést linie, aby byl vytvořený obrázek co nejpodobnější námi zvolenému originálu. Tento problém více či méně úspěšně řeší dva námi vymyšlené algoritmy.

2 Algoritmy

V této sekci představíme algoritmy, popíšeme jejich výhody i nevýhody a určíme jejich asymptotickou časovou složitost pomocí notace O. Jestliže jste s touto notací neseznámeni, doporučujeme si o ní něco přečíst.

2.1 Všechny možnosti!

Asi jednodušší myšlenka, která nás při řešení napadla a stala se proto základem našeho prvního algortimu, byla: zkusíme všechny možné čáry a vybere jen k nejlepších. Aby to nebylo tak jednoduché, musíme si nějak chytře definovat, jak dobrá je každá čára, abychom mohli vybrat ty nejlepší.

Tento problém se dá řešit následujícím způsobem: na prázdné plátno nakreslíme jakoukoli čáru a poté celý tento obrázek odečteme od toho původního, abychom viděli, jak moc se liší. Díky tomu, že zatím řešíme pouze černobílé obrázky, stačí odečíst každý pixel z jednoho obrázku s korespondujícím pixelem z toho druhého. Všechny tyto rozdíly odpovídajících si pixelů poté sečteme, umocníme na druhou a vydělíme počtem pixelů.[3] Této hodnotě dále budeme říkat rozdíl čáry a originálu. Tento postup zopakujeme pro všechny další možné čáry.

Následně všechny čáry podle této hodnoty seřadíme a vybereme k nejlepších, tedy k těch s nejmenším rozdílem čáry a originálu.

Teď už by to mohlo vypadat, že jsme hotoví a můžeme si užívat string-artu, jak jen budeme chtít. To by se nesměl v našem dosavadním řešení skrývat jeden menší problém. Protože každou čáru zkoušíme na plátno přidat právě jednou, je čára vždy úplně černá. To znamená, že v každém bodě čára buď je, nebo není, což nám ve výsledku vytvoří velmi nepřirozené obrázky, neboť nemáme možnost velmi tmavá místa odlišit od těch jen trochu tmavých.

Abychom toto vyřešili, musíme nějak zařídit, aby mezi dvěma špendlíky mohlo vést více než jedna čára. Toho bychom mohli dosáhnout triviálním rozšířením toho algoritmu. Prostě před přidáním každé čáry seřadíme všech n^2 čar. To ovšem z časové složitosti $O(n^2)$ udělá $O(kn^2)$. Čímž jsme si docela dost pohoršili, protože $k \gg n$. Naštěstí existuje i lepší způsob, ten budiž nalezen v následující podsekci.

[1] V průběhu přeformulujeme naši úlohu tak, aby namísto konstantního počtu nitek byla konstantní odchylka od originálu, kterého se snažíme dosáhnout.

[2] O to se Jáchym rovněž pokusil (zda úspěšně či neúspšně měli možnost posoudit všichni účastníci podzimního soustředění).

[3] Této metodě se v praxi říká MSE (Mean Squared Error) a pro její používanost jsme se rozhodli pro ni.

```
def string_art(n,k)
    lines = [[(start,end) for start in range(n)] for end in range(n)]
    lines.sort(key= lambda line: difference(line))
    return lines[k:]
```

Obrázek 1: První algoritmus ve zjednodušené verzi

```
def string_art(n,k)
    final = [0]
    while len(final)<=k:
        start = final[-1]
        lines = [(start, end) for end in range(n)]
        lines.sort(key= lambda line: difference(line))
        final.append(lines[0][1])
    return final
```

Obrázek 2: Druhý algoritmus ve zjednodušené verzi.

2.2 Postupně zlepšujeme

Abychom napravili obě nedokonalosti našeho prvního řešení, musíme celou předchozí ideu překopat. Místo toho abychom v každém kroku počítali rozdíl čáry a originálu pro všechny možné dvojice bodů, začneme pouze s jedním random vybraným vrcholem a spočítáme tuto hodnotu pro všechny čáry z něho vedoucí. Z těch vybereme tu nejlepší a její koncový bod nastavíme jako počátek, opět spočítáme rozdíl každé čáry z něho vedoucí a originálu. Z koncového bodu nové nejlepší čáry spustíme celý proces znovu. Toto opakujeme k-krát

Tímto jsme efektivně vyřešili oba problémy minulého algoritmu. Ovšem vyskytnul se nám jeden další. Jestliže bude na obrázku jeden velmi černý pruh, algoritmus bude dávat čáry jen do něj a nikam jinam.

Řešení tohoto je jednoduché: po přidání každé čáry do našeho string-artu stejnou čáru ale bílou, nakreslíme do původního obrázku.

Tento algoritmus má časovou složitost $O(kn)$. V praxi se nám vyplatilo nemít žádný pevný počet čar k, ale radši porovnávat, jak daleko je celý string-artový obrázek od originálu. Jakmile se tento rozdíl dostane pod nějakou konstantu, celý program zastavíme. Důvod, proč jsme pro popsání neuvažovali tuto variantu, je kvůli její nejasné časové složitosti.

3 Výsledky

S programy představenými v předchozích kapitolách se nám povedlo převést spoustu obrázku do string-artové podoby. Mezi ty nejzajímavější patří spící Dláža. Také se nám podařilo vytvořit strig-art video tak, že jsme do string-artu převedli každý snímek zvlášť, a poté je zas spojili. Mezi poslední zajímavosti patří barevný string-art. Toho jsme docílili tak, že jsme každý z barevných kanálů převedli samostatně na string-art variantu a poté všechny tři jednobarevné string-arty spojili dohromady.