

2. Automatické rozpoznávání objektů v mapách

Matěj Klimeš, Tomáš Zbíral
ZS 2024/25, číslo skupiny: 2, datum zpracování: 5.11.2024

1 Zadání

1. Implementujte algoritmus pro vyhledávání vzorů v Müllerově mapě Čech na základě obrazové korelace. Vyberte jeden z objektů typu obec s kostelem jako okno pro vyhledávání a na základě vhodné hodnoty korelace vyhledejte všechny pozice obcí s kostelem na mapovém listu. Výsledek odevzdejte jako pixelové souřadnice kostelů.
2. Implementujte algoritmus obrazové segmentace pomocí metody k-Means nad obrazem Topografické mapy ČSSR v měřítku 1:25000. Výsledkem bude extrahovaná plocha lesů očištěná o vrstevnice a další překryvnou kresbu, naopak budou z lesů odečteny lesní průseky. Otvory v lesním prostoru větší než 50 pixelů zachovajte. Výsledek odevzdejte jako pole pixelových souřadnic.

1.1 Řešené bonusové úlohy

Nad rámec zadání byly řešeny tyto bonusové úlohy:

- Prořeďte souřadnice o vícenásobná opakování.
- Vytvořte skript, který přijímá jako vstup 5 vzorů obcí s kostelem, které jsou průměrovány.
- Vytvořte segmentovaný obraz mapy pomocí metody GraphCut.

Kromě těchto bonusových úloh byl vytvořen ještě algoritmus pro odfiltrování liniových objektů, které v masce zůstali.

2 Teoretický úvod

Algoritmus pro segmentaci využívající korelace je jedním z nejjednodušších algoritmů, které lze pro tuto úlohu využít. V prostředí MATLAB se pro tuto úlohu hodí využití funkce normxcorr2, která se řídí následující rovnicí [3]:

$$\gamma(u, v) = \frac{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}] [t(x - u, y - v) - \bar{t}]}{\sqrt{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x - u, y - v) - \bar{t}]^2}}, \quad (1)$$

kde f je obrázek, \bar{t} je průměrná hodnota vzoru a $\bar{f}_{u,v}$ je průměrná hodnota $f(x, y)$ pod vzorem. Prořeďení souřadnic o vícenásobná opakování proběhl tak, že jsme nastavili minimální vzdálenost vzorů $d = \sqrt{dX^2 + dY^2}$, seřadili jsme nalezené obrazy kostelů podle velikosti korelačního koeficientu a poté jsme iterovali přes tyto obrazy a souřadnice obrazu porovnávali vůči všem již zařazeným obrazům. Náš skript přijímá jako vstup 5 vzorů obcí s kostelem, které nejsou průměrovány přímo. Místo toho vypočteme korelační koeficient pro všechny

5 vzorů a ten pak průměrujeme. Nejdříve nám to přišlo lepší, ale obě metody by měly mít stejné výsledky a ta, kde by se průměrovaly vzory přímo by měla být rychlejší.

K-means je algoritmus poprvé definovaný Stuartem Loydem [1] v roce 1957. Algoritmus využívá myšlenky, že objekty, ze kterých chceme vytvořit shluky, jsou chápány jako body v euklidovském prostoru. Algoritmus funguje tak, že iterativně přiřazuje body k centroidům (v první iteraci voleným náhodně), které jsou poté přeypočítány do těžiště shluků. Poté jsou body přiřazeny k novým centroidům a výpočet se opakuje. Tento výpočet je konvergentní.

3 Pracovní postup a získané výsledky

V rámci první úlohy jsme dostali zadaný **mapový list Müllerovi mapy Čech** ve formátu **.jpg**. Vyvořili jsme skript, pomocí kterého jsme z mapy ručně vybrali 5 vzorů a jejich parametry uložily do matice **positions.mat** ve formátu (**left, top, width, height**). Zadaný obraz mapového listu, stejně jako vzory jsme převedli do barevného systému **lab** a pro segmentaci využili kanál **I**. Pro tento obraz jsme vypočetli korelační koeficient pomocí funkce **normxcorr2** [2] a ten pak porovnávali vůči námi zvolené minimální hodnotě korelačního koeficientu (Ukázka kódů 1). Kód pro filtraci nalezených vzorů o vícenásobná opakování lze vidět zde (Ukázka kódů 2).

```
1 mincorr = 0.547;
2 [r,c] = size(corr);
3 count = 0;
4 hold on
5 for row=1:r
6     for col=1:c
7         actual_coef = abs(corr(row,col));
8         if actual_coef >= mincorr
9             count = count + 1;
10            detections(count,1) = row;
11            detections(count,2) = col;
12            detections(count,3) = actual_coef;
13        end
14    end
15 end
```

Ukázka kódů 1: MATLAB kód pro detekci symbolů obcí s kostelem.

V rámci druhé úlohy jsme dostali zadaný **list technické mapy TM25** ve formátu **.jpg**. Tento list jsme převedli do 3 různých barevných systémů a to RGB, lab a hue. Pro všechny tyto barevné systémy jsme provedli segmentaci pomocí k-means algoritmu, abychom mohli porovnat, který z barevných kanálů je nejvhodnější pro segmentaci lesa. Jako nejvhodnější se jevil kanál **b** ze systému **lab** (Obrázek 1). Tento obrázek jsme pak očistili od liniových prvků, které zůstali v masce a to pomocí vlastního algoritmu, který funguje tak, že projíždí masku pomocí čtvercového okna a kontroluje, jak velká část pixelů v okně je součástí masky. Pokud je součástí pouze příliš malá část, všechny pixely v tomto okně jsou vyřazeny z masky. Výsledek tohoto algoritmu je vidět na Obrázku 2. Tuto masku následně "zkompaktníme" pomocí morfologického uzavření a zkombinujeme s maskou, kterou jsme získali pomocí metody GraphCut. Díky kombinaci těchto dvou masek získáme nejvhodnější masku, jelikož obě metody mají své výhody. K-means algoritmus dobře segmentuje i malé části lesa, metoda GraphCut naopak dobře segmentuje i okraje velkých lesů. Masku, kterou jsme získali kombinací těchto masek je vidět na Obrázku 3.



Obrázek 1: Segmentace lesa pomocí algoritmu k-means v barevném kanálu **b** z barevného systému **lab**.



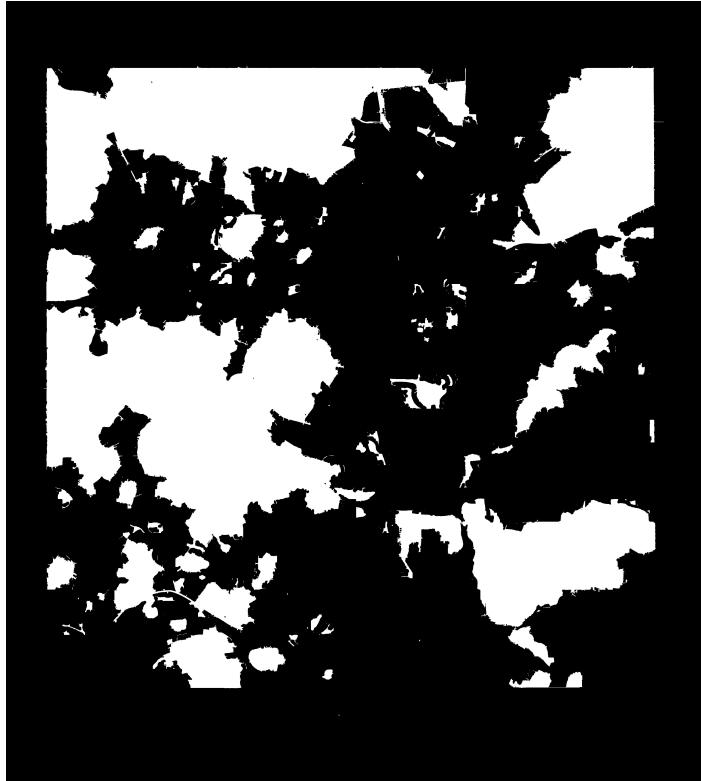
Obrázek 2: Maska očištěná o liniové prvky.

```

1 unique_detections = [];
2 min_distance = 10;
3
4 for i = 1:size(detections, 1)
5     current_detection = detections(i, 1:2);
6     if isempty(unique_detections)
7         unique_detections = current_detection;
8     else
9         distances = sqrt(sum((unique_detections - current_detection).^2, 2));
10        if all(distances > min_distance)
11            unique_detections = [unique_detections; current_detection];
12        end
13    end
14 end

```

Ukázka kódu 2: MATLAB kód pro filtrace vícenásobných opakování.



Obrázek 3: Kombinovaná maska.

4 Závěr

Řešení první úlohy mi přišlo jednoduché a zřejmé. Bylo by zajímavé studentům ukázat, jaké výsledky lze pro stejnou úlohu dostat pomocí složitějších algoritmů, jako např. algoritmy strojového učení. V druhé úloze mi přišlo zajímavé porovnávat jednotlivé kanály a jejich chování ve spojení s algoritmem k-means. Je zajímavé, že některé kanály či barevné systémy jsou zcela nevhodné pro tento algoritmus. Bylo by také zajímavé, kdyby nějaká z podúloh vyžadovala po studentovi nějakou míru optimalizace hyperparametrů daných funkcí v programu MATLAB.

5 Přílohy

Příloha 1 - MATLAB skript pro úlohu A

Příloha 2 - MATLAB skript pro k-means algoritmus v úloze B

Příloha 3 - MATLAB skript pro následné operace s maskou získanou pomocí k-means metody

V Praze dne 5.11.2024

Matěj Klimeš, Tomáš Zbíral

Reference

- [1] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [2] MathWorks. *normxcorr2*. The MathWorks, Inc., Natick, Massachusetts, 2023. Accessed: 2024-11-11.
- [3] Jae-Chern Yoo and Tae Hee Han. Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28:819–843, 2009.