

Západočeská univerzita v Plzni

KKY/ZDO

Semestrální práce

Pig leg surgery

Plzeň 2022

Lukáš Vladař, Lukáš Kölbl

Obsah

1	Návod ke spuštění programu	4
1.1	Načtení jednoho zipu	4
1.2	Kombinace výsledků	4
1.3	Zpracování videa	4
2	Popis programu	5
2.1	Zpracování trénovacích dat	5
2.2	Sledování nástroje	7
3	Závěr	9

Kapitola 1

Návod ke spuštění programu

V této části bude popsáno, jakým způsobem obsluhovat dané části programu a co je potřeba udělat pro jejich správné spuštění.

1.1 Načtení jednoho zipu

Jako první je nutné zajistit, aby daný se zip soubor nacházel ve stejné složce jako program samotný. Dále je nutné změnit proměnnou `cisloSouboru` na správné číslo zipu, který chceme zpracovávat. Poté stačí program `nacteni_jednoho_zipu.py` spustit. Výsledkem pak bude pickle soubor označený daným číslem zpracovaného zipu.

1.2 Kombinace výsledků

Do proměnné `"soubory"` je nutné zadat všechny čísla zip souborů, pro které budeme chtít pickle soubory zpracovávat. Všechny pickle soubory je nutné mít opět ve stejné složce jako program samotný. Výsledkem je pak finální pickle soubor obsahující data histogramu a pravděpodobností pohybu.

1.3 Zpracování videa

Je nutné změnit název proměnné `"videoName"` a mít video ve formátu `.mp4`. Výsledný pickle soubor je nutné mít opět ve stejné složce jako program samotný. Výsledkem je pak `.json` soubor s výslednými anotacemi. Je také vygenerováno video z daných framů s vyznačeným hrotem. Oba tyto soubory jsou uloženy do stejné složky jako je daný program.

Kapitola 2

Popis programu

2.1 Zpracování trénovacích dat

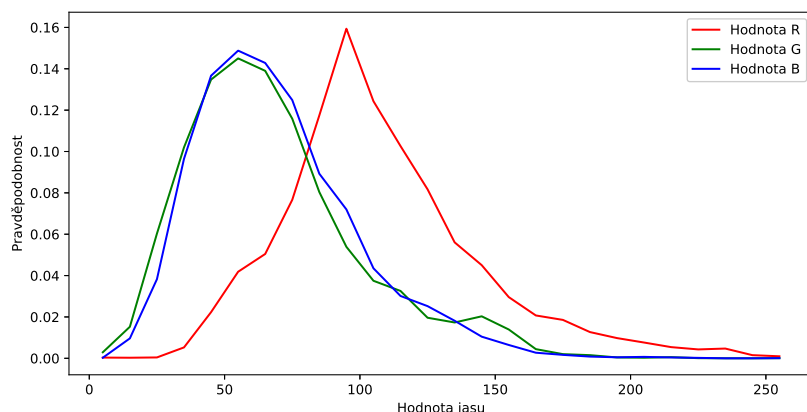
Pro natrénování algoritmu sledování chirurgických nástrojů máme k dispozici sadu ZIP archivů obsahujících anotovaná data. Archivy obsahují jednotlivé snímky videí uložené ve formátu PNG a dále anotace těchto dat ve formátu XML.

Program `nacteni_jednoho_zipu.py` načte jeden z těchto archivů, rozbalí jej a spočítá z dat určité charakteristiky.

Pro každý frame v trénovacím videu je vyříznut čtvereček o velikosti 20×20 px v okolí pixelu, který byl anotován jako hrot nástroje a z tohoto čtverce jsou počítány následující charakteristiky: Průměrné hodnoty R, G a B, průměrná hodnota výstupu Robertsova filtru a průměrná hodnota výstupu rozdílové metody analýzy pohybu (pro každý pixel obsažený v oblasti je uvažováno číslo 1, pokud se jasová hodnota tohoto pixelu oproti minulému framu změnila o více než o 0,1, v opačném čísle 0. Ze všech těchto hodnot je spočten průměr, a ten je vrácen jako výstup analýzy pohybu.).

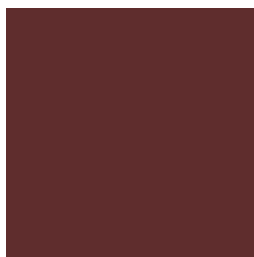
Celkem je tedy spočteno pět hodnot, které charakterizují hrot chirurgického nástroje. Tyto hodnoty jsou poskládány do jakéhosi příznakového vektoru. Dále je odhadována hustotní funkce daného „náhodného“ vektoru – celý prostor přípustných hodnot je rozdělen na malé oblasti a následně je spočten histogram udávající, kolik vektorů získaných z trénovacích dat leží v dané oblasti.

Z histogramu byly spočteny marginální pravděpodobnosti jednotlivých barevných složek, viz obrázek níže. Můžeme pozorovat, že nejvyšších hodnot u hrotu nástroje obvykle nabývá barevná složka R.



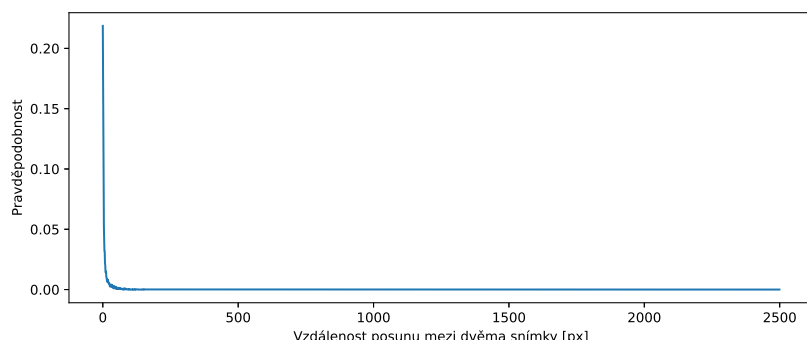
Obrázek 2.1: Marginální pravděpodobnosti pro barevné složky R, G a B

Můžeme dále zkontrolovat, jaká barva se v histogramu vyskytuje nejčastěji (tj. jakou barvu bude mít pravděpodobně čtvereček odpovídající hrotu nástroje „needle holder“ po zprůměrování). Jedná se o barvu s hodnotami přibližně $R = 95$, $G = 45$, $B = 45$, tj. o barvu vykreslenou níže. Takovou barvu zřejmě opravdu lze u hrotu nástroje očekávat.



Obrázek 2.2: Nejpravděpodobnější průměrná barva čtverečku obsahujícího hrot nástroje

Podobně je také zkonstruován histogram vzdáleností, o které se hrot nástroje posune mezi dvěma následujícími snímky. Histogram je vykreslen na obrázku níže. Můžeme pozorovat, že pravděpodobné jsou pouze velmi krátké pohyby nástroje, pro delší vzdálenosti se pravděpodobnost blíží nule.



Obrázek 2.3: Odhad pravděpodobnosti vzdálenosti posunu nástroje

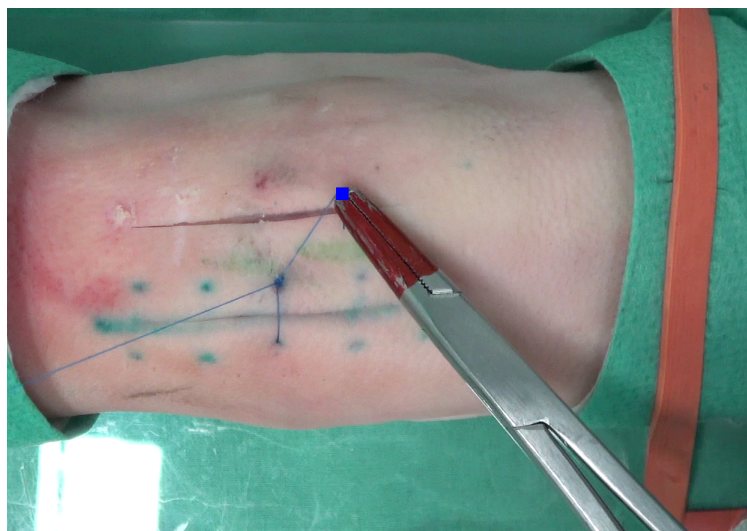
Tyto histogramy jsou vyhodnoceny pro každý ze ZIP archivů a jsou uloženy do souborů. Program `kombinace_vysledku.py` následně všechny tyto histogramy načte a spočítá z nich jeden histogram, který zachycuje rozdělení příznakových vektorů v celé trénovací sadě. Všechny hodnoty jsou dále vyděleny celkovým počtem uvažovaných vektorů, aby odpovídaly pravděpodobnostem. Dále jsou histogramy uloženy do souboru `vysledky.pckl`.

2.2 Sledování nástroje

Pro samotné sledování nástroje „needle holder“ byl vytvořen program `zpracovani_video.py`. Program nahraje video ve formátu MP4 a zároveň nahraje dříve získané histogramy. Následně je spuštěn Viterbiův algoritmus.

Každý snímek videa je rozdělen na čtverečky o velikosti 20×20 px. Pro každý z těchto čtverečků je (stejně jako pro čtvereček z anotovaných dat obsahující needle holder) spočten pětirozměrný příznakový vektor a pomocí histogramu je odhadnuta pravděpodobnost, že se jedná o hrot nástroje. Výpočet průměrné barvy čtverečku, průměrné hodnoty Robertsova filtru a průměrné hodnoty rozdílové analýzy pohybu je urychlen využíváním integrálního obrazu. Pro zrychlení celého algoritmu je uvažováno pouze 500 čtverečků, které mají tuto pravděpodobnost maximální. Cena každého čtverečku je spočtena jako součin pravděpodobnosti, že se jedná o hrot nástroje a součinu ceny minulého čtverečku a pravděpodobnosti přechodu z tohoto čtverečku na čtvereček aktuální, přičemž ze všech možných minulých čtverečků je vybrán ten, který má tuto cenu nejvyšší. V posledním framu je vybrán čtvereček s nejvyšší cenou, a poté je zpětným trasováním určena nejpravděpodobnější cesta nástroje v celém videu. Kvůli zamezení numerickým problémům bylo namísto pravděpodobností počítáno s jejich logaritmy.

Odhadnutá cesta nástroje je poté vykreslena do videa – nástroj je zde označen modrým čtvercem. Video je uloženo pod názvem `vystup.mp4`.



Obrázek 2.4: Ukázka označení hrotu nástroje ve výstupním videu

Na obrázku výše můžeme vidět označení hrotu nástroje ve výstupním videu. Dále je programem vytvořen soubor `vysledne_anotace.xml`, který obsahuje informace o tom, kde se hrot nástroje „needle holder“ pravděpodobně na každém snímku nachází.

Kapitola 3

Závěr

V práci bylo provedeno zpracování trénovacích dat na základě zvolených příznaků. Jejich aplikace pro nalezení hrotu nástroje a generování toho výstupu jako anotací a videa s vyznačeným hrotem v každém framu. Všechny dílčí části jsou dostupné v git repozitáři.

Ukázalo se, že algoritmus nefunguje zcela dle očekávání – nesleduje nástroj „needle holder“, ale značka označující hrot nástroje se zdánlivě chaoticky pohybuje po celé obrazovce. Přesto je možné si povšimnout, že algoritmus za hrot nástroje obvykle považuje části obrazu, kde se nachází hrana, často hrana reprezentující červený pás připevňující prasečí nohu k pracovnímu stolu a je nutné poznamenat, že tato část obrazu se hrotu nástroje velmi podobá.

Špatná funkčnost programu může být způsobena několika faktory – chybnou implementací (ačkoliv tato možnost nám nepřijde příliš pravděpodobná, neboť jak bylo uvedeno výše, program nachází v obrazu místa, která se hrotu nástroje alespoň podobají). Možné je rovněž to, že volba příznakového vektoru popisující části obrazu, není dostatečně vypovídající (uvažujeme pouze průměrnou hodnotu barvy apod.). Problém může být rovněž v provedeném zjednodušení, kdy algoritmus v každém snímku zkoumá pouze 500 nejpravděpodobnějších čtverečků, což má vést k jeho zrychlení. Problémem mohou být rovněž anotace dat (není jisté, jakým způsobem byla data přesně anotována – kam byl umístěn hrot nástroje v případě, kdy byl nástroj rozevřený apod.) a nakonec musíme zmínit také možnost, že zvolený algoritmus je pro řešení úlohy zcela nevhodný.

Ačkoliv program nesleduje hrot nástroje, podařilo se nám alespoň „přiblížit se“ hledanému řešení, tj. najít v každém snímku část obrazu, která se nástroji „needle holder“ podobá.