

Analýza signálu a obrazu v praxi I

RNDr. Zbyšek Posel, Ph.D.



Ústí nad Labem 2020

Kurz: Analýza signálu a obrazu v praxi I

Obor: Aplikovaná informatika

Klíčová slova:

Anotace: Kurz je v první části zaměřen na zopakování základních metod pro analýzu signálu v časové i frekvenční oblasti a dále na rozšíření znalostí o pokročilejší metody a vybrané metody filtrace. Tyto metody jsou poté aplikovány do oblasti zpracování fyziologických dat (EKG, EMG aj.). Druhá část předmětu je zaměřena na zpracování obrazu, kde jsou probrány základní metody a algoritmy pro analýzu obrazu.. Studenti v průběhu předmětu vypracovávají ve skupinách dvě seminární práce, které na závěr obhajují formou prezentace a validace vypracovaného programu v Pythonu. Využívány budou zejména knihovny jazyka Python, jako například NumPy, SciPy, Mahotas, OpenCV

Jazyková korektura nebyla provedena, za jazykovou stránku odpovídá autor.

© Katedra informatiky, PřF, UJEP v Ústí nad Labem, 2020

Autor: RNDr. Zbyšek posel, Ph.D.

Obsah

Úvodní slovo	4
1 Analýza signálu v časové oblasti	6
2 Analýza signálu ve frekvenční oblasti	9
3 Vlnková transformace	12
4 Digitální filtrace signálů	15
5 Počítačové zpracování fyziologických signálů	18
6 Filtrace fyziologických signálů	21
7 Počítačové metody analýzy obrazu	24
8 Digitální filtrace obrazových dat	27
9 Morfologické operátory	30
10 Segmentace obrazu	34

Úvodní slovo

Opora k předmětu *Analýza signálu a obrazu v praxi I* je tematicky rozdělena do dvou částí. V první části, která představuje většinu kurzu, jsou probrány jednotlivé disciplíny zpracování jednorozměrného signálu. První dvě kapitoly této části představují opakování základních metod a principů zpracování signálu v časové i frekvenční oblasti a jejich implementace v jazyku Python. Zpracování signálu v časově-frekvenční oblasti je rozšířeno ve třetí kapitole o vlnkovou transformaci. Další kapitoly obsahují přehled o digitálních filtrech v časové i frekvenční oblasti s aplikací například na filtraci řeči zaznamenané ve venkovních podmínkách. Zbylé dvě kapitoly, pátá a šestá, jsou zaměřeny na aplikaci výše nabytých znalostí na zpracování reálných fyziologických signálů, dostupných z veřejných databází. Všechny výše zmíněné kapitoly jsou doplněny o implementace v jazyku Python, které lze prostudovat po přihlášení do e-learningového systému MOODLE. Druhá část opory je zaměřena na základní metody zpracování obrazu. Jednotlivé kapitoly jsou zaměřeny tak, aby byly ke konci kurzu využity pro segmentaci obrazu pomocí Watershed algoritmu. První kapitola druhé části obsahuje popis reprezentace obrazových dat a jejich transformace. Implementace digitálních filtrů v prostorové i frekvenční oblasti je vysvětlena v další kapitole, kde jsou představeny i jednotlivé balíky jazyka Python. Morfologické operátory aplikované na binarizovaný snímek jsou stěžejní částí další kapitoly, společně s tvarovými deskriptory, pomocí kterých se charakterizují již segmentované buňky. Segmentace buněk v poslední kapitole pak obsahuje výše zmiňovaný watershed algoritmus, v rámci kterého dojde k propojení znalostí z dříve probraných kapitol. Princip watershed algoritmu je vysvětlen při zpracování snímků získaných z veřejně dostupných databází. I pro tuto část kurzu jsou kapitoly doplněny o implementace v jazyku Python, dostupné v e-learningovém systému MOODLE. V následující kapitole jsou uvedeny příklady seminárních prací zaměřených na zpracování signálu a obrazu z oblasti medicínských dat. Seminární práce představují ucelenou úlohu, v rámci které dojde k propojení znalostí z jednotlivých kapitol. Zadání práce jsou zaměřena tak, aby byly zpracovávána reálná data a aby implementované algoritmy zohledňovaly problémy a výstupy s tím související.

Příklad seminární práce

V průběhu kurzu budou zadány dvě seminární práce, na kterých budou moci studenti pracovat v rámci speciálně vyčleněných seminářů. V rámci plnění seminárních prací je možné vytvořit skupiny (nejvýše po 3 lidech), a odevzdat jednu práci za skupinu na konci semestru. Rozdělením do skupin jsou procvičovány dovednosti zaměřené na práci v týmu, sdílení znalostí, výstupů aj.

Zpracování biologického a fyziologického signálu

Pomocí Hilbertovy nebo vlnkové transformace proveďte detekci QRS komplexu v EKG signálu. Před samotnou detekcí proveďte filtraci signálu pro odstranění artefaktů způsobených například pohybem svalů. Zdrojové EKG záznamy si stáhněte z veřejně dostupné MIT-BIH Arrhythmia databáze (viz <https://physionet.org/content/mitdb/1.0.0/>). Svou detekci porovnejte s anotacemi, které jsou pro každý signál v databázi dostupné. Přesnost detekce prezentujte pomocí statistických ukazatelů (např. pozitivní a negativní prediktivita) a navrhněte úpravy Vašeho kódu pro zvýšení jeho efektivity.

Zpracování obrazu

Pomocí watershed algoritmu proveďte segmentaci buněk na snímku pořízeného mikroskopem. Jako zdroje dat buněk můžete využít například <http://www.cellimagelibrary.org/home>. Snímek nejdříve předzpracujte s využitím vhodné transformace barev. U segmentovaných buněk spočítejte základní tvarové ukazatele, jako je kulatost, elipsovitosť aj.

Výstupem seminární práce je přehledný protokol, ve kterém bude prezentováno

- zadání práce,
- rozdělení rolí v týmu, nastavení sdílení dat, nástroje pro týmovou práci aj.,
- metody využité pro analýzu signálu a obrazu,
- výsledky analýzy signálu,
- výsledky analýzy obrazu,
- shrnutí výsledků a návrh opatření pro zkvalitnění analýzy,
- zhodnocení práce v týmu, identifikované problémy a návrhy na jejich zlepšení.

1 Analýza signálu v časové oblasti



CÍLE KAPITOLY

Cílem této úvodní kapitoly je poskytnout přehled základních metod analýzy signálu v časové oblasti s důrazem na konvoluci, kovarianci a korelaci, které se budou dále využívat. Dále je cílem nastavení prostředí v jazyku Python, tak aby byly k dispozici všechny metody potřebné pro analýzu signálu. V této kapitole se dozvíte

- základní matematický popis signálu (amplituda, frekvence, fázový posun),
- poruchy a šum v signálu (náhodný, periodický, aj.) a metody pro posílení signálu v šumu,
- funkcionality jazyka Python, které budeme používat (balíky NumPy a SciPy),
- význam korelace, kovariance a konvoluce na praktických příkladech.



KLÍČOVÁ SLOVA

Matematický popis signálu, šum v signálu, korelace, kovariance, konvoluce.

Náplň látky je popsána v následujících zdrojích

- Základní informace o signálech a jejich matematickém popisu najdete například v kapitolách 1 a 2 této knihy [1].
- Náповědu k používání NumPy a SciPy balíků jazyka Python najdete zde [2] a zde [3].

praktické příklady korelace, konvoluce a kovariance lze dále nalézt v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Popsat jednoduchý signál matematickým modelem a vysvětlit význam jeho parametrů.
- Pracovat s Numpy poli a příslušnými metodami implementovanými ve SciPy balíku Pythonu.
- Provést (auto)korelaci signálu a zjistit tak jeho periodu.
- Provést posílení signálu v šumu pomocí technik průměrování.
- Provést kovarianci a konvoluci signálu a vysvětlit význam získaných koeficientů.

OTÁZKY

1. Jak vypadá auto-korelace periodického signálu tvaru sinová vlna, obdélník a pila?
2. Jaký je vliv počtu vzorků na přesnost technik pro posílení signálu v šumu?
3. Co je kovarianční matice?

OTÁZKY K ZAMYŠLENÍ

1. Je Vámi navržený způsob manipulace s daty optimální?
2. Bylo by možné Váš program škálovat na větší objemy dat?

ÚKOLY

1. Vytvořte skript pro generování periodického signálu s náhodnou složkou šumu. Uvažujte sinový, obdélníkový a pilový signál a volitelnou úroveň šumu.
2. Vytvořte skript s technikami pro posílení signálu v šumu. Proveďte analýzu vlivu počtu period na kvalitu získaného signálu.
3. Proveďte kovarianci a konvoluci signálu se šumem a bez šumu.

2 Analýza signálu ve frekvenční oblasti



CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled základních metod analýzy signálu ve frekvenční oblasti s důrazem na pochopení a aplikaci Fourierovy analýzy a její implementaci ve formě diskrétní Fourierovy transformace (DFT).

V této kapitole se dozvíte

- význam parametrů signálu ve vztahu k frekvenční oblasti (spektrum frekvencí),
- možnosti a limity Fourierovy transformace,
- princip počítačové implementace diskrétní Fourierovy transformace (DFT) na jednoduchý signál s pomocí vlastního programu a vestavěných SciPy funkcí,
- aplikace DFT na signály se šumem a základní princip filtrace.



KLÍČOVÁ SLOVA

frekvenční spektrum, Fourierova transformace, diskrétní Fourierova transformace, filtrace

Náplň látky je popsána v následujících zdrojích

- Teoretický základ k Fourierově transformaci lze získat například v 16. kapitole zde [4] a ve 28. kapitole zde [5]
- Diskrétní Fourierova transformace je popsána například ve 3. kapitole zde [1].
- Počítačová implementace DFT je popsána pomocí funkce `numpy.fft` zde [2] nebo pomocí `scipy.fft` zde [3].

Příklad implementace DFT a výše zmíněných funkcí najdete dále v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Popsat výhody a nevýhody DFT.
- Mít povědomí o implementaci DFT a ovládat implementaci pomocí SciPy funkcí.

- Interpretovat frekvenční spektrum jednoduchého signálu a signálu se šumem.

OTÁZKY

1. Jaké jsou výhody a nevýhody Fourierovy transformace?
2. Co vše nám řekne spektrum frekvencí o signálu?
3. Jak lze efektivně filtrovat šum se specifickou frekvencí?

OTÁZKY K ZAMYŠLENÍ

1. Je možné detekovat, kdy nastala událost s danou frekvencí v signálu?
2. Jaké jsou metody pro urychlení Vašeho programu pro DFT?

ÚKOLY

1. Napište skript, který provede Diskrétní Fourierovu transformaci jednoduchého signálu bez šumu a se šumem.
2. Implementujte SciPy funkci na stejný signál a porovnejte výstup.
3. Navrhněte jednoduchý postup pro filtraci specifické frekvence v signálu.

3 Vlnková transformace



CÍLE KAPITOLY

Cílem této kapitoly je seznámení s aplikací vlnkové transformace na periodické signály. V první části je kladen důraz na pochopení přechodu od Fourierovy transformace, přes okénkovou modifikaci až k vlnkové transformaci a škálogramu. Ve druhé části je kladen důraz na aplikaci diskrétní nebo spojité vlnkové transformace na jednoduché periodické signály různých tvarů a na signály bez šumu a se šumem.

V této kapitole se dozvíte

- jednotlivé modifikace Fourierovy transformace,
- princip vlnkové transformace,
- použití vlnkové transformace pomocí SciPy,
- vliv tvaru použité vlnky a signálu na podobu škálogramu.



KLÍČOVÁ SLOVA

vlnková transformace, škálogram, vlnkové koeficienty

Náplň látky je popsána v následujících zdrojích

- Přechod od Fourierovy transformace k vlnkové lze najít například zde [7].
- Popis škálogramu a diskrétní vlnkové transformace lze najít například zde [6].
- Popis SciPy funkce `scipy.signal.cwt` lze najít zde [3].
- Alternativu ke SciPy funkci s názvem `PyWavelets` lze najít zde [8].

Příklady implementace vlnkové transformace na jednoduchý periodický signál lze dále najít v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Popsat přechod od Fourierovy k vlnkové transformaci.
- Mít povědomí o tom, kdy se používá spojité a kdy diskrétní vlnková transformace.
- Mít schopnost provést diskrétní transformaci jednoduchého signálu pomocí SciPy.

- Umět interpretovat získaný škálogram.

OTÁZKY

1. Kdy se nejčastěji používá spojitá, a kdy diskrétní vlnková transformace?
2. Jaké vlnky se nejčastěji používají na periodické signály a proč?
3. Jaká je časová náročnost výpočtu diskrétní vlnkové transformace?

OTÁZKY K ZAMYŠLENÍ

1. Můžete určit, kdy naopak není vhodné nebo nutné vlnkovou transformaci používat?

ÚKOLY

1. Nageenerujte jednoduchý periodický signál a proveďte jeho vlnkovou transformaci.
2. Proveďte transformaci pomocí více druhů vlnek. Porovnejte a interpretujte získané škálogramy.
3. Výše uvedené proveďte pro periodické signály různých tvarů.
4. Porovnejte škálogramy signálů se šumem a bez šumu. Změňte i úroveň šumu a demonstруйте jeho vliv na podobu škálogramu.

4 Digitální filtrace signálů



CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled o digitálních filtrech využívaných před samotným zpracováním signálu a o oblastech jejich použití. Dále je cílem implementovat filtry v jazyku Python na vybrané signály s vysokofrekvenčním, nízkofrekvenčním nebo náhodným šumem.

V této kapitole se dozvíte

- rozdělení a typy digitálních filtrů,
- oblasti použití jednotlivých filtrů,
- implementace jednotlivých filtrů pomocí balíků Pythonu.



KLÍČOVÁ SLOVA

Kalmanův filtr, Wienerův filtr, FIR filtr, IIR filtr, adaptivní filtr

Náplň látky je popsána v následujících zdrojích

- Matematický popis filtrace signálu pomocí lineárních filtrů lze nalézt v 5. kapitole zde [1].
- Matematický popis filtrace signálu pomocí nelineárních filtrů lze nalézt v 10. kapitole zde [1].
- Matematický popis filtrace signálu pomocí adaptivních filtrů lze nalézt v 11. kapitole zde [1].
- Více se lze o filtraci signálů dozvědět například zde [9].
- Implementace filtrů v Pythonu pomocí SciPy (`scipy.signal`) lze nalézt zde [3].
- Implementaci Kalmanova filtru v Pythonu pomocí balíku `pyKalman` lze nalézt zde [10].

Implementace vybraných filtrů na signály s různou úrovní šumu lze najít dále v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Přehled o digitálních filtrech a jejich použití.
- Povědomí o tom, který filtr je vhodný pro filtraci vysoko nebo nízkofrekvenčního šumu
- Schopnost implementovat vybrané filtry na signály.
- Schopnost interpretovat vlivy jednotlivých parametrů filtru na kvalitu filtrace a zkreslení signálu

OTÁZKY

1. Proč je potřeba aplikovat digitální filtry?
2. Jaké jsou principy digitálních filtrů?
3. Jaké filtry použijete na sinový signál s adaptivním šumem?

OTÁZKY K ZAMYŠLENÍ

1. Jak budete postupovat v případě, kdy se frekvence šumu v průběhu zpracovávání mění?

ÚKOLY

1. Nagenerte jednoduché periodické signály s vysoko a nízkofrekvenčním šumem.
2. Navrhněte a aplikujte filtry pro výše uvedené signály
3. Proveďte filtraci záznamu nahrávky řeči ve venkovních podmínkách. Záznam lze získat například pomocí vlastního mobilního telefonu.
4. Na příkladu filtrace řeči demonstруйте aplikaci jednotlivých filtrů a jejich kvalitu.

5 Počítačové zpracování fyziologických signálů



CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled metod a jejich aplikací pro analýzu fyziologických signálů s cílem detekce vybraných událostí, které v signálu mohou nastat. Přehled metod bude zúžen na tři nejčastější diagnostické signály, a to elektrokardiogram, elektorencefalogram a elektromyogram. V neposlední řadě budou v této kapitole představeny veřejně dostupné repositoře fyziologických signálů.

V této kapitole se dozvíte

- co je to fyziologický signál a proč je vhodné ho počítačově analyzovat,
- jaké jsou nejčastěji měřené fyziologické signály a jak se měří,
- jaké informace a události jednotlivé signály nesou a jak se interpretují,
- jaké jsou metody počítačového zpracování fyziologických signálů,
- jaké jsou repositoře fyziologických signálů.



KLÍČOVÁ SLOVA

EKG, EEG, EMG, depolarizace, repolarizace, QRS komplex, akivační potenciál, Hilbertova transformace

Náplň látky této kapitoly je popsána v následujících zdrojích.

- Rozdělení a měření základních fyziologických signálů lze nalézt zde [12] a zde [13].
- Úvod do počítačového zpracování těchto signálů lze nalézt zde [11].
- Jeden z nejznámějších repositořů s fyziologickými signály lze nalézt zde [14].
- Implementaci nástrojů repositoře v Pythonu lze nalézt zde [15].
- Detaily ohledně Hilbertovy transformace lze nalézt v tomto článku [16].

Práci s fyziologickými signály a jejich anotacemi pomocí balíku wfdb [15] v Pythonu lze dále nalézt v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Mít povědomí o fyziologických signálech a jejich měření.
- Umět popsat jednotlivé signály a události, které v nich mohou nastat.
- Mít přehled o počítačových metodách pro zpracování fyziologických signálů.
- Umět získat fyziologický signál z veřejně dostupné databáze, který je opatřen anotacemi.

OTÁZKY

1. Jaké jsou události v EKG signálu, které se nejčastěji detekují?
2. Jaké jsou rozdíly v měření a interpretaci povrchového a jehlového EMG?
3. Jaké jsou počítačové metody pro analýzu a detekci událostí v EKG signálu?
4. Jaké jsou veřejně dostupné repozitáře fyziologických signálů?

OTÁZKY K ZAMYŠLENÍ

1. Jaké jsou problémy při detekci EKG signálů pomocí Hilbertovy transformace?
2. Co jsou to anotace signálu a k čemu jsou dobré?

ÚKOLY

1. Z veřejně dostupného repozitáře stáhněte EKG signál s anotacemi.
2. Načtený signál načtět do NumPy pole a vizualizujte jeho průběh.

6 Filtrace fyziologických signálů



CÍLE KAPITOLY

Cílem kapitoly je popsat externí a interní faktory, které ovlivňují kvalitu fyziologického signálu. Jedná se zejména o rušení způsobené pohybem svalů v těle, pohybem měřicích kabelů přístroje nebo rušením sítě. Dále je cílem kapitoly poskytnout přehled o vybraných digitálních filtrech, které se pro odstranění těchto artefaktů nejčastěji používají.

V této kapitole se dozvíte

- jaké jsou faktory ovlivňující přítomnost artefaktů v signálu,
- jaké digitální filtry se používají a jaké jsou možnosti jejich implementace.



KLÍČOVÁ SLOVA

artefakt, horní propust, dolní propust, pásmová propust, MA filtry

Náplň látky je popsána v následujících zdrojích.

- Přehled artefaktů vyskytujících se ve fyziologických signálech lze nalézt například v kapitole 3.2 zde [17], konkrétně pro EKG signál například ve 2. kapitole zde [18].
- Definice a použití filtrů v časové oblasti lze nalézt v kapitole 3.5 zde [17].
- Definice a použití filtrů ve frekvenční oblasti lze nalézt v kapitole 3.6 zde [17].
- Definice jednotlivých filtrů lze nalézt také v kapitolách 5 a 10-12, zde [1].
- Implementace filtrů v jazyku Python lze nalézt zde [19].

Příklady implementace filtrů na vybrané signály z veřejně dostupné databáze Physionet.org [14] lze dále nalézt v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Přehled o artefaktech, které se ve fyziologických signálech vyskytují.
- Přehled o filtrech, které se používají k jejich odstranění.
- Schopnost implementovat vybrané filtry na reálný fyziologický signál.

- Schopnost interpretovat vliv filtrů na tvar a kvalitu signálu.

OTÁZKY

1. Jaké artefakty se vyskytují ve fyziologických signálech?
2. Jaké digitální filtry se používají k jejich odstranění a jak ovlivňují filtrovaný signál?
3. Jakým způsobem lze implementovat tyto filtry pomocí Pythonu?

OTÁZKY K ZAMYŠLENÍ

1. Je možné digitální filtry použít při snímání signálu real-time?
2. Co je největším problémem digitálních při použití filtrů na fyziologické signály?

ÚKOLY

1. Z veřejně dostupné databáze stáhněte fyziologický signál.
2. Postupně aplikujte vybrané filtry a popište jejich vliv filtrů na kvalitu signálu.

7 Počítačové metody analýzy obrazu



CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled základních metod analýzy obrazu s důrazem na barevné modely obrazových dat a transformace mezi nimi.

V této kapitole se dozvíte

- jaké jsou základní metody pro analýzu obrazu nízké a vysoké úrovně,
- jaká jsou barevné modely, kterými se obrazová data reprezentují,
- jaké aplikace využívají konkrétní barevné modely,



KLÍČOVÁ SLOVA

barevné modely bodů, RGB, HSL, HSI, NTSC

Náplň látky je popsána v následujících zdrojích.

- Přehled základních principů počítačové grafiky včetně barevných modelů lze nalézt zde [21] nebo zde [22].
- Transformace mezi jednotlivými barevnými modely lze nalézt v kapitole 6 zde [24].
- Přehled základních metod analýzy obrazu vysoké a nízké úrovně lze nalézt např. ve 3. kapitole zde [20].

Příklady transformací mezi jednotlivými barevnými modely lze dále nalézt v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Mít povědomí o metodách analýzy obrazu.
- Mít povědomí o barevných modelech a aplikacích, které je využívají.
- Implementovat transformace mezi jednotlivými barevnými modely.



OTÁZKY

1. jaké jsou metody analýzy obrazu vysoké i nízké úrovně?
2. Jaké jsou barevné modely a jaké aplikace je využívají?
3. Pro jakou analýzu se jednotlivé barevné modely využívají?



OTÁZKY K ZAMYŠLENÍ

1. Jaké balíky jazyka Python obsahují metody analýzy obrazu?
2. Jaké jsou balíky jazyka Python pracující s barevným modelem?



ÚKOLY

1. Načtěte obrázek v modelu RGB pomocí jednoho z balíčků jazyka Python (např. OpenCV, Mahotas aj.).
2. Proveďte transformaci mezi RGB modelem a jinými vybranými modely. Výsledek vizualizujte.

8 Digitální filtrace obrazových dat



CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled o základních metodách digitální filtrace obrazových dat, která předchází analýze samotné. Dále jsou představeny metody filtrace v prostorové i frekvenční oblasti. V této kapitole se dozvíte

- jaké jsou digitální filtry používané v prostorové i frekvenční oblasti,
- kde se tyto filtry aplikují,
- které balíky jazyka Python mají filtry implementované.



KLÍČOVÁ SLOVA

normalizace histogramu pozadí, maska, Gaussův filtr, Laplaceův filtr, Fourierův filtr

Náplň látky je popsána v následujících zdrojích.

- Přehled digitálních filtrů v prostorové i frekvenční oblasti lze nalézt v kapitole 10 a 11 zde [23] nebo v kapitole 2 a 3 zde [24].
- Přehled funkcí vybraných balíčků jazyka Python pro zpracování obrazu scikit-image [25], OpenCV [26] a Mahotas [27].
- Aplikace jednotlivých filtrů (adaptivní, inverzní nebo Wienerův) lze nalézt například zde [24].

Aplikace a vliv vybraných filtrů na obrazová data je dále demonstrován na příkladech dostupných v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Přehled o digitálních filtrech používaných v prostorové i frekvenční oblasti.
- Povědomí o balíčcích jazyka Python, které mají filtry implementované.
- Implementovat jednoduché prostorové filtry (Gaussův filtr, normalizace histogramu pozadí aj.).

- Implementovat jednoduché filtry ve frekvenční oblasti (Fourierův filtr, horní nebo dolní propust').
- Interpretovat vliv jednotlivých filtrů na obrazová data.

OTÁZKY

1. Kdy a proč se implementují digitální filtry na obrazová data?
2. Znáte nějaké komerční aplikace, které mají filtry implementované?
3. Jaká filtrace se skrývá pod názvem *Blur* a *Unsharp mask*?

OTÁZKY K ZAMYŠLENÍ

1. Jak budete řešit filtraci obrazu na krajích snímku?

ÚKOLY

1. Načtěte obrazová data ve formátu RGB a převed'te je to modelu stupňů šedi.
2. Na transformovaný obraz aplikujte Gaussův filtr, uvažujte různě velké masky.
3. Na původní i transformovaný obraz aplikujte Fourierův filtr a porovnejte jejich spektra.
4. Postupným odstraněním vysokých a nízkých frekvencí demonstруйте jejich vliv na kvalitu obrazu.

9 Morfologické operátory



CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled o základních morfologických operátorech, jejich aplikace při analýze obrazu a implementaci v jazyku Python. Dále je cílem kapitoly poskytnout přehled o základních tvarových ukazatelích, kterými se charakterizují objekty nalezené v obrazových datech.

V této kapitole se dozvíte

- co je prahování obrazu a jaké známe metody,
- co jsou to morfologické operátory a kde se používají,
- jaké jsou základní tvarové ukazatele a jak se měří.



KLÍČOVÁ SLOVA

prahování histogramu, binarizace, morfologický operátor, dilatace, eroze, kulatost, perimetr, kompaktnost

Náplň látky je popsána v následujících zdrojích.

- Přehled metod binární matematické morfologie lze nalézt např. v kapitole 12 zde [23] nebo v kapitole 3 zde [20].
- Popis objektů pomocí tvarových ukazatelů lze nalézt např. v 11. kapitole zde [24] nebo v prezentaci zde [28].

Vliv jednotlivých morfologických operátorů na binarizovaný snímek je demonstrován na příkladech dostupných v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste měli mít následující dovednosti a znalosti.

- Povědomí o metodách prahování histogramu a binarizaci obrazových dat.
- Mít povědomí o základních morfologických operátorech.
- Implementovat tyto operátory v jazyku Python.
- Popsat vliv morfologických operátorů v závislosti na velikost použité masky.

- Implementovat tvarové ukazatele v jazyku Python.
- Interpretovat jednotlivé tvarové ukazatele a jejich vzájemný vztah.

OTÁZKY

1. Co je prahování histogramu a proč ho musím udělat?
2. Jaké jsou základní morfologické operátory?
3. Jaký je vliv velikosti masky na binarizovaná data?
4. Proč se u objektů měří tvarové ukazatele a kde se využívají?



OTÁZKY K ZAMYŠLENÍ

1. jaké jsou balíky jazyka Python, které obsahují morfologické operátory?
2. Ve kterých aplikacích nachází tyto operátory stále uplatnění?
3. Jaké kombinace tvarových ukazatelů jsou výhodné a které naopak nejsou?



ÚKOLY

1. Získejte histogram obrazových dat v modelu stupňů šedi.
2. Proveďte prahování a binarizaci dat. Vliv různých hodnot prahu vizualizujte.
3. Aplikujte morfologické operátory na binární snímek. Vizualizujte vliv velikosti masky.
4. Získejte základní tvarové ukazatele (kulatost, elispovitost, perimetr aj.) objektů před a po aplikaci morfologických operátorů.



MÍSTO PRO VAŠE POZNÁMKY

A series of horizontal dotted lines for taking notes.

10 Segmentace obrazu



CÍLE KAPITOLY

Cílem této kapitoly je poskytnout přehled o metodách segmentace obrazu buněk pořízených mikroskopy. Důraz je kladen na segmentaci pomocí watershed algoritmu, případně jeho modifikací. V neposlední řadě jsou také představeny veřejně dostupné databáze snímků buněk.

V této kapitole se dozvíte

- co je segmentace obrazu,
- co jsou regionální a lokální maxima,
- co je watershed algoritmus a kde se aplikuje,
- jak probíhá detekce hran objektů,
- jaké jsou veřejné databáze snímků buněk.



KLÍČOVÁ SLOVA

watershed algoritmus, regionální maxima, detekce hran

Náplň látky je popsána v následujících zdrojích.

- Princip segmentace obrazu lze nalézt například zde [29].
- Princip Watershed algoritmu lze nalézt například zde [31]
- Implementace watershed algoritmu v Open Source knihovnách lze nalézt například zde [30].

Aplikace watershed algoritmu pomocí některého z balíčků Pythonu [25],[26], [27], lze dále nalézt v e-learningovém systému MOODLE.



SHRNUTÍ

Po prostudování byste mít následující dovednosti a znalosti.

- Mít povědomí o oblastech použití segmentace obrazu.
- Mít povědomí o principu fungování watershed algoritmu.
- Znat (veřejně dostupné) zdroje dat snímků buněk pořízených mikroskopy.

- Implementovat watershed algoritmus.
- Detekovat hrany jednotlivých objektů a spočítat tvarové ukazatele jednotlivých buněk.

OTÁZKY

1. Jaké znáte veřejné repozitáře snímků buněk pořízených mikroskopy?
2. Jaké znáte balíky jazyka Python, které mají implementovaný watershed algoritmus?
3. Co jsou regionální a lokální maxima a proč je potřebujeme?
4. Jakým způsobem lze detekovat hrany objektů s využitím výstupů watershed algoritmu?

OTÁZKY K ZAMYŠLENÍ

1. Jaký je vliv Gaussova filtru na detekci objektů?
2. Jaký je vliv morfologických operátorů na detekci objektů?

ÚKOLY

1. Z veřejného repozitáře získajte snímek buněk pořízených mikroskopem např. ve viditelném spektru.
2. Zhodnoťte, zda je nutné provést normalizaci histogramu pozadí nebo převod do jiného barevného modelu.
3. Pomocí vybraného balíku jazyka Python, proveďte segmentaci obrazu pomocí watershed algoritmu.
4. Vizualizujte všechny kroky před segmentací obrazu, tedy aplikaci Gaussovy masky, binarizaci, lokální a regionální maxima.
5. Porovnejte úspěšnost segmentace v závislosti na velikosti Gaussovy masky nebo masky morfologických operátorů.

Literatura

- [1] JAN, Jiří. *Číslíková filtrace, analýza a restaurace signálů*. Vutium Brno, 2002. ISBN 80-214-1558-4.
- [2] NumPy [online]. NumPy developers: ©2020 [cit. 19.03.2020], Dostupné z: <https://numpy.org/>
- [3] SciPy [online]. SciPy developers: ©2020 [cit. 19.03.2020], Dostupné z: <https://www.scipy.org/>
- [4] REKTORYS, Karel a kol. *Přehled užité matematiky-svazek 1*. SNTL Praha, 1988. ISBN 04-022-88-01.
- [5] REKTORYS, Karel a kol. *Přehled užité matematiky-svazek 2*. SNTL Praha, 1988. ISBN 04-022-88-02.
- [6] ŠVEC, Martin. *Waveletové transformace*. UJEP, 2008. ISBN 978-80-7044-987-5.
- [7] HLAVÁČ, Václav. *Vlnková transformace* [online].[cit. 19.03.2020]. Dostupné z:<http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/14WaveletsCz.pdf>.
- [8] PyWavelets - Wavelet Transforms in Python [online]. The PyWavelets Developers: ©2006-2020 [cit. 19.03.2020], Dostupné z: <https://pywavelets.readthedocs.io/en/latest/>
- [9] SHENOI, B. A. *Introductin to Digital Signal Processing and Filter Design*. John Wiley&Sons, 2005. ISBN 0471656380.
- [10] PyKalman [online]. Daniel Duckworth: ©2012 [cit. 19.03.2020], Dostupné z: <https://pykalman.github.io/>
- [11] MOHYLOVÁ, Jitka; KRAJČA, Vladimír *Zpracování biologických signálů*. VŠB-TOU, 2006. ISBN 978-80-248-1491-9.
- [12] SVATOŠ, Josef *Biologické signály I: geneze, zpracování a analýza*. ČVUT Praha, 1998. ISBN 80-01-01822-9.
- [13] ROZMAN, Jiří a kol. *Elektronické přístroje v lékařství*. Academia, 2006. ISBN 80-200-1308-3.
- [14] PhysioNet - The Research Resource for Complex Physiologic Signals [online]. MIT Laboratory for Computational Physiology [cit. 19.03.2020], Dostupné z: <https://physionet.org/>
- [15] The WFDB Software Package [online]. [cit. 19.03.2020], Dostupné z: <https://archive.physionet.org/physiotools/wfdb.shtml>

- [16] BENITEZ, D. et al. The use of the Hilbert transform in ECG signal analysis. *Computers in Biology and Medicine*. 2001, **31**(5), 399-406. ISSN 0010-4825.
- [17] RANGAYYAN, Rangaraj M. *Biomedical signal analysis*. IEEE Press, 2015. ISBN 978-0-470-0-91139-6.
- [18] GACEK, Adam; PEDRYCZ, Witold. *ECG Signal Processing, Classification and Interpretation*. Springer, 2012. ISBN 978-0-85729-867-6.
- [19] UNPINGCO, José. *Python for Signal Processing*. Springer, 2014. ISBN 978-3-319-01341-1.
- [20] HRACH, Rudolf. *Počítačová fyzika*. UJEP, 2003. ISBN 80-7044-522-X
- [21] ŽÁRA, Jiří; BENEŠ, Bedřich; SOCHOR, Jiří, FELKEL, Petr. *Moderní počítačová grafika*. Computer Press, 2005. ISBN 80-251-0454-0.
- [22] SKÁLA, Václav. *Světlo, barvy a barevné systémy v počítačové grafice*. Academia, 1993, ISBN 80-200-0463-7.
- [23] HLAVÁČ, Václav; SEDLÁČEK, Miloš. *Zpracování signálů a obrazů*. ČVUT, 2007, ISBN 80-01-02114-9.
- [24] GONZALEZ, Rafael C.; WOODS, Richard E.; EDDINS, Steven L. *Digital Image Processing using Matlab*. McGraw Hill, 2010. ISBN 978-0-07-070262-2.
- [25] scikitimage-image processing in python [online]. The scikit-image development team: ©2020 [cit. 19.03.2020], Dostupné z: <https://scikit-image.org/>.
- [26] OpenCV [online], OpenCV team: ©2020 [cit. 19.03.2020], Dostupné z: <https://opencv.org/>
- [27] Mahotas: Computer Vision in Python [online], Luis Pedro Coelho: ©2008-2016 [cit. 19.03.2020], Dostupné z: <https://mahotas.readthedocs.io/en/latest/>
- [28] HLAVÁČ, Václav. *Object / 2D image descriptors* [online].[cit. 19.03.2020]. Dostupné z: <http://cmp.felk.cvut.cz/~drbohlav/TeachPres/51-2DImageDescription.pdf>.
- [29] DOUGHERY, Geoff. *Digital Image Processing for Medical Applications*. Cambridge University Press, 2009. ISBN 978-0-521-86085-7.
- [30] KORNILOV, Anton S. et al. An Overview of Watershed Algorithm Implementations in Open Source Libraries. *Journal of Imaging*. 2018, **4**(10), 123-138. ISSN 2313-433X.
- [31] BARNES, Richard et al. Priority-Flood: An Optimal Depression-Filling and Watershed-Labeling Algorithm for Digital Elevation Models. *Computers & Geosciences*. 2014, **62**, 117-127. ISSN 0098-3004.