



Hodnocení kvality řezu

Semestrální práce ZDO

1 Zadání

1.1 Úvod

Vzdělávání na dálku prostřednictvím on-line prostředků se v minulosti osvědčilo a epidemiologický a energeticko-ekonomický přínos nadále zvyšuje jeho význam. Některé předměty se však zdají být téměř nemožné vyučovat prostřednictvím videa. Mezi takové předměty patří výuka chirurgie. Cílem naší aplikace je vytvořit program, který byl napomáhal výuce právě tohoto předmětu.

Pomocí nástrojů umělé inteligence a strojového učení se snažíme snížit nároky na učitele a usnadnit objektivní hodnocení správnosti výkonu studentů. Výuka pak probíhá prostřednictvím videochatu a měření na základě umělé inteligence doplňuje hodnocení výsledků.

1.2 Úkol

Cílem této úlohy je vyhodnotit kvalitu chirurgického šití na základě obrazu řezu a stehu. Studentům bude poskytnuta sada obrázků zobrazujících řezy a stehy provedené během chirurgického zákroku. Úkolem studentů bude pomocí metod počítačového vidění extrahovat a segmentovat obrazová data a poté pomocí metod strojového učení provést kvalitativní analýzu šití. Studenti budou hodnotit kvalitu šití na základě parametrů, jako je rovnoměrnost a kolmost. Výsledky analýzy budou prezentovány pomocí grafických vizualizací a metrik kvality sešívání.

- Očekává se, že odevzdaná práce bude mít podobu odkazu na git repozitář.
- Název úložiště obsahuje identifikaci vašeho týmu.
- Závěrečná zpráva ve formátu PDF je v kořenovém adresáři úložiště.
- V úložišti je soubor `src/run.py`.
- Prvním argumentem je vždy výstupní soubor `.json`
- Druhým argumentem může být `-v`, který spustí vizuální režim s ladicími obrázky.
- Soubor `src/run.py` přijímá jako argumenty názvy souborů s obrázky.

Struktura výstupního souboru popisuje linii řezu a její křížení se stehy. Polohy křížení se měří v pixelech od levého začátku linie řezu.

2 Řešení

Protože úloha byla zadána do dvojice, bylo možné vyvíjet paralelně dva nezávislé postupy. Každý člen týmu rozvíjel vlastní nápad a postup, za vzájemné spolupráce a sdílení informací o tom, které nápady fungují lépe a které hůře.

Cílem tohoto postupu bylo zvýšit pravděpodobnost nalezení funkčního řešení v kratším čase. Na závěr bylo v plánu použít lepší z obou postupů a nebo jejich kombinaci.

2.1 První způsob

První způsob zpracování sady obrázků zobrazujících řezy a stehy provedené během chirurgického zákroku, umožňuje nalezení pouze hlavní přímky, která představuje řez. V tomto způsobu se dále nepokračovalo k nalezení stehů, jelikož se paralelně přišlo na další řešení, které umožňuje nalezení řezů včetně stehů najednou.

Nalezení hlavní přímky (jizvy)

V tomto případě se k nalezení hlavní přímky využívá především Houghova transformace. Metoda aproximuje řez pomocí 1 nebo 2 přímek (*1line*, *2line*).

Postup algoritmu:

- 1) Převod obrázku z RGB do HSV
- 2) Prahování (práh = Otsu)
- 3) Morfologická operace → skeletonizace
- 4) Houghova transformace → čáry → body (začátek a konec)
- 5) Lineární regrese (aproximace bodů přímkou)
- 6) Zpracování a filtrace vzdálených bodů od přímky podle threshold
- 7) Přepočet přímky po odstranění vzdálených bodů (*1line*)
- 8) Výpočet lomené čáry → aproximace jizvy 2 přímkami (*2line*)
- 9) Finální reprezentace jizvy - výběr výsledku podle kritériální funkce J

Popis algoritmu:

Nejprve se převede obrázek z RGB do HSV, jehož hlavní výhodou je to, že i pro obrázky se špatným jasovým rozložením, dokáže dobře segmentovat jednotlivé části, nicméně někdy HSV může naopak fungovat hůře než v oblasti RGB. HSV má 3 kanály (Hue, Saturation, Value), výsledný obrázek je v této metodě před prahováním určen jako lineární kombinace těchto tří kanálů, váhy jednotlivých kanálů byly nastavovány experimentální metodou.

```
# převod obrázku z RGB do HSV
img = skimage.color.rgb2hsv(img_original)

# nastavení různých vah jednotlivých kanálů
hue = 0.15 * img[:, :, 0]
saturation = (0.7 * img[:, :, 0]) * img[:, :, 1]
value = 0.7 * img[:, :, 2]

# lineární kombinace HSV
img_hsv = hue + saturation - value
```

Po převodu do HSV obrázků projde prahováním, kde je práh zvolen automaticky pomocí Otsuovi metody. Výsledkem je tedy naprahovaný binární obrázek, který je následně podroben morfologické operaci, konkrétně skeletonizaci. Nabízí se zde možnost použít i jiné morfologické operace, ať už před nebo po skeletonizaci, jako je například eroze/otevření pro odstranění malých částí, nicméně to v tomto případě není potřeba. Skelet obrázku je použit pro Houghovu transformaci, konkrétně tedy pro pravděpodobnostní Houghovu čáru (*probabilistic hough line*). Tato metoda se snaží najít čáry, které jsou velmi pravděpodobné. Lze zde nastavovat několik parametrů, hlavními parametry je délka čáry a největší možná mezera mezi body pro hledání čáry. Nejprve byla snaha nastavovat parametr délky čáry co možná největší, protože to představuje hlavní (nejdelší) přímkou (řez), nicméně později se ukázalo, že efektivnější je hledat menší čáry, a později si odfiltrovat čáry, respektive body, které nejsou součástí hlavní jizvy.

```
# Prahování - Otsuova metoda
threshold = filters.threshold_otsu(img_hsv) # práh
binary_image = img_hsv > threshold # naprahovaný obrázek

# Skeletonizace
skeleton = skimage.morphology.skeletonize(binary_image)

# Detekce přímek pomocí Houghovy transformace
lines = probabilistic_hough_line(skeleton, threshold=20, line_length=5, line_gap=2)
```

Houghova transformace nalezne čáry, které jsou reprezentovány počátečním a koncovým bodem. Z těchto bodů se pomocí lineární regrese vypočítá přímka. Následně se pro každý bod vypočítá vzdálenost daného bodu od přímky. Pomocí práhu (`threshold`) se odstraní příliš vzdálené body. Je to z důvodu, že nalezené přímka aproximuje řez, nicméně z Houghovi transformace lze dostat i čáry, respektive body, které reprezentují např. vpich. Nicméně tyto body jsou od přímky vzdálenější než body na řezu, kolem kterého by se v blízkosti měla pohybovat aproximovaná přímka. Práh (`threshold`) je nastavený automaticky pomocí histogramu (lokální minimum) a navíc jsou přidány konstanty pro minimální a maximální práh.

Po odstranění nepotřebných bodů se opět vypočítá pomocí lineární regrese přímka, která představuje jedno z možných finálních řešení. Druhým potenciálním řešením je aproximace 2 přímkami. Toho lze dosáhnout tím, že z dostupných bodů, kromě počátečního a koncového bodu jizvy, se postupně zkouší rozdělit seřazené body na dvě části, dále se z těchto dvou částí aproximují dvě přímky a vypočte se celková vzdálenost bodů od daným přímek (= kritériální funkce J), tento postup se pro každý bod opakuje. Výsledkem je aproximace dvou přímek, pro které je nejmenší hodnota J . Finálním výsledkem je 1 nebo 2 přímky, které aproximují jizvu podle toho, jestli je rozdíl kritériální funkce $J \geq 25$, potom je jako finální výsledek 2 přímky (*2line*), jinak 1 přímka (*1line*). Hodnota 25 představuje penalizaci rozdělení přímek, protože je v tomto případě snaha o co nejjednodušší aproximaci.

Na následujících Obrázcích x,y,z jsou znázorněné příklady/postup