VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019 Bc. Martin Berky

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

VYTVOŘENÍ BEZCHYBNÉ FOTOGRAFIE Z NARUŠENÉ VIDEOSEKVENCE
CLEAN PHOTO OUT OF CORRUPTED VIDEOSEQUENCE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Martin Berky**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Mgr. Pavel Rajmic, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Audio inženýrství**
Ústav telekomunikací

Student: Bc. Martin Berky ***ID:*** 162259
Ročník: 2 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Vytvoření bezchybné fotografie z narušené videosekvence

POKYNY PRO VYPRACOVÁNÍ:

Diplomová práce se bude věnovat separaci pohyblivých objektů od neměnného pozadí ve videosekvenci. Student se seznámí s běžnými metodami pro řešení tohoto problému a také s přístupem, který využívá tzv. řídké reprezentace. Po nastudování teorie bude navržen model a algoritmus, který dojde numerickým řešením k separaci pozadí od pohyblivé složky. Student algoritmus implementuje v Matlabu či Pythonu a ověří na simulovaném signálu. Posléze bude zpracováno reálné vlastní video, kde úkolem je vytvořit jeden snímek (tj. pozadí) z poslouposti zarušených snímků (tj. např. lidé chodící před fotografovanou budovou). Student porovná dosažené výsledky s běžnými metodami.

DOPORUČENÁ LITERATURA:

[1] Rajmic, P.; Daňková, M.: Úvod do řídkých reprezentací signálů a komprimovaného

snímání. Vysoké učení technické v Brně, 2014.

[2] Candes, E. J.; Wakin, M. B.: An introduction to compressive sampling. IEEE Signal Processing Magazine, ročník 25, č. 2, 2008: s. 21–30, ISSN 1053-5888.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 16.5.2019

Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato diplomová práce se zabývá separací pohyblivých objektů od statického, neměnného pozadí ve video sekvenci. V práci jsou popsány běžné metody separace a přístup využívající tzv. řídké reprezentace signálu. V praktické části práce byla vytvořena videa, na kterých je ověřován navržený algoritmus implementovaný v prostředí Matlab, pro získání pozadí z posloupnosti zarušených snímků a porovnání metod.

KLÍČOVÁ SLOVA

Řídká reprezentace signálů, komprimované snímání, medianova metoda, Matlab, video sekvence, separace pozadí, pohyblivé objekty, zpracování obrazu

ABSTRACT

This diploma thesis deals with separation of moving objects from static unchanging background in video sequence. In this thesis are described common method of separation and access using sparse signal representation. In the practical part of thesis was created video sequences, on which is verified the designed algorithm, implemented in Matlab, for obtaining background from damaged video frames and comparing this methods.

KEYWORDS

Sparse signal reprezentation, compressive sampling, median method, Matlab, video sequence, background substraction, moving objects, image processing

BERKY, Martin. *VYTVOŘENÍ BEZCHYBNÉ FOTOGRAFIE Z NARUŠENÉ VIDEOSEKVENCE*. Brno, 2019, 66 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „VYTVOŘENÍ BEZCHYBNÉ FOTOGRAFIE Z NARUŠENÉ VIDEOSEKVENCE“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Mgr. Palvu Rajmicovi Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Tato diplomová práce se zabývá problematikou separace pozadí z narušené statické video sekvence. V dnešní době rozvoje chytrých technologií a převážně chytrých telefonů, kdy se jednotlivý výrobci předhání ve vývoji nejlepšího fotoaparátu na trhu, ať už co do rozlišení, snímače nebo umělé inteligence. Pořízení fotografie je možné během několika vteřin, stejně tak i videa a jeho okamžité sdílení s ostatními uživateli. Jistě každý z nás již někdy zažil situaci, kdy si chtěl vyfotografovat zajímavou budovu na dovolené nebo výletě a před tímto fotografovaným objektem prošli náhodní chodci nebo projel automobil, či jiný rušivý element, který snímek foceného pozadí znehodnotil. A přesně tato, zdánlivě neřešitelná situace, je motivací pro vznik tématu této diplomové práce a vytvoření aplikace pro separaci.

Již v roce 2013 přišla na trh tchajwanská firma HTC s novým chytrým telefonem s označením One, s novým převratným fotoaparátem s názvem Zoe. Předností nové technologie fotoaparátu byl především nový typ senzoru s větší plochou světločivných buněk. Hlavní odlišnou funkcí fotoaparátu ovšem byl fakt, že pořizování videa a fotek nezačínalo ve chvíli stisknutí spouště, ale chvíli před ní. Což na jednu stranu znamenalo větší náročnost na paměť a spotřebu energie, avšak na straně druhé byly velice zajímavé funkce práce s fotografií/video sekvencí. Díky faktu zaznamenání většího počtu snímků (celkově asi 20 snímků ve 3 vteřinové video sekvenci) bylo možné při fotografii více počtu lidí vybrat u konkrétního obličeje povedený úsměv bez mrknutí nebo také při sekvenci snímků akčního záběru vložit do jedné fotografie pohybovaný objekt v čase, ale především potom funkci mazání objektů. Při focení pozadí s rušivými objekty na snímku telefon přímo nabídl vymazání některých nechtěných objektů. Jedná se o stejnou problematiku jako v této práci, s tím rozdílem, že v chytrých telefonech HTC byla funkce implementována v reálném čase.

Z výše uvedeného příkladu vyplývá, že se jedná o získání neměnného pozadí ze zarušeného statického video snímku, vlivem pohybujících se rušivých elementů, které bychom mohli nazvat popředím. Zkušenější fotografové, či grafici jistě znají funkci statistika v programu Photoshop od firmy Adobe, která je doposud hojně využívaná, za pomoci které lze nezarušené pozadí získat. Tato mediánová metoda bude blíže rozebrána v první kapitole.

V úvodní kapitole jsou popsány teoretické základy videosignálů, krátký vhled do historie digitálního videa, popsány základní parametry video signálů. Je zde popsáno digitální zpracování obrazu, jelikož toto téma úzce souvisí se zpracováním videa, které si jednoduše představujeme jako rychlou sekvenci snímků. V další části jsou popsány prvky zrakového vnímání a naznačená struktura lidského oka, jelikož těchto znalostí se využívá, ať už při konstrukci digitálních video kamer, či fotoaparátu, tak například při komprimaci nebo vhodném nastavení základních parametrů

video sekvence. Následně je krátce popsána kolorimetrie, barva a barevné prostory, které navazují na fyziologickou podstatu oka a lidského vnímání, což je v praxi také využíváno. Je zde detailněji rozebrána úloha oddělení pohybujících se objektů od pozadí a popsána výše zmiňovaná mediánová metoda, která je demonstrovaná na vlastních video snímcích a primárně ověřena za pomoci mediánové statistiky, v programu Adobe Photoshop. Je zde popsáno oddělení pozadí ze sekvence zarušených snímků využitím tohoto softwaru.

Další část diplomové práce popisuje řídkost a nízko hodnostní reprezentaci, protože právě řídká reprezentace signálů je v posledních desítkách let, až do současnosti, velkým a stále se rozvíjejícím trendem pro zpracování video dat, pro různé aplikace. Popsáno je také komprimované snímání, které nabourává klasickou strukturu a souvisí s řídkostí. Jsou zde také popsány proximální metody řešení.

Třetí kapitola se věnuje již praktické části, kde je popsána metoda oddělení pozadí pomocí mediánu v programu Matlab. Jejímž výsledkem je shodná funkce s funkcí statistika v programu Adobe PS, představené v úvodní kapitole. Je zde také krátká sekce věnovaná pořízení testovacích video sekvencí.

Předposlední část se věnuje konkrétní implementaci návrhu řešení pomocí řídkých reprezentací. V rámci této kapitoly je proveden i experiment v programu Matlab a na konkrétních získaných video datech provedeno ověření těchto metod.

Poslední část je věnována porovnání a vyhodnocení dvou zpracovaných metod, na ni navazuje závěr.

1 VIDEOSIGNÁLY A JEJICH ZPRACOVÁNÍ

Zjednodušeně si lze video představit jako rychlé střídání jednoho snímku za druhým, čímž vyvolává pro lidský mozek iluzi plynulého pohybu. Historický vývoj video signálů je tedy úzce spjatý s vývojem a zpracováním obrazu samotného. Video by tedy šlo nazvat „obrazovou sekvencí“, protože časově proměnný obraz je reprezentován časovou posloupností statických snímků/obrázků. Před vývojem digitálního záznamu a zpracování videa resp. obrazu bylo tradičně zaznamenáváno, ukládáno a přenášeno v analogové podobě. [1]

1.1 Historie digitálního videa

Historie digitální kinematografie se datuje ke konci osmdesátých let minulého století, kdy registrujeme první snahu firmy Sony. Úplně prvním digitálním snímkem na světě byl film Rainbow z roku 1996, který byl jako první pořízený a zpracovaný digitálně. Nicméně za první celovečerní film pro komerční účely, který byl výhradně natočen a zpracován na digitálních zařízeních, je považován film Broadcast z roku 1998. Tento rok také souvisí s první výrobou rekordérů a profesionálních kamer HDCAM s rozlišením 1920x1080 pixelů, založených na technologii CCD. [8]

1.2 Parametry videa

Jelikož video, jako sekvence snímků jdoucích za sebou, je mnohem náročnější na paměť zařízení, bylo nutné data komprimovat. Cílem komprese dat je tedy snížení jejich objemu, měřitelného v bitech. Kodek (angl. codec) je softwarový nebo hadrwarový prvek, který zakóduje resp. dekóduje datový proud. Parametry digitálního obrazu jsou výška, šířka a barevná hloubka. Výška a šířka se obecně označuje jako rozlišení, což platí i pro video. Dalšími důležitými parametry digitálního videa je především snímková frekvence (frame rate), která nám udává počet snímků za vteřinu. A jak již bylo napsáno, datový tok nám určuje počet bitů za jednu vteřinu. Video kodek je pak kodek videa, popsaný výše, který ukládá video tok do tzv. multimediálního kontejneru, díky kterému je možná kombinace různých multimediálních datových souborů, jako například audio, video nebo titulky, do jednoho souboru. Častou chybou je zaměňování pojmu videokodeku s formátem videa. Formátem je konkrétně míněn standard/specifikace, na rozdíl od kodeku, který je jeho konkrétní implementací. [10]

Dalším parametrem souvisejícím se snímkovou rychlostí jsou užívané standardy. PAL standard, používaný zejména v Evropě, Asii a Austrálii, stejně jako SECAM,

užívaný ve Francii, Rusku a části Afriky, specifikuje 25 snímků za vteřinu. NTSC standard, užívaný v USA, Kanadě a Japonsku, je specifikován jako 29,97 snímků za vteřinu. Nicméně užití těchto standardů s příchodem digitálního záznamu a zpracování není z geografického hlediska tak striktní. Naopak s vývojem nejmodernějších technologií je tendence vytvořit záznam v co největším možném objemu dat a posléze jej vhodně zpracovat, podle jeho využití. Nicméně u snímkové frekvence neplatí pravidlo čím více, tím lépe. Aby měl lidský mozek pocit, že je obraz plynulý, stačí snímková frekvence asi 16 snímků za vteřinu. Vyšší snímková frekvence, řádově 120 nebo 240 i více, se používá především pro zpomalování video záběrů. Jedním z posledních parametrů, která si uživatel video kamery nebo digitální zrcadlovky volí je výběr mezi interlaced (prokládaný) a progressive (progresivní). Rozdíl mezi těmito dvěma parametry je, že progresivní skenování aktualizuje pro každou novou skenovací dobu všechny řádky v každém snímku v sekvenci. Prokládaní bylo vynalezeno jako způsob, jak omezit blikání v mechanických a CRT video displejích, bez zvýšení počtu kompletních snímků za vteřinu.

Pokud bychom měli dvě videa, jedno s rozlišením 640x480 pixelů a druhé 800x600, jejich poměr stran (angl. aspect ratio) je 4:3. Tento poměr je typický pro použití neširokoúhlého obrazu. Druhým hojně využívaným širokoúhlejším poměrem je 16:9, což je dáno tím, že pixel nemusí být vždy čtvercový.

Datový tok určuje množství dat přenesených za jednu sekundu video záznamu, podílí se tedy na výsledné velikosti videa, jeho jednotkou je bps (počet bitů za vteřinu). Stejně jako u snímkové frekvence, výsledná kvalita nezáleží primárně na co nejvyšším datovém toku, ale především na použitém video kodeku. Používají se dva druhy datových toků. Prvním z nich je konstantní datový tok CBR, který má pevně danou hodnotu a tudíž se během videa nemění. Naproti tomu variabilní datový tok VBR může velmi ovlivnit výslednou velikost videa, jelikož jeho velikost se může měnit v závislosti na dynamice scény a změn v obrazu. Kdy u dynamických scén a akčních záběrů je datový tok vyšší, něž-li u statických scén a záběrů, kde se obraz příliš nemění, čímž lze dosáhnout podstatně menší velikosti, ale zachování výsledné kvality u dynamických scén.

Pro posouzení video kvality slouží objektivní hodnocení, jako například parametr poměr špičkového signálu k šumu PSNR nebo subjektivní hodnocení pomocí expertního pozorování. Mnoho těchto subjektivních metod je popsáno v doporučené ITU-R. [11, 13, 14]

1.3 Digitální zpracování obrazu

Obraz může být definován jako dvourozměrná funkce $f(x, y)$, kde x a y jsou prostorové (rovinné) souřadnice a amplituda f v každém páru souřadnic se nazývá

intenzita nebo šedá úroveň obrazu v bodě. Když x,y a hodnota amplitudy f jsou všechny konečné, diskrétní veličiny, nazýváme obraz digitálním obrazem. Digitální obraz se tedy skládá z konečného počtu prvků, z nichž každý má konkrétní umístění a hodnotu. Tyto prvky se označují jako obrazové prvky nebo pixely. Pixel je nejpoužívanější termín pro označení prvků digitálního obrazu.

1.3.1 Vzorkování a kvantizace obrazu

Proces vzorkování odvozuje diskrétní sadu datových bodů v (obvykle) jednotných vzdálenostech. Ve své nejjednodušší podobě je vzorkování vyjádřeno matematicky, jako násobení původního obrazu funkcí, která „měří“ jas obrazu v diskrétním umístění nekonečné šířky/plochy/objemu v 1-D/2-D/3-D případech.

$$f_s[n \cdot \Delta x] = f[x] \cdot s[x; n \cdot \Delta x], \tag{1.1}$$

kde $f[x]$ je rozložení jasu vstupního obrazu, $s[x; n \cdot \Delta x]$ funkce vzorkování a $f_s[n \cdot \Delta x]$ je vzorkovaný vstupní obraz, definovaný v souřadnicích $n \cdot \Delta x$.

Pro (poněkud méně přísný) účel můžeme uvažovat vzorkovací funkci jako „uchycení“ hodnoty funkce spojité vstupní funkce $f[x, y]$ na specifická místa oddělená jednotnými intervaly Δx a Δy, kde $\Delta x = \Delta y$:

$$f_s[n, m] = f[n \cdot \Delta x, m \cdot \Delta y]. \tag{1.2}$$

Jinými slovy, pod koberec se zametají některé detaily.

Proces kvantování (A/D konverze) převádí spojitou hodnotu energie („světlosti“ nebo „jasu“) měřený ve vzorku (nebo odvozeném signálu, jako je napětí měřeného signálu) k jednomu z diskrétních souborů šedých úrovní (někdy zvaný jako digitální počet, i když jeho zkratka „DC“ může být zaměňována se zkratkou pro stejnosměrný proud). Například měřený signál ve vzorku umístěném v $[x0, y0]$ může být $f[x0, y0] = 1,234567890 \cdots \frac{W}{mm^2}$. Toto číslo je převedeno na nějakou celočíselnou hodnotu, třeba $f_q = 42$. Rozsah povolených celých čísel je určen různými faktory v kvantizeru. Například bychom mohli mít $0 \leq f_q \leq (f_g)_{max}$, kde maximální hodnota $(f_g)_{max}$ je dána počtem bitů v kvantizeru: $(f_g)_{max} = 2^8 - 1 = 255$ pro 8-bitový kvantizer a $(f_g)_{max} = 2^{12} - 1 = 4095$ pro (stále častější) 12-bitový kvantizer ve vědeckých digitálních kamerách. [12]

Výsledkem reprezentace digitálního obrazu, po kvantování a vzorkování, je tedy matice reálných čísel o velikosti m řádků a n sloupců. [9]

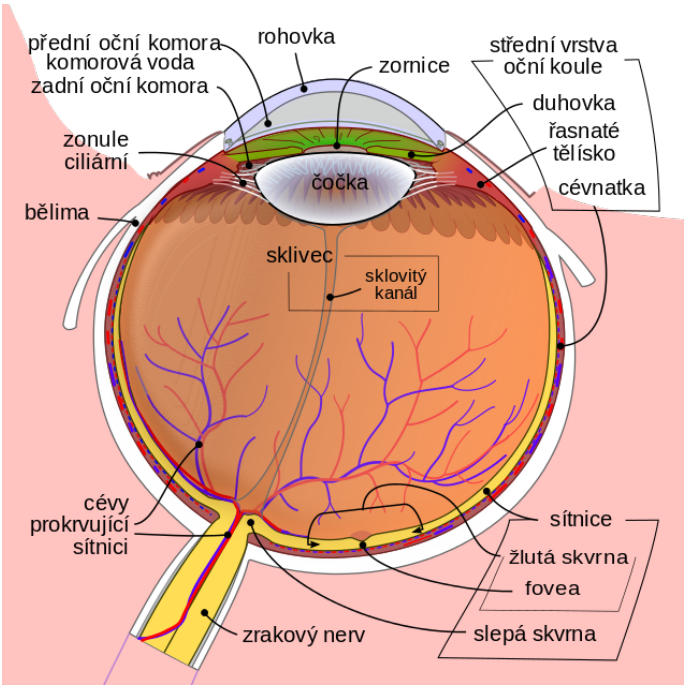
1.4 Prvky zrakového vnímání

Ačkoliv oblast digitálního zpracování obrazu je postavena na základech matematických formulací, lidská intuice a analýza hrají ústřední roli při výběru jedné techniky

oproti druhé, jelikož volba je často založena na subjektivním vnímání. Proto je vhodné rozvíjení chápání lidského vizuálního vnímání, alespoň v základním pokrytí jednotlivých aspektů, zejména v mechanice a parametrech souvisejících s utvářením obrazu v oku. Pochopení fyzického omezení lidského vidění a porovnání lidského a elektronického zobrazování, z hlediska rozlišení a schopnosti přizpůsobení změnám v osvětlení, jsou nejen zajímavé, ale také důležité z praktického hlediska. [9]

1.4.1 Struktura lidského oka

Obrázek 1.1 ukazuje zjednodušený horizontální průřez lidského oka. Oko je téměř koule s průměrem asi 20 mm. Tři membrány obklopují oko: Rohovka a bělmo vnější kryt, cévnatka a sítnice vnitřní. Rohovka je tuhá, průhledná tkáň, pokrývající přední povrch oka. Spojitě rohovka s bělmem je neprůhledná membrána, která uzavírá zby-



Obr. 1.1: Průřez lidského oka (převzato z [15]).

tek optického glóbu. Cévnatka leží přímo pod bělmem, jako membrána obsahuje síť krevních cév, které slouží oku jako hlavní zdroj výživy. Dokonce i drobné poranění cévnatky, často mylně nepovažované za vážné, může často vést k vážnému poškození očí, v důsledku zánětu, který omezuje průtok krve. Povrch cévnatky je silně pigmentovaný, a proto pomáhá snížit množství vnějšího světla vstupujícího do oka a zpětného rozptylu uvnitř oční bulvy. Na vnější straně je cévnatka rozdělena na řasnaté tělísko a duhovku, která se uzavírá nebo rozšiřuje se vstupujícím množstvím světla do oka. Centrální otvor duhovky (zornice) se mění v průměru od přibližně 2

do 8 mm. Přední strana duhovky obsahuje viditelný pigment oka, na rozdíl od zadní části, která obsahuje pigment černý.

Čočka je tvořena soustředěnými vrstvami vláknitých buněk a je zavěšena vlákny, které se připojují k řasnatému tělísku. Obsahuje 60 až 70% vody, asi 6% tuku a více bílkovin, než kterákoli jiná tkáň v oku. Čočka je zbarvená mírně žlutou pigmentací, která se s věkem zvyšuje. V extrémních případech může docházet, vlivem vychýlení čočky, k šedému zákalu a ztrátě jasného vidění. Čočka absorbuje přibližně 8% spektra viditelného světla. Infračervené a ultrafialové záření je pohlcováno značně proteiny ve struktuře oka a v nadměrné míře jej mohou poškodit. Nejvnitřnější membránou oka je sítnice, která lemuje vnitřek celé zadní části stěny oka. Jestliže je oko správně zaostřeno, světlo objektu vně oka je zobrazeno na sítnici. Obrazové vidění je zajištěno distribucí diskrétních světelných receptorů po povrchu sítnice.

Existují dva druhy receptorů: čípky a tyčinky. Čípků je v každém oku mezi 6 a 7 miliony. Jsou primárně lokalizovány v centrální části sítnice, zvané jamka (fovea), a jsou vysoce citlivé na barvu, díky nim je možné rozlišit jemné detaily, proto každý z nich je spojen s vlastním nervovým zakončením. Svaly ovládající oko, otáčí oční bulvou až poté, co spadá obraz, či předmět na jamku. Vidění čípků se nazývá fotopické. Počet tyčinek je mnohem větší, přibližně 75 až 150 milionů a jsou rozloženy po povrchu sítnice. Větší distribuční oblast, a také fakt, že několik tyčinek je spojeno s jedním nervovým zakončením, snižuje množství detailů rozpoznaných těmito receptory. Tyčinky slouží k získání hlavního, celkového obrazu, zorného pole, nejsou zapojeny do barevného vidění a jsou citlivé na nízké úrovně osvětlení. Například objekty jevící se jako jasně zbarvené za denního světla, jsou viděny za nočního světla jako bezbarvé, protože jsou stimulovány pouze tyčinky. Tento jev je známý jako skotopický.

Samotná oční jamka je kruhovým zářezem v sítnici asi 1,5 mm v průměru. Avšak z hlediska dalších interpretací by bylo užitečnější mluvit o čtvercových nebo obdelníkových. Při určité svobodě interpretace, můžeme vidět jamku jako čtvercové pole senzorů o velikosti 1,5 mm x 1,5 mm. Hustota čípků v této oblasti sítnice je přibližně 150 000 prvků na mm^2. Na základě této aproximace je počet čípků v oblasti nejvyšší ostrosti oka asi 337 000 prvků. Co se týče snímací schopnosti CCD zobrazovacího čipu se středním rozlišením, může mít počet prvků v poli receptorů ne větším než 5 mm x 5 mm. Ačkoliv schopnost člověka integrovat inteligenci a zkušenosti s vidění, činí tento typ porovnání nebezpečným. [9]

1.4.2 Vytváření obrazové informace v oku

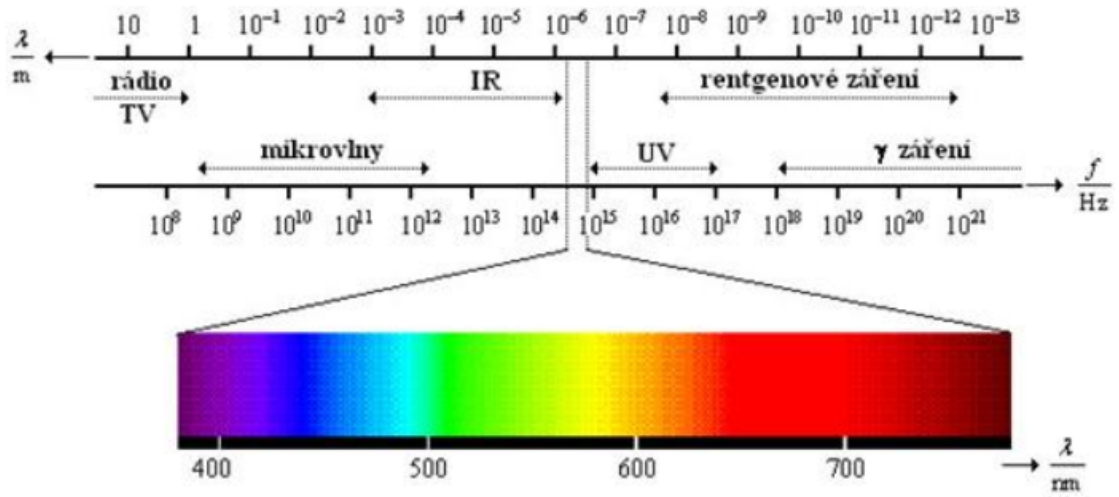
Hlavní rozdíl mezi čočkou oka a obyčejnou čočkou je ten, že oční je flexibilní. Poloměr zakřivení předního povrchu čočky je řízen pnutím ve vláknech řasnatého tělíska.

Pro zaměření vzdálených objektů, zplošťují svaly čočku. Obdobně při zaostřování blízkých objektů svaly umožňují čočce být tlustší. Vzdálenost mezi středem čočky a sítnicí (nazývaná jako ohnisková vzdálenost) se pohybuje přibližně od 17 mm do 14 mm, jako refrakční výkon čočky, zvyšující se z minima na maximum. Když oko zaostří na objekt vzdálenější než asi 3 m, čočka vykazuje nejnižší refrakční výkon. Jak bylo napsáno výše, sítnicový obraz je primárně zobrazován v oblasti oční jamky. Vnímání pak probíhá relativní stimulací světelných receptorů, které transformují energii záření na elektrické impulsy, které jsou nakonec dekódovány samotným mozkem. [9]

1.5 Světlo, kolorimetrie

Z fyzikální podstaty je světlo typ elektromagnetického vlnění, které může být viděno okem, jako šíření sinusových vln různých vlnových délek. Může být považováno za proud nehmotných částic, z nichž každá putuje vlnovým prostorem rychlostí světla. Každá částice této energie se nazývá foton.

Již v roce 1666, Sir Isaac Newton objevil, že když paprsek slunečního světla prochází skleněným hranolem, objevující se paprsek světla není bílý, ale sestává ze spojitého spektra barev, od fialové až po červenou. Jak je znázorněno na obr. 1.2,



Obr. 1.2: Elektromagnetické spektrum s přiblíženým viditelným spektrem (převzato z [16]).

rozsah barev, které vnímáme ve viditelném spektru, představuje velice malou část elektromagnetického spektra. Na jednom konci spektra jsou rádiové vlny s vlnovými

délkami řádově 10^9 krát delší než viditelné světlo, naopak na druhém konci spektra jsou gama paprsky s vlnovými délkami řádově 10^{-6} krát menšími než viditelné spektrum.

Elektromagnetické spektrum může být vyjádřeno podmínkami vlnové délky, frekvence nebo energie. Vlnová délka λ a frekvence f jsou vztaženy k:

$$\lambda = \frac{c}{f}. \tag{1.3}$$

kde c je rychlost světla ($2,988 \times 10^8 \frac{m}{s}$). Vlnová délka je vzdálenost, kterou vlna urazí v daném prostředí za jednu periodu.

Pro zjednodušení se barevné spektrum dělí na šest širších oblastí: fialová, modrá, zelená, žlutá, oranžová a červená. Barvy, které lidi vnímají na objektech, jsou určeny povahou světla odraženého od objektu. Barva je tedy psychofyzikálním jevem. Tělo, které odráží světlo a je relativně vyvážené ve všech viditelných vlnových délkách se jeví pozorovateli jako bílé. Nicméně tělo, které umožňuje odrazivost v omezeném rozsahu viditelného spektra, vykazuje některé odstíny barev. Například zelené objekty, odrážející světlo s vlnovými délkami primárně v rozsahu 500 až 570 nm, absorbují většinu energie na jiných vlnových délkách. [9]

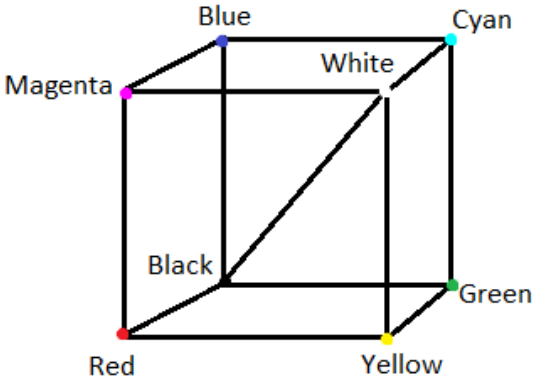
1.6 Barevné modely

Barevný prostor je matematická reprezentace množiny barev. Tři nejpopulárnější barevné modely jsou RGB (užívaný v počítačové grafice) YIQ/YUV nebo YCbCr (užívaný ve video systémech) a barevný model CMYK (užívaný především pro barevný tisk). Nicméně žádný z těchto barevných prostorů není přímo spojení s intuitivním představením odstínu, saturaci a jasu. To vedlo k dočasnému sledování jiných modelů, jako jsou HSI a HSV, za účelem zjednodušení programování, zpracování a manipulací koncovými uživateli. [2]

1.6.1 RGB

Červená, zelená a modrá jsou tři primární, aditivní barvy (jednotlivé složky se sčítají, pro vytvoření požadované barvy) a jsou reprezentovány trojrozměrným kartézským souřadnicovým systémem, jak je vidět na obr. 1.3. Uvedená úhlopříčka krychle, se stejným množstvím každé primární složky, představuje různé stupně šedi. Barevný prostor RGB je nejrozšířenější volbou pro počítačovou grafiku, protože displeje používají červenou, zelenou a modrou pro vytvoření požadované barvy. Poněvadž barevný model RGB existuje již několik let, je možné využít množství stávajících softwarových postupů, zjednodušuje architekturu a design systému. Nicméně RGB není

příliš efektivní při práci s obrazy v reálném světe, protože všechny tři RGB komponenty musí mít stejnou šířku pásma pro generování jakékoliv barvy uvnitř RGB krychle. Výsledkem je rámcová vyrovnávací paměť, která má stejnou hloubku pixelů a rozlišení zobrazení pro každou komponentu RGB, co není obvykle nejúčinnější metoda. Pokud bychom chtěli změnit intenzitu nebo barvu daného pixelu, musí být z vyrovnávací paměti načteny tři hodnoty, vypočtena intenzita nebo barva, provedené modifikace zpět zapsány do vyrovnávací paměti rámce. Pokud by systém přistupoval k obrazu uloženému přímo ve formátu intenzity a barev, byly by některé kroky zpracování rychlejší. Pro každou složku je vyhrazeno přesně 8 bitů, celkem tedy 24



Obr. 1.3: Barevný RGB model v kartézské soustavě souřadnic

.

bitů na pixel. Na jednotlivou složku připadá $2^8 = 256$ úrovní/kvantovacích hladin. Celkový počet barev tedy odpovídá $256^3 = 16777216$. [2, 10]

1.6.2 Šedotónový obraz

Jak již bylo v předchozí kapitole naznačeno, je možné redukovat barevnou informaci ve stupních šedi. Vychází se ze skutečnosti, že šedý odstín je součtem složek R, G a B. K dispozici máme 256 stupňů šedi při 8-bitovém uložení v paměti, pro samotnou reprezentaci postačí již pouze jeden kanál a ne tři. [17]

Převod z barevných modelů do stupňů šedi není vždy tak jednoznačným úkonem. Konkrétně se jedná o to, že z vícerozměrných dat je nutné získat pouze jednorozměrnou veličinu, většinou nazývanou jako jas. Pojmem desaturace je myšleno určení jasu barvy jako průměrné hodnoty složek r,g, b. Následující vzoreček řeší přepočet, se zohledněním různé míry citlivosti oka na různé barvy, a podle toho je barevným

(a) (b)

Obr. 1.4: Obraz v barevném modelu RGB (a) a ve stupních šedi (b).

kanálům přiřazena různá váha

$$Y(r,g,b) = 0,299r + 0,587g + 0,114b \quad nebo \tag{1.4}$$

$$Y(r,g,b) = 0,2126r + 0,7152g + 0,0722b,. \tag{1.5}$$

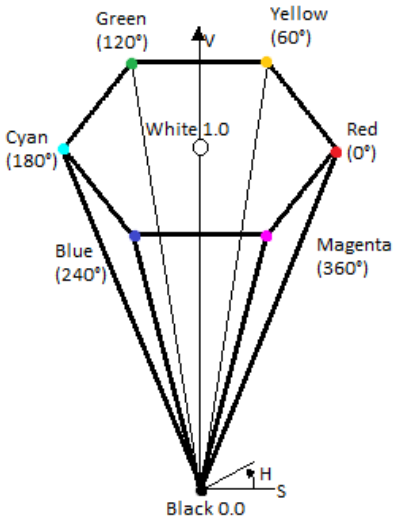
z čehož druhý z uvedených (1.5) odpovídá mezinárodnímu doporučení ITU-R BT.709-3 pro HDTV. [19]

1.6.3 YCbCr

Barevný model YCbCr byl vynalezen jako součást normy ITU-R BT.601 během vývoje celosvětového digitálních komponent video standardu. Tento barevný model vychází z barevného prostoru YUV, ve kterém je oddělen jas s chromatickými složkami, které ovlivňují výslednou barvu. Při kompresi obrazu je možno se dopustit větších nepřesností u chromatických složek (chrominance), než u jasové složky Y(luminance), kterou zrak rozpoznává lépe. Komponenty Cb a Cr jsou modrý a červený chrominanční komponenty. Při převodu obrazu z RGB do YCbCr jsou výsledkem tři samostatné složky obrazu, se kterými lze např. při komprimaci pracovat samostatně. [2, 10]

1.6.4 HSV

Barevné prostory HSI a HSV byly vyvinuty pro intuitivnější manipulaci s barvou, a aby se více přibližovali způsobu, jakým lidé vnímají a interpretují barvu. Rozdíl mezi HSI a HSV je výpočet jasové složky I/V, která určuje distribuci a dynamický rozsah obou jasových složek I nebo V a saturace S. Barevný prostor HSI je nejlepší pro výpočet tradičních funkcí zpracování obrazu, jako například konvoluce, ekvalizace, histogram atd., ty pracují především s hodnotou jasu, protože I je na R, G a B nezávislá. Barevný prostor HSV se preferuje pro manipulaci s odstínem a sytostí barvy, protože nabízí větší dynamický rozsah saturace. Obrázek znázorňuje barevný



Obr. 1.5: Šestiúhelníkový barevný model HSV

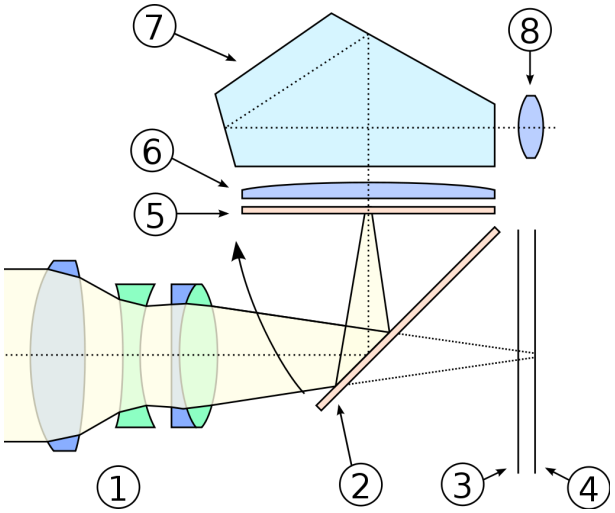
.

model HSV, horní část šestiúhelníku odpovídá hodnotě $V = 1$ nebo maximální intenzitě barev. Bod v základně šestiúhelníku je černý (hodnota $V = 0$). Doplňkové barvy jsou proti sobě v úhlu 180° (naměřené hodnoty H), úhlem kolem svislé osy (V), počínaje červenou v 0°. Hodnota S je poměr v rozsahu od 0 do 1 na svislé středové ose V. Přidání bílé odpovídá snížení hodnoty S, beze změny V, naopak přidání černé odpovídá poklesu V, beze změny S. Barevné tónování je způsobeno snížením S a V.

1.7 DSLR

DSLR je zkratka angl. slova (Digital single-lens reflex camera), který by se dal do češtiny přeložit jako digitální jednooká zrcadlovka. Jedná se o digitální kameru, kombinující optiku a mechanismus zrcadlovky s jedním objektivem a digitálním snímačem, namísto dřívějšího fotografického filmu. Schéma reflexního designu je primárním rozdílem mezi DSLR a jinými digitálními fotoaparáty. V reflexním provedení se světlo pohybuje skrz objektiv, poté do zrcadla, které střídavě posílá do hledáčku nebo obrazového snímače. Tradiční alternativou by bylo mít hledáček se samostatným objektivem, z tohoto designového důvodu název jednooká. Při užití jen jedné čočky se obraz viděný v hledáčku příliš neliší od toho, co zachytil senzor fotoaparátu.To je také předností DSLR, kterým se odlišuje od jiných typu digitálních fotoaparátů, kdy hledáček představuje přímý optický pohled skrze objektiv, ještě předtím, než je zachycen obrazovým snímačem fotoaparátu a následně zobrazen na digitální obra-

zovce. Na obr. (1.6) je vidět jednoduché schéma funkčnosti DSLR fotoaparátu. Kde



Obr. 1.6: Schéma DSLR. [Převzato z [22]]

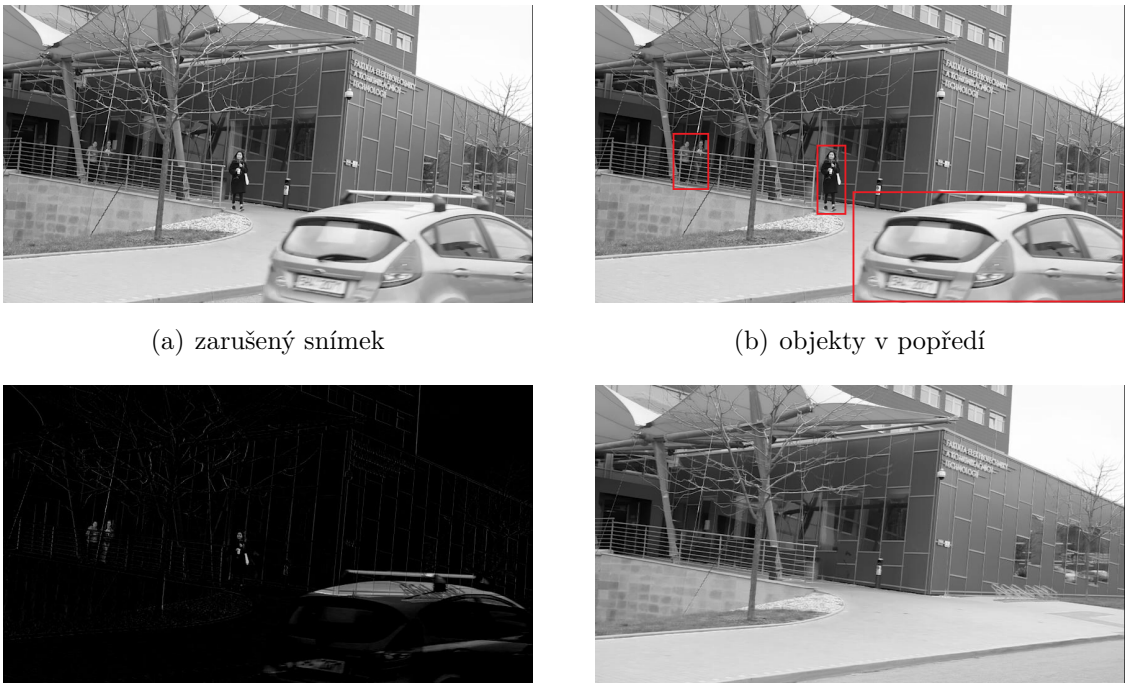
.

1. je objektiv, 2. polopropustné zrcadlo, 3. uzávěrka, 4. obrazový senzor, 5. matná zaostřovací plocha, 6. kondenzní čočka, 7. pentaprizmatický hranol, 8. hledáček.

1.8 Úloha oddělení pohybujících se objektů od pozadí

Jak již bylo naznačeno v úvodu. Jestliže budeme vytvářet statické video určitého objektu nebo obecněji řečeno neměnného pozadí, které bude zarušené pohybujícími objekty v popředí, je možné tyto rušivé elementy vyfiltrovat a získat snímek pozadí.

Pro různé počítačové aplikace zpracování obrazu, či videa, je odečítání pozadí rychlým způsobem, jak lokalizovat pohyblivé objekty ve videu pořízeném statickou kamerou. Z tohoto hlediska bývá detekce pohybu často prvním krokem vícekrokového, propracovanějšího počítačového sledovacího systému (sledování vozidel, rozpoznání osob, sledování divoké přírody atp.). Uvedené příklady jsou také důvodem, proč je obvykle potřeba, aby tento systém byl co nejrychlejší a nejjednodušší. Většina metod odečítajících pozadí zakládá na faktu, že pohybující objekty mají odlišné barvy než pozadí. Jak již bylo několikrát zmíněno, velice důležitou osvědčenou podmínkou je pečlivé statické umístění s pevným pozadím, ideálně bez šumu (což je i náš případ). Ne vždy ovšem je odečítání pozadí snadné nebo dokonce možné. Videa mohou být postižena poměrem signál/šum, způsobených nízkou kvalitou kamery,



(a) zarušený snímek

(b) objekty v popředí

(c) separované popředí

(d) separované pozadí

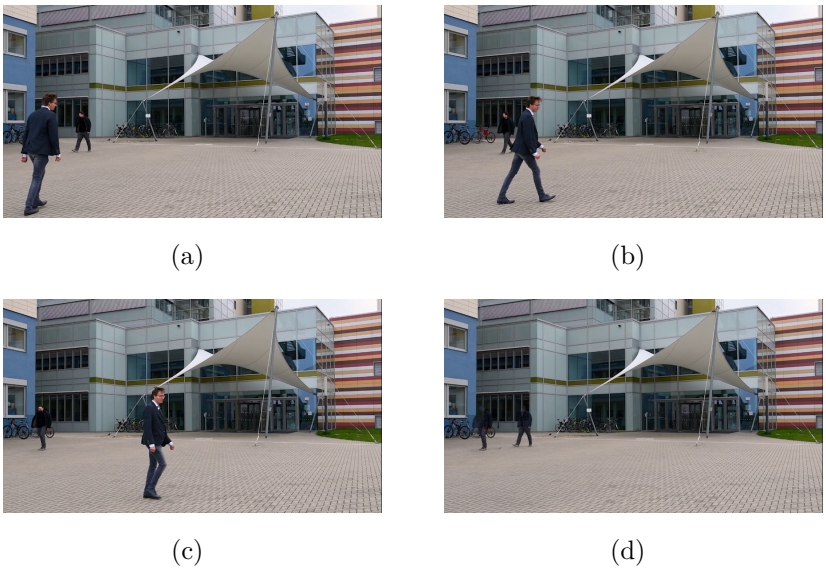
Obr. 1.7: Demonstrace oddělení pozadí a popředí.

artefakty komprese, či velmi rušné prostředí. Špatné výsledky by mohly být indukovány také změnami osvětlení ve videu (ať už postupnými, či skokovými), dostatečně nepevným pozadím (vlny na vodní hladině, otřesy objektů vlivem větru) nebo chvění fotoaparátu/kamery při pořizování snímku. Krom těchto vnějších vlivů může nastat špatný výsledek ve chvíli, kdy pohybující se objekt bude vykazovat velice podobné barvy těm v pozadí, čímž by došlo k maskování. [20]

Na obrázku (1.7) je naznačena přesná situace jednoho snímku (*framu*) z video sekvence. Na snímku (a) vidíme jeden tento *frame* z videa natáčeného pozadí budovy, před kterou prochází lidé, v tomto konkrétním příkladě 3 postavy a projíždějící automobil. Tyto rušivé, pohyblivé elementy v popředí jsou označeny v červených rámečcích na obrázku (b). Následující snímek (c) demonstruje již samotné separované popředí na černém pozadí. Poslední ze čtveřice snímků (d) zobrazuje separované pozadí objektu zájmu bez pohyblivého popředí.

1.9 Mediánová metoda

V úvodu bylo naznačeno, že na trhu zpracování digitálního obrazu, potažmo videa, existuje profesionální program Photoshop od firmy Adobe, který je ve velké míře komerčně využívaný pro zpracování a úpravu fotografií. Tento program obsahuje



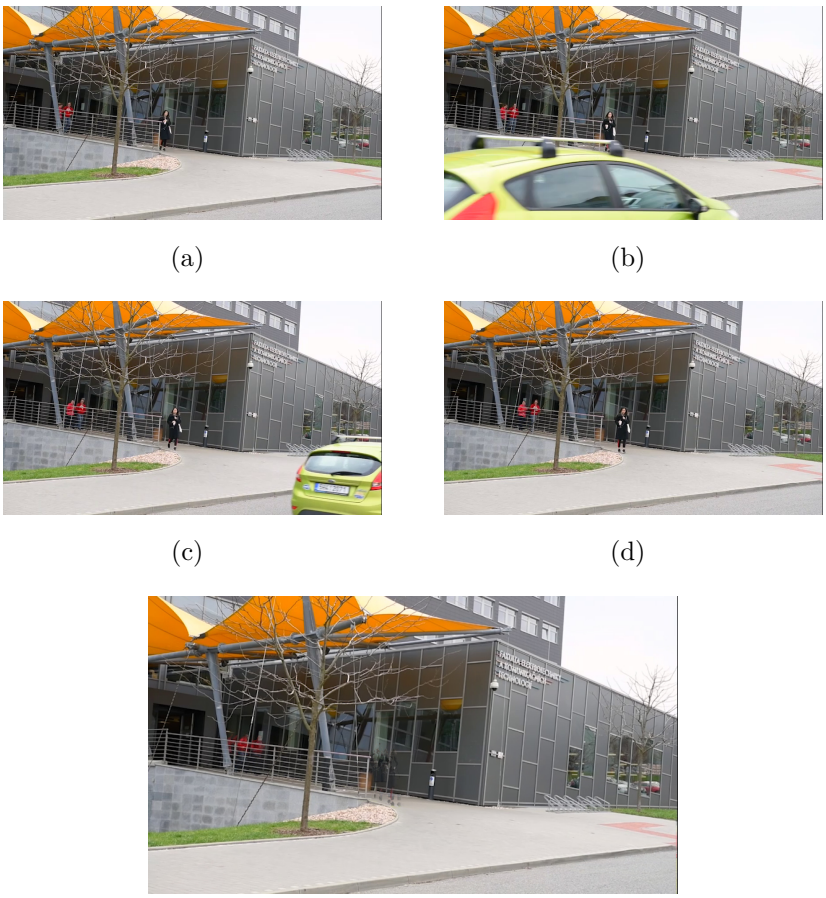
(a) (b)

(c) (d)

Obr. 1.8: Sekvence snímků (a),(b),(c) a výsledné pozadí (d).

klasické úpravy obrazu, jako jsou barevné úpravy, ořez, změna velikosti, retušování, malování, kreslení a textové nástroje atd., až po sofistikovanější funkce, jako například práce s vrstvami, efekty, či užití různých zásuvných VST modulů. Mimo tuto nepřebernou škálu funkcí obsahuje také řešení pro naše potřeby oddělení neměnného pozadí. Pokud bychom z námi natočené video sekvence získali/vystřihli několik snímků, tak abychom měli sekvenci jednotlivých obrázků, jelikož Photoshop pracuje s jednotlivými obrázky, ne s video soubory. K vystřižení jednotlivých snímků z video souboru můžeme využít program VLC Media Player, využívaný především pro přehrávání právě video souborů. V záložce: *Video>Vytvořit snímek* si můžeme uložit aktuální snímek z časové osy videa do primární složky v PC - *Tento počítač/Obrázky*.

V programu Photoshop v následující pořadí záložek *Soubor>Statistika>Skripty...*, kde vybereme režim balíčku Medián a načteme sekvenci snímků. Program nám po krátké chvíli vytvoří pozadí bez zarušení, který si můžeme vyexportovat. Na obr.(1.8) můžeme vidět sekvenci tří jednotlivých vystřižených snímků z vlastního video souboru (a), (b) a (c), poslední snímek (d) je již výsledkem mediánové statistky v programu Photoshop, vyexportovaný ve stejné kvalitě jako zdrojové obrázky. Stejně tak na obr.(4.2) můžeme vidět sekvenci snímků vystřižených z vlastního videa (obrázky (a)-(d)). A na obrázku (e) získané pozadí. Jelikož v první variantě na obr.(1.8) měla Mediánová statistika k dispozici pouze tři obrázky, ve druhé variantě o jeden více, na výsledných pozadích jsou vidět artefakty snímku. Oba výsledné snímky separovaného pozadí v případě obr.(1.8) i (4.2) jsou znehodnoceny „duchy“, případně rozmazanými tmavými, či barevnými místy. Tyto artefakty jsou způsobeny nedo-



(a) (b)

(c) (d)

(e)

Obr. 1.9: Sekvence snímků (a) - (d) a výsledné pozadí (e)

statkem obrazových informací, jelikož byl načten jen opravdu malý počet snímků ze statického videa. Při vytvoření více snímků z videa a jejich načtení, řádově 10 a více, se tyto artefakty na výsledných pozadích neobjevily.

2 ŘÍDKÁ A NÍZKOHODNOSTNÍ REPREZENTACE

Signály (zvuk, obraz, video) s tzv. řídkou reprezentací jsou pod drobnohledem odborníků bezmála již třicet let, s největší vlnou na přelomu tisíciletí. Prozatím stěžejní výsledky v teorii řídkých reprezentací v oblasti zpracování signálů byly prezentovány mezi roky 2000 až 2006. Za prvními publikovanými aplikacemi stáli většinou matematici, jelikož stěžejními oblastmi tohoto zaměření jsou především: lineární algebra, funkcionální analýza, teorie pravděpodobnosti a optimalizace. Řídkou reprezentací je myšleno, že daný signál může být vyjádřen přesně nebo lze z vybraného reprezentačního systému velmi schopně aproximovat lineární kombinací velmi malých počtů vektorů. V dnešní době je k dispozici obrovské množství aplikací, především v oblasti zpracování obrazu, což je dáno častou řídkou reprezentací obrazů ve vlnkové transformaci. Mezi nejpopulárnější aplikace lze jistě zařadit tzv. komprimované snímání, které přeškročilo omezení klasického vzorkovacího teorému, čímž významně ovlivnilo také oblast teorie informace.

Další oblastí zájmu se také stala tzv. nízkohodnostní reprezentace. Což znamená, že je možné uspořádat signál do matice, která má buď přesně anebo přibližně nízkou hodnost, kdy takovýchto signálů je nečekaně hodně. Aplikace v tomto směru dokáží doplňování chybějících údajů datových polí, ale i zpracování videa.[3]

2.1 Řídké reprezentace signálů

Reprezentaci signálů můžeme řešit jako soustavu lineárních rovnic $\mathbf{Ax} = \mathbf{y}$, kde $\mathbf{A}$ je matice plné řádkové hodnosti, $\mathbf{y}$ je známý vektor (pozorování, signál, měření) a $\mathbf{x}$ je neznámý, hledaný vektor. Takovýto systém se označuje jako nedourčený, tzn. že máme více neznámých než kolik obsahuje rovnic. Předpokládá se, že matice $\mathbf{A}$ je plné hodnosti, proto existuje nekonečně mnoho řešení. Pojem řídkost definuje vztah 2.1. Vektor $\mathbf{x}$ nazveme *k-řídkým*, pokud platí následující:

$$\|\mathbf{x}\|_0 \leq k. \tag{2.1}$$

Výraz $\|\mathbf{x}\|_0$ je normou vektoru. Norma vektoru je číslo, které udává jeho „velikost“. Nejčastější normou v praxi je norma eukleidovská, zde je však definována *lp*-norma:

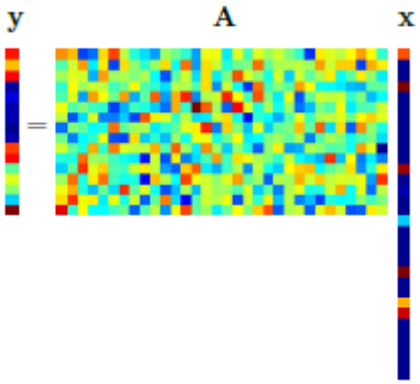
$$\begin{aligned}
\|\mathbf{x}\|_p &:= \left(\sum_{i=1}^{N} |x_i|^p\right)^{\frac{1}{p}} \quad pro\ 1 \leq p < \infty \\
\|\mathbf{x}\|_p &:= \sum_{i=1}^{N} |x_i|^p \quad pro\ 1 < p < \infty \\
\|\mathbf{x}\|_\infty &:= \max_i |x_i|\,, \\
\|\mathbf{x}\|_0 &:= |supp\,(x)|\,.
\end{aligned} \tag{2.2}$$

O normu se přísně vzato jedná jen v případě pro $1 \leq p < \infty$. Pro zjednodušení, je však pro všechna p označena jako l_p-norma. Norma $\|\cdot\|_1$ znamená součet absolutních hodnot prvků vektoru a norma $\|\cdot\|_0$ počet nenulových složek vektoru.

Nás zajímají řídká řešení, které budou obsahovat co nejvíce nulových složek, jde tedy o nultou normu, konkrétně tuto úlohu:

$$arg\min_x \|\mathbf{x}\|_0 \ \ vzhledem\ k\ \mathbf{Ax} = \mathbf{y}, \tag{2.3}$$

kde známe vektor $\mathbf{y} \in \mathbb{C}^m$ a matici $\mathbf{A} \in \mathbb{C}^{m \times N}$, která definuje lineární zobrazení. Předpokládáme pouze případy, kdy $m < N$, resp. $m << N$, a $\mathbf{A}$ je plné řádkové hodnosti. Schéma příkladu je na obr. 2.1.



Obr. 2.1: Schéma nedourčeného systému rovnic $\mathbf{Ax} = \mathbf{y}$. Převzato z [3]

.

Matice $\mathbf{A}$ se nejčastěji nazývá slovník (disctionary), její sloupce se nazývají atomy (atoms). [3]

V počítači je obraz reprezentován jako matice, jejíž jednotlivé prvky jsou hodnotami určité barevné škály. Abychom dostali vektor $\mathbf{y}$ a mohli sestavit lineární rovnici, je potřeba obraz zvektorizovat (naskládat po sloupcích za sebe do jednoho vektoru).[18]

Obvykle může být signál pod vlivem šumu, či chyb řešení, což může způsobit odchýlení od přesného měření. V praxi se proto povoluje malá odchylka:

$$arg\min_x \|\mathbf{x}\|_0 \;\; vzhledem\ k \;\; \|\mathbf{Ax} = \mathbf{y}\|_p \leq \delta, \tag{2.4}$$

kde nejčastěji uvažujeme $p = 2$.

2.2 Postačující podmínky jednoznačnosti řešení

Pro rozhodování o existenci a vlastnostech řešení je pojem *spark* (doslova „jiskra“) důležitou vlastností. Číslo spark($\mathbf{A}$) je definováno jako nejmenší počet sloupců matice $\mathbf{A}$, které jsou lineárně nezávislé

$$arg\min_x \|\mathbf{x}\|_0 = \min_{\mathbf{z}\in ker\mathbf{A}} \; \mathbf{z}\neq 0 \, \|\mathbf{z}\|_0 \,, \tag{2.5}$$

kde $ker\boldsymbol{A}$ je jádro lineárního zobrazení určeného maticí $\mathbf{A}$ (tj. podprostor tvořený všemi řešeními homogenní soustavy lineárních rovnic $\mathbf{Ax}$=$\mathbf{0}$).

Pro nenulovou matici $\mathbf{A} \in \mathbb{C}^{m\times N}$, kde $m < N$, platí $spark(\mathbf{A}) \in \{2, ..., m+1\}$, současně hodnoty 2 je dosaženo, když jeden sloupec je přímo násobkem jiného. Čím je menší spark, tím řidší musí být vektor $\mathbf{x}$, aby byla zachována jedinečnost řešení.

Jestliže má soustava $\mathbf{Ax} = \mathbf{x}$ splňující

$$\|\mathbf{x}\|_0 < \frac{spark\mathbf{A}}{2}, \tag{2.6}$$

pak $\mathbf{x}$ je nutně nejřidší možné řešení a žádné jiné řešení se stejnou řídkostí neexistuje. Vzhledem k tomu, že nalezení $spark(\mathbf{A})$ je výpočetně náročné, je snaha nalézt jednodušší způsob zjištění jedinečnosti řešení.

Vzájemná koherence (mutual coherence) matice $\mathbf{A}$ je definovaná jako největší absolutní normalizovaný skalární součin dvou různých sloupců matice $\mathbf{A}$,

$$\mu(\mathbf{A}) = \max_{1\leq 1, k\leq N} \frac{\left|\mathbf{a}_j^T \mathbf{a}_k\right|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2}, \tag{2.7}$$

kde $\mathbf{a}_j$ značí j-tý sloupec matice $\mathbf{A}$.

Vzájemnou koherenci rovna nule má pouze unitární matice, jelikož její sloupce jsou po dvou ortogonální, tedy čitatel je roven nule, oproti tomu nedourčená soustava bude mít koherenci nenulovou.

Pro libovoulnou matici $\mathbf{A}$ platí:

$$spark(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})}, \tag{2.8}$$

což nám umožňuje zdola ohraničit spark($\mathbf{A}$) s nižší výpočtovou náročností. Z čehož plyne: Pokud má soustava $\mathbf{Ax} = \mathbf{b}$ řešení $\mathbf{x}$ splňující

$$\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right), \tag{2.9}$$

pak $\mathbf{x}$ je nutně nejřidší možné a jediné takové. Navíc lze toto řešení dosáhnout l_1-minimalizací. Snahou je hledat co nejvíce nekoherentní slovníky (ty co mají nejnižší koherenci), přibližující se ortogonální matici. [3]

2.3 Metody řešení

Předpokládáme-li, že $spark(\mathbf{A}) > 2k_0$ a existuje k_0-řídké řešení soustavy. Pak je toto řešení nejřidší možné a jednoznačné. Pro vyhledání přesného řešení by bylo potřeba projít všech($\frac{m}{k_0}$) kombinací podmnožin atomů matice, což je většinou velmi časově namáhavé. Z tohoto důvodu se vytvořilo několik aproximačních metod řešení tohoto problému, které ačkoliv nejsou tolik přesné, jsou rychlejší. Dělí se do dvou nejvýznamnějších kategorií - algoritmy hladové a relaxační.[3]

2.3.1 l_1 relaxace

(Pseudo)norma l_0 není konvexní funkce a není tudíž možné pro tento problém užít žádnou z dnes řady dostupných metod a algoritmů konvexní optimalizace. Jelikož ale normy l_p jsou konvexní pro $p \geq 1$, přichází v úvahu otázka, zda-li by nešlo nultou normu aproximovat nejbližší konvexní normou tj. l_1, čímž by se řešila úloha

$$arg\min_x \|\mathbf{x}\|_1 \ \ vzhledem\ k\ \mathbf{Ax} = \mathbf{y}. \tag{2.10}$$

Jak bude dále ukázáno, za určitých podmínek je možné využít l_1 normu namísto l_0, v některých případech se i řešení shodují.[3]

2.3.2 Podmínky ekvivalence řešení l_0- a l_1- minimalizace

Vlastnost nulového prostoru

Pro $T \subset \{1, ..., n\}$ označíme $\mathbf{x}_T \in \mathbb{C}^n$ vektor odvozený z $\mathbf{x} \in \mathbb{C}^n$ tak, že prvky na pozicích patřících do množiny T zachovámě a zbytek vynulujeme. Komponent T značíme $T^c = \{1, ..., n\} / T$.

Řekněme, že matice $\mathbf{A} \in \mathbb{C}^{m \times n}$ má vlastnosti nulového prostoru (Null Space Property, NSP) řádu k s konstantou $\gamma \in (0, 1)$, zda-li platí:

$$\|\eta_T\|_1 \leq \gamma \|\eta_{T^c}\|_1, \tag{2.11}$$

pro všechny množiny $T \subset \{1, ..., N\}$, $|T| \leq k$ a pro všechny vektory $\eta \in ker\mathbf{A}$. Vlastnost nulového prostoru zajišťuje, že k-řídké řešení je jednoznačné a lze jej nalézt za pomoci l_1-minimalizace. Ověření podmínky však není jednoduché. Reálné signály nebývají řídké v pravém slova smyslu, avšak mají místo nulových složek malé nenulové hodnoty, je tedy potřeba definovat chybu aproximace řídkým vektorem. Chyba nejlepší aproximace vektoru $\mathbf{x} \in \mathbb{C}^n$ k-řídkým vektorem $\mathbf{z}$ v normě l_p je definována jako

$$\sigma_k(\mathbf{x})_p = \inf_{\mathbf{z} \in \sum_k^n} \|\mathbf{x} - \mathbf{z}\|_p , \tag{2.12}$$

kde $\sum_k^n = \{\mathbf{x} \in \mathbb{C}^n : \|\mathbf{x}\|_0 \leq k\}$. Následující tvrzení dává odhad chyby i v ostatních případech.

Nechť matice $\mathbf{A} \in \mathbb{C}^{m \times N}$ splňuje NSP řádu s konstantou $\gamma \in (0, 1)$. Pak $\mathbf{x} \in \mathbb{C}^N$, $\mathbf{y} = \mathbf{A}\mathbf{x}$ a $\mathbf{x^*} \in \mathbb{C}^N$ je řešení l_1- minimalizace a $\sigma_k(\mathbf{x})_1$ je chyba nejlepší aproximace $\mathbf{x}$ k-řídkým vektorem v l_1 normě. Pak

$$\|\mathbf{x} - \mathbf{x^*}\|_1 \leq \frac{2(1+\gamma)}{1-\gamma} \sigma_k(\mathbf{x})_1. \tag{2.13}$$

Jestliže za splnění předpokladů tvrzení existuje nějaké nejvýše k-řídké řešení soustavy $\mathbf{y} = \mathbf{A}\mathbf{x}$, pak $\sigma_k(\mathbf{x})_1 = 0$ a nula na pravé straně rovnice vynutí $\mathbf{x} = \mathbf{x^*}$, protože $\|\mathbf{x} - \mathbf{x^*}\|_1 = 0$, což znamená, že ono řídké řešení nalezneme i l_1-optimalizací.

Zároveň však platí opak, že pokud lze ze soustavy $\mathbf{A}\mathbf{x} = \mathbf{y}$ rekonstruovat všechny k-řídké vektory $\mathbf{x}$ pomocí l_1-minimalizace, pak matice $\mathbf{A}$ splňuje NSP řádu k s nějakou (jinou) konstantou $\gamma \in (0, 1)$. NSP s větší z těchto dvou konstant je tak ekvivalentní řídké l_1-rekonstrukci. [3, 18]

2.4 Komprimované snímání

Komprimované snímání (compressed sensing, compressive sampling, CS) si klade za cíl přístup za předpokladu řídkosti ve vhodné reprezentaci, snímání signálu lineárně a neadaptivně, pouze tolikrát kolik je třeba.

Konvenčním přístupem ke vzorkování signálů nebo obrazů následují Shannonův slavný teorém, kdy vzorkovací frekvence musí být nejméně dvojnásobkem maximální frekvence signálu (tzv. Nyquistův teorém). Ve skutečnosti je tento přístup základem téměř veškerých protokolů získávání signálu užívaných ve spotřební audio a vizuální elektronice, zdravotnických zobrazovacích zařízení, rozhlasových přijímačů atd. U signálů jako jsou obrazy, které jsou přirozeně omezeny, vzorkovací rychlost není definována Shannonovým teorémem, ale požadovaným časovým nebo prostorovým rozlišením. V takovýchto systémech je ovšem běžné používání antialiasingového nízkoprůchodového filtru, aby se před vzorkováním signál omezil, Shannonova věta tedy hraje implicitní roli. [3, 4, 6]

Obecně můžeme vyjádřit rekonstrukci signálu pomocí CS jako optimalizační problém:

$$\min_{M} f(\mathbf{M}) \; vzhledem \; k \; E(\mathbf{M}) = \mathbf{Y}. \tag{2.14}$$

kde $\mathbf{Y}$ jsou vstupní data, $\mathbf{M}$ hledaný výstupní obraz a E je lineární operátor. Funkce $f(\mathbf{M})$ je matematický zápis empirických znalostí vstupních dat, která se také někdy označuje jako regularizace. Velmi často obsahuje normy:

$$\begin{aligned}
&\|\mathbf{X}\|_0 \; počet \; nenulových \; x_{i,j}, \\
&\|\mathbf{X}\|_1 \;\; \sum_{i,j} |x_{i,j}|, \\
&\|\mathbf{X}\|_2 \; \sqrt{\sum_{i,j} |x_{i,j}|^2}, \\
&\|\mathbf{X}\|_\infty \; \max_i |x_i|, \\
&\|\mathbf{X}\|_* \; \sum_i \sigma_i(\sigma_i \; singulární \; čísla \; \mathbf{X}).
\end{aligned} \tag{2.15}$$

Optimalizační úlohu CS je možné řešit více způsoby. Většinou se může v měření vyskytovat šum, potom by se změnil tvar na

$$\min_{M} \frac{1}{2} \|E(\mathbf{M}) - \mathbf{Y}\|_2^2 + \lambda f(\mathbf{M}), \tag{2.16}$$

kde $\|\cdot\|_2$ představuje Frobeniovu normu. Regularizační parametr $\lambda > 0$ je volitelným parametrem, určující charakter měření. Jeho nastavení je pro výsledek velice důležité, jestliže je parametr λ příliš velký, rekonstrukce by byla tzv. přeregularizovaná, v opačném případě velmi nízké hodnoty λ by se regularizace prakticky neprojevila a výsledek by obsahoval artefakty způsobené podvzorkováním dat. Pokud je $f\mathbf{M}$ konvexní funkce, je možné úlohu řešit algoritmy konvexní optimalizace. Často využívané jsou proximální metody, či metoda multiplikátorů střídajících směr (ADMM). [6]

2.4.1 Nízká maticová hodnost

Komprimované snímání CS je možné také využít pro signály, které chápeme jako matice (nejčastěji tedy obrazy, videa, či datové záznamy). Namísto řídkosti se zde pracuje s předpokladem nízké hodnosti matice $\mathbf{X} \in \mathbb{R}^{n_1} \times n_2$, případně $\mathbf{X} \in \mathbb{C}^{n_1} \times n_2$. Měření má opět tvar lineární kombinace, nyní prvků z $\mathbf{X}$. Formálně tedy lineární zobrazení $A : \mathbb{R}^{n_1} \times n_2 \to \mathbb{R}^m$, kde $m << n_1 n_2$, tedy měření $\mathbf{y}$ dostaneme tak, že

$$\mathbf{y} = A(\mathbf{X}) \in \mathbb{R}^m. \tag{2.17}$$

Abychom mohli rekonstruovat $\mathbf{X}$ z $\mathbf{y}$, budeme předpokládat, že $\mathbf{X}$ má hodnost nejvýše $rank(\mathbf{X}) = r << min\{n_1, n_2\}$. Prostý přístup k řešení optimalizačního problému

$$arg\min_{x} rank(\mathbf{X}) \; vzhledem \; k \; \mathbf{y} = a(\mathbf{X}), \tag{2.18}$$

je znovu NP-těžký, avšak můžeme jej relaxovat.

V části 2.1 je uveden SVD rozklad matice a je uvedeno, že matice $\mathbf{X}$ má hodnost r tehdy, jestliže vektor $\sigma = \sigma(\mathbf{X})$ singulárních čísel má řídkost r, tj. jestliže platí $rank(\mathbf{X}) = \|\sigma(\mathbf{X})\|_0$. Zde je již možnost k relaxaci analogické s relaxací na l_1-normu pro vektory. Nukleární norma je tedy definována jako l_1-norma singulárních čísel, $\|\mathbf{X}\|_* = \|\sigma(\mathbf{X})\|_1$, tedy můžeme uvažovat relaxovaný minimalizační problém

$$arg\min_x \|\mathbf{X}\|_* \ vzhledem\ k\ \mathbf{y} = a(\mathbf{X}), \tag{2.19}$$

Tento problém je konvexní, a tudíž účinně řešitelný, v poslední době se vyskytuje řada algoritmů pro dosažení optima. [3, 6]

2.5 Konvexnost

Konvexní optimalizační problém je typ úlohy, kdy

$$min\ f_0(x)\ vzhledem\ k\ f_i(x) \leq b_i,\ i = 1, ..., m, \tag{2.20}$$

kde funkce $f_0, ..., f_m : \mathbf{R}^n \to \mathbf{R}$ jsou konvexní, t. vyhovují

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \tag{2.21}$$

pro všechny $x, y \in \mathbf{R}^n$ a všechny $\alpha, \beta \in \mathbf{R}$ s $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$. Problém nejmenších čtverců a lineární programování jsou oba speciální případy obecného konvexního optimalizačního problému.

Obecně neexistuje žádný analytický vzorec pro řešení konvexního optimalizačního problému, avšak existují velmi účinné metody jejich řešení. Zatím nelze tvrdit, že řešení obecných konvexních optimalizačních problémů je vyspělou technologií, jako např. řešení nejmenších čtverců nebo lineárního programování, jistě je ale velmi atraktivní výzkumnou oblastí a bude do budoucna velmi rozvinuto.

Problém nejmenších čtverců je optimalizačním problémem bez omezení (tj. $m = 0$) a cíl, kterým je součet čtverců z hlediska formy $a_i^T x - b_i$:

$$min\ f_0(x) = \|\mathbf{A}x - b\|_2^2 = \sum_{i=1}^{k} (a_i^T x - b_i), \tag{2.22}$$

kde $A \in \mathbb{R}^{k \times n}$ s $k \geq n$, a_i^T jsou řádky matice $\mathbf{A}$ a vektor $x \in \mathbf{R}^n$ je optimalizační proměnná.

Řešení problému nejmenších čtverců může být omezeno na řešení sady lineárních rovnic

$$(\mathbf{A}^T\mathbf{A})x = \mathbf{A}^T b, \tag{2.23}$$

takže máme analytické řešení $x = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T b$. Pro řešení problému existují dobré algoritmy/softwarové implementace s vysokou přesností a velmi dobrou spolehlivostí. Problém nejmenších čtverců lze vyřešit v čase přibližně úměrném n^2k, se známou konstantou. Současné stolní počítače jsou schopné řešit tento problém se stovkami proměnnými a tisíci podmínkami během několika sekund. V mnoha případech můžeme řešit mnohem větší problémy nejmenších čtverců, využitím některých speciálních struktur v koeficientech matice $\mathbf{A}$. Předpokládáme-li například, že matice $\mathbf{A}$ je řídká, což znamená, že má mnohem méně než kn nenulových položek. Využitím řídkosti obvykle můžeme řešit problém nejmenších čtverců mnohem rychleji než n^2k. [21]

2.6 Proximální algoritmy

V dnešní době je možné pro řešení optimalizačních úloh využít nepřebernou škálu algoritmů. Jak již bylo naznačeno, žádná z těchto metod nemůže najít vždy přesné řešení v polynomiálním čase, proto se přistupuje k metodám aproximativním. Dle [3] lze s trochou zjednodušení tvrdit, že se algoritmy dělí do tří skupin, i když vzhledem k vzestupu nových metod se ukazuje, že toto dělení je velmi hrubé.

- *Hladové (greedy) algoritmy*, jejichž hlavní princip tkví v hledání jednoho (případně více) „nejvýznamnějších“ atomů. Důležitým faktem je, že v dalším průběhu algoritmu již vybraný atom neztrácí podíl na konečném řešení. Výhodou je nízká složitost těchto metod, naopak nevýhodou je skutečnost, že není zaručené dosažení globálního optima, dokonce se může stát, že algoritmy zcela selžou. Mezi zástupce této kategorie se řadí MP (Matching Pursuit), OMP (Orthogonal Matching Pursiut) a další odvozeniny.
- *Relaxační* algoritmy nesou název po metodě založené na l_1-relaxaci. Tyto algoritmy se snaží dosáhnout řešení přesnému nebo alespoň blízkému za určitých podmínek. Jmenovitě BP (Basic Pursuit), modifikovaná LARS (Least Angle Regresion, homotopy method), IRLS (někdy také FOCUS) nebo Dantzig Selector. Vzhledem k tomu, že l_1-norma je konvexní funkce,lze využít tzv. proximální algoritmy.
- *Prahovací* (thresholding) a *hybridní* algoritmy jsou mimo tyto kategorie a využívají přednosti z jednotlivých skupin. Jedním z dalších je A*OMP, využívající A*-algoritmu prohledávání informačních stromů, dále pak také algoritmy aproximativního předávání zpráv v grafovém modelu (approximate message passing).

Proximální algoritmy jsou postupy z teorie optimalizace, zahrnující celé spektrum konvexních úloh včetně relaxovaných s l_1-normou. Ačkoliv konvexní úlohy byly známy již delší dobu, svoji oblibu získali až především díky řídkým reprezentacím.

Optimalizace vychází z iterativní minimalizace činitelů, kdy jsou známy podmínky, za jakých algoritmus konverguje k optimu úlohy. U proximálních metod nebývá rychlost tak vysoká, jako u některých algoritmů, za cenu využívané flexibility. [3]

2.7 Proximální optimalizační metody

Optimalizační úlohu (2.16) komprimovaného snímání je možné zdárně řešit pomocí tzv. proximálních metod, které je ideální využít na součet dvou a více konvexních funkcí, z nichž často bývá diferencovatelná pouze jedna, případně v některých algoritmech nemusí být diferencovatelná žádná z nich. Ty jsou vybudovány na proximálním operátoru, který je zobecněním projekčního operátoru na konvexní množinu. [6]

2.7.1 Proximální operátor

Proximální operátor $prox_f(\mathbf{x})$ zdola polospojité konvexní funkce f v bodě $\mathbf{x}$ je definován jako

$$prox_f(\mathbf{x}) = arg\min_y \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + f(\mathbf{y}). \tag{2.24}$$

Pokud za funkci f vezmeme indikátorovou (charakteristickou) funkci ι_C množiny C

$$\iota_C : \mathbf{x} \mapsto \begin{cases} 0, \mathbf{x} \in C \\ \infty, \mathbf{x} \notin C \end{cases}, \tag{2.25}$$

proximální operátor je totožný s projekčním operátorem

$$arg\min_y \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \iota_C(\mathbf{y}). \tag{2.26}$$

V algoritmech se také často využívá vlastnosti pro proximální operátor pro tzv. Fenchel-Rockafellarovu konjugovanou funkci f^* k funkci f. Navíc platí Moreauova identita: pro každé $\mathbf{x}$ a $\sigma > 0$ platí

$$prox_{\sigma f^*}(\mathbf{x}) = \mathbf{x} - \sigma prox_{\frac{f}{\sigma}}(\frac{\mathbf{x}}{\sigma}). \tag{2.27}$$

Pokud je funkce f separovatelná, tj. $f(\mathbf{u},\mathbf{v}) = f_1(\mathbf{u}) + f_2(\mathbf{v})$, lze separovat i proximální operátor:

$$\begin{aligned} prox_f(\mathbf{x},\mathbf{y}) =& arg\min_{\mathbf{u},\mathbf{v}} \frac{1}{2}\left\|\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} - \begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix}\right\|^2 + f(\mathbf{u},\mathbf{v}) = \\ &= arg\min_{\mathbf{u},\mathbf{v}} \frac{1}{2}\left\|\mathbf{x}-\mathbf{u}\right\|^2 + \frac{1}{2}\left\|\mathbf{y}-\mathbf{v}\right\|^2 + f_1(\mathbf{u}) + f_2(\mathbf{v}) = \\ &= arg\min_{\mathbf{u}} \frac{1}{2}\left\|\mathbf{x}-\mathbf{u}\right\|^2 + \frac{1}{2}\left\|\mathbf{y}-\mathbf{v}\right\|^2 + f_1(\mathbf{u}) + f_2(\mathbf{v}) = \\ &= \begin{pmatrix} prox_{f_1}\mathbf{x} \\ prox_{f_2}\mathbf{y}\end{pmatrix} \end{aligned} \tag{2.28}$$

Tento vztah je možné zobecnit na případ i více separovatelných proměnných. S pomocí této vlastnosti lze vyčíslit proximální operátor po separovatelných částech nezávisle na sobě.[6]

2.7.2 Proximální operátor pro l_1-normu a nukleární normu

Dle definice (2.24) lze napsat (zjednodušený zápis pouze pro vektor délky n):

$$prox_{\lambda\|\cdot\|_1}(\mathbf{x}) = arg\min_y \left\{ f(\mathbf{y}) := \frac{1}{2}\left\|\mathbf{x}-\mathbf{y}\right\|_2^2 + \lambda\left\|(\mathbf{y})\right\|_1 \right\}, \tag{2.29}$$

což lze rozepsat jako

$$\hat{y} = arg\min_y \frac{1}{2}\sum_{i=1}^{n}(x_i - y_i)^2 + \lambda\sum_{i=1}^{n}|y_i|\,. \tag{2.30}$$

Toto minimum se bude postupně hledat na jednotlivých intervalech, kde je absolutní hodnota diferencovatelná (tj. provede se derivace dle jednotlivých proměnných a položí se rovno nule):

1. $y_i > 0$

$$\frac{\partial f}{\partial y_i} = y_i - x_i + \lambda = 0,$$
$$y_i = x_i - \lambda > 0 \Rightarrow x_i > \lambda,$$

2. $y_i < 0$

$$\frac{\partial f}{\partial y_i} = y_i - x_i - \lambda = 0,$$
$$y_i = x_i + \lambda < 0 \Rightarrow x_i < -\lambda.$$

3. Jediným případem, který nebyl uvažován, je $y_i = 0$, které je hledaným minimem pro zbylé případy x_i, tedy $-\lambda \leq x_i \leq \lambda$.

Celkově tedy lze proximální operátor l_1-normy napsat jako

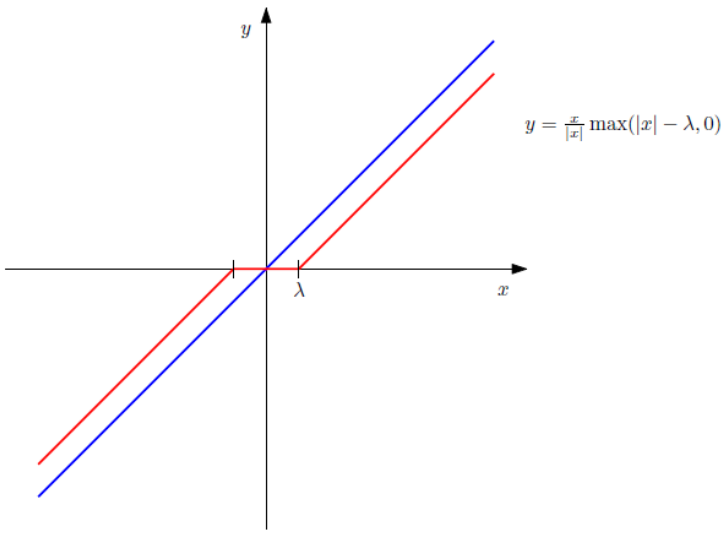
$$\hat{y}_i = \frac{x_i}{|x_i|} max(|x_i| - \lambda, 0). \tag{2.31}$$

Tato funkce se nazývá měkké prahování (Shrinkage operator), znázorněné na obr.(2.2). Měkké prahování se provádí pro každou složku vektoru $\mathbf{x}$ zvlášť. Proto se symbolicky označuje jako $prox_{\lambda\|\cdot\|_1} = soft(\mathbf{x}, \lambda)$.

Jelikož nukleární norma je definovaná jako l_1-norma singulárních čísel (viz (2.19)), je tedy proximální operátor nukleární normy měkké prahování singulárních čísel, formálně:

$$prox_{\lambda\|\cdot\|_*}(\mathbf{X}) = \sum_{l=1}^{n} soft(\sigma_l, \lambda)\mathbf{u}_l\mathbf{v}_l^* =: svt(\mathbf{X}, \lambda). \tag{2.32}$$

Pro podotknutí, hodnost matice se určuje dle počtu nenulových singulárních čísel a nukleární norma je aproximací hodnosti matice. Lze tedy vidět, že použitím proximálního operátoru nukleární normy nastává snižování hodnosti matice $\mathbf{X}$, jelikož se nulují ta singulární čísla, spadající pod zadaný práh λ.[6]



Obr. 2.2: Identita (červeně) a měkké prahování s prahem λ (modře). Převzato z [6]

.

2.8 Metoda multiplikátorů střídavého směru ADMM

ADMM je schopná řešit problém ve formě

$$min_{\mathbf{x}} f(\mathbf{x}) + g(A\mathbf{x}), \tag{2.33}$$

kde $\mathbf{x} \in \mathbb{C}^N$ a A:$\mathbb{C}^N \to \mathbb{C}^P$ je lineární operátor. Předpokládáme, že f, g jsou reálné konvexní funkce (možno i komplexní) proměnných. Tento problém může být formulovaný následovně:

$$min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + g(A\mathbf{x}) \;\; vzhledem\ k\ A\mathbf{x} - \mathbf{z}. \tag{2.34}$$

Pro vyřešení je formulován tzv. Augmented Lagradian multiplikátor jako

$$L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^\top (A\mathbf{x} - z) + \frac{\rho}{2} \|A\mathbf{x} - \mathbf{z}\|_2^2 , \tag{2.35}$$

kde $\rho > 0$ je váhovací parametr. ADMM se skládá ze tří kroků:

$$\begin{aligned} \mathbf{x}^{i+1} &= arg \min_x L_\rho(\mathbf{x}, \mathbf{z}^i, \mathbf{y}^i) \\ \mathbf{z}^{i+1} &= arg \min_z L_\rho(\mathbf{x}^{i+1}, \mathbf{z}, \mathbf{y}^i) \\ \mathbf{y}^{i+1} &= y^i + \rho(A\mathbf{x}^{i+1} + \mathbf{z}^{i+1}) \\ &, \end{aligned} \tag{2.36}$$

Je možné převézt ADMM do tzv. škálované formy, což je často více vhodné, to se provádí definováním reziduálního $\mathbf{r} = A\mathbf{x} - \mathbf{z}$. V tom případě mohou být poslední dva členy Augmented Lagragianového multiplikátoru, jako

$$\mathbf{y}^\top r + \frac{\rho}{2} \|\mathbf{r}\|_2^2 = \frac{\rho}{2} \left\| \mathbf{r} + \frac{1}{\rho}\mathbf{y} \right\|_2^2 - \frac{1}{2\rho} \|\mathbf{y}\|_2^2 = \frac{\rho}{2} \|\mathbf{r} + \mathbf{u}\|_2^2 - \frac{\rho}{2} \|\mathbf{u}\|_2^2 , \tag{2.37}$$

kde $\mathbf{u}$ je škálovaná dvojitá proměnná (scaled dual variable) jako $\mathbf{u} = \frac{\mathbf{y}}{\rho}$. Druhá ekvivalence je jen substitucí. První může být ověřena s použitím následujícího, za předpokladu, že $\mathbf{r}$ a $\mathbf{y}$ jsou reálné:

$$\begin{aligned} \frac{\rho}{2} \left\| \mathbf{r} + \frac{1}{\rho}\mathbf{y} \right\|_2^2 - \frac{1}{2\rho} \|\mathbf{y}\|_2^2 &= \frac{\rho}{2} \left\langle \mathbf{r} + \frac{1}{\rho}\mathbf{y}, \mathbf{r} + \frac{1}{\rho}\mathbf{y} \right\rangle - \frac{1}{2\rho} \langle \mathbf{y}, \mathbf{y} \rangle \\ &= \frac{\rho}{2} \left(\langle \mathbf{r}, \mathbf{r} \rangle + \left\langle \mathbf{r}, \frac{1}{\rho}\mathbf{y} \right\rangle + \left\langle \frac{1}{\rho}\mathbf{y}, \mathbf{r} \right\rangle + \left\langle \frac{1}{\rho}\mathbf{y}, \frac{1}{\rho}\mathbf{y} \right\rangle \right) + \frac{1}{2\rho} \langle \mathbf{y}, \mathbf{y} \rangle \\ &= \frac{\rho}{2} \|\mathbf{r}\|_2^2 + \frac{1}{2}\mathbf{r}^\top \mathbf{y} + \frac{1}{2}\mathbf{y}^\top \mathbf{r} \\ &= \frac{\rho}{2} \|\mathbf{r}\|_2^2 + \mathbf{y}^\top \mathbf{r} \\ &. \end{aligned} \tag{2.38}$$

Po výše uvedené úpravě je Augmented Lagrangian ve zmenšeném tvaru

$$L_\rho(\mathbf{x}, \mathbf{u}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{r} + \mathbf{u}\|_2^2 + \frac{\rho}{2} \|\mathbf{u}\|_2^2 . \tag{2.39}$$

Zmenšená verze ADMM je vyjádřena jako

$$\begin{aligned}\mathbf{x}^{i+1} &= arg\min_{x}\left(f(\mathbf{x}) + \frac{\rho}{2}\left\|A\mathbf{x} - \mathbf{z}^{i} + \mathbf{u}^{i}\right\|_2^2\right)\\ \mathbf{z}^{i+1} &= arg\min_{z}\left(g(\mathbf{z}) + \frac{\rho}{2}\left\|A\mathbf{x}^{i+1} - \mathbf{z} + \mathbf{u}^{i}\right\|_2^2\right)\\ \mathbf{u}^{i+1} &= u^{i} + A\mathbf{x}^{i+1} - \mathbf{z}^{i+1}\end{aligned} \tag{2.40}$$

Lze si povšimnout, že při minimalizování $L_\rho(\mathbf{x}, \mathbf{u}, \mathbf{z})$ přes $\mathbf{x}$ v prvním kroku bylo možné vynechat člen $g(\mathbf{z})$. Tento člen totiž není závislý na $\mathbf{x}$, proto nehraje žádnou roli při hledání argumentu minima $L_\rho(\mathbf{x}, \mathbf{u}, \mathbf{z})$. To samé platí pro $\frac{\rho}{2}\|\mathbf{u}\|_2^2$, získávaný stejnou cestou.[23]

2.9 Robust Principal Component Analysis

Za předpokladu, že máme matici dat, která je superpozicí nízko hodnostní složky a řídké složky, které je možné za určitých předpokladů obnovit, pomocí vhodného konvexního programu zvaného Analýza hlavních komponent (PCP). [5]

Předpokládáme-li, že máme velkou datovou matici $\mathbf{M}$, která je složena z:

$$M = L_0 + S_0, \tag{2.41}$$

kde L_0 je nízkohodnostní a S_0 je řídká složka, a obě jsou libovolné velikosti. Neznáme nízkohodnostní sloupeček a řádek z L_0, ani jejich dimenze. Stejně tak neznáme umístění nenulových položek z S_0, ani kolik jich je. Prokazatelné správné a škálovatelné řešení výše popsaného problému by mělo pravděpodobně dopad na proces vědeckého objevování, který je náročný zejména na data. Nedávný boom vysoce-rozměrných dat ve vědě, představuje výzvu pro mnoho oblastí, jako je obrazové, video a multimediální zpracování, analýza webových stránek, vyhledávání, biomedicinského zobrazování a bioinformatiky. V takovýchto aplikačních doménách je běžné, že obsahují tisíce nebo někdy až miliardy dimenzí, s počtem vzorků někdy o stejných velikostech.

Analýza hlavních komponent (PCA) je pravděpodobně nejrozšířenějším statistickým nástrojem pro analýzu dat a redukce dimenzí dnešní doby. Nicméně, jeho křehkost s ohledem na hrubě poškozená pozorování však často ohrožuje jejich platnost - jedna hrubě poškozená položka v matici $\mathbf{M}$ by mohla znamenat odhadovanou $\hat{L}$ libovolně daleko od správného L_0. Bohužel hrubé chyby jsou nyní všudy přítomné v moderních aplikacích (zpracování obrazu, webová analýza dat nebo bioinformatika), kde některá měření mohou být libovolně poškozena (špatnou manipulací s daty nebo selháním senzoru) nebo jednoduše irelevantní k nízko hodnostní struktuře, kterou se snažíme identifikovat. V průběhu několika dekád byly zkoumány a

navrženy řady přístupů. Ať už techniky vlivové funkce, střídavé minimalizace nebo náhodného vzorkování. Žádný z nich však bohužel neposkytuje polynomiální algoritmus se silnými zárukami výkonu za širokých podmínek. Robustní analýza hlavních komponent se snaží obnovit matici L_0 s nízkou hodností z vysoce poškozených dat $M = S_0 + L_0$, na rozdíl od malého šumového výrazu N_0 v klasické analýze hlavních komponent.

Existuje mnoho důležitých oblastí a aplikací, ve kterých lze studovaná data přirozeně modelovat jako nízko hodnostní plus řídké. Všechny tyto statistické aplikace, hledající robustní hlavní komponenty, samozřejmě odpovídají tomuto modelu. Uvedené příklady jsou inspirované současnými výzvami v oblasti informatiky, buď je objektem zájmu nízko hodnostní složka nebo naopak řídká složka.

- **Video**. Vzhledem k sekvenci pozorovaných video snímků, často potřebujeme identifikovat činnost, která vyčnívá z pozadí. Pokud vektorizujeme jednotlivé video snímky jako sloupečky matice $\mathbf{M}$, pak nízko hodnostní složka L_0 přirozeně koresponduje se statickým pozadí a řídká složka S_0 zachycuje pohyblivé objekty v popředí. Každý jednotlivý snímek však obsahuje tisíce nebo desítky tisíc pixelů a každý fragment videa obsahuje stovky nebo tisíce snímků. Rozložení matice $\mathbf{M}$ takovýmto způsobem by nebylo možné pokud bychom neměli skutečně škálovatelné řešení.
- **Rozpoznávání obličejů**. Dle[5] je známo, že obrazy konvexního, Lambertovského povrchu, pod měnícím se osvětlením, zahrnují nízko hodnostní podprostor. Tato skutečnost byla hlavním důvodem, proč jsou nízko hodnostní modely vetšinou efektivní pro snímková data. Zejména obrazy lidské tváře mohou být dobře aproximovány nízko hodnostním podprostorem. Schopnost správně načíst tento podprostor je zásadní v mnoha aplikacích jako je rozpoznávání tváří. Ačkoli reálné obrazy obličeje často trpí samo stínováním, zvláštnostmi nebo sytostí jasu, což činí tento úkol obtížným.
- **Latentně sémantické indexování**. Webové vyhledávače často potřebují analyzovat a indexovat obsah obrovského souboru dokumentů. Populární schéma je právě LSI, jehož základní myšlenkou je shromáždit dokument versus termín matici $\mathbf{M}$, jejíž položky typicky kódují význam výrazu nebo slova pro dokument. PCA (nebo SVD) tradičně rozloží matici jako nízko hodnostní část plus reziduum, které není nezbytně řídké, tak jak bychom chtěli. Pokud by bylo možné rozložit matici $\mathbf{M}$ jako součet nízko hodnostní složky L_0 a řídké složky S_0, pak L_0 by mohla zachytit běžná slova používaná ve všech dokumentech, zatímco S_0 zachytí několik klíčových slov, která nejlépe odlišují jednotlivé dokumenty od ostatních
- **Filtrování hodnocení a spolupráce**. Problém předvídání vkusu uživatelů nabývá rostoucího významu u online obchodů a reklamy. Firmy sbírají hod-

nocení uživatelů pro různé produkty, např. filmy, knihy, hry nebo webové nástroje. Problémem bývá použití neúplných hodnocení poskytnutých uživateli na některé produkty, aby předpověděli preferenci uživatele na dané služby. Vzhledem k tomu, že proces shromažďování údajů často nemá žádnou kontrolu, je náročnější, protože musíme současně dokončit a opravit chyby. Což znamená, že musíme odvodit nízko hodnostní složku L_0 od neúplných a poškozených položek.

Podobné problémy mohou nastat v mnoha jiných aplikacích, jako je například grafický model učení, lineární systém identifikace nebo rozklad v optických systémech. V některých aplikacích sledování objektů může být využíváno naopak pouze řídké složky S_0, jako například trajektorie míčku při tenisovém zápase. [5]

3 ODDĚLENÍ POHYBUJÍCÍCH SE OBJEKTŮ OD POZADÍ POMOCÍ MEDIÁNU

Pro oddělení pozadí od pohybujícího se popředí bude využito programu Matlab. Díky některým implementovaných funkcím v image processing toolboxu můžeme ověřit mediánovou metodu oddělení pozadí, představenou v úvodní kapitole, a vytvořit novou funkci, která bude pracovat stejně jako mediánová statistika v programu Adobe Photoshop. S tím rozdílem, že program Photoshop od firmy Adobe zpracovává pouze načtené jednotlivé snímky z video sekvence. Prostředí Matlabu nám umožní načíst celý videosoubor a zpracovat jej jako celek bez „vystřihování“ jednotlivých framů/snímků, tak jak to bylo nutné v Adobe PS. Byla vytvořena funkce *medianmethod.m*, kterou lze vyvolat pomocí příkazové řádky v Matlabu.

V hlavičce funkce je základní popis této funkce a v několika krocích popsán její postup. Příkazem $info\ medianmethod.m$ lze tyto základní informace o funkci vyvolat. Pro spuštění funkce není potřeba zadávání žádných vstupních parametrů, funkce je implementována tak, že ji stačí vyvolat. Po vyvolání z příkazové řádky se otevře GUI programu, které vyžaduje otevření požadovaného video souboru a jeho umístění v PC.

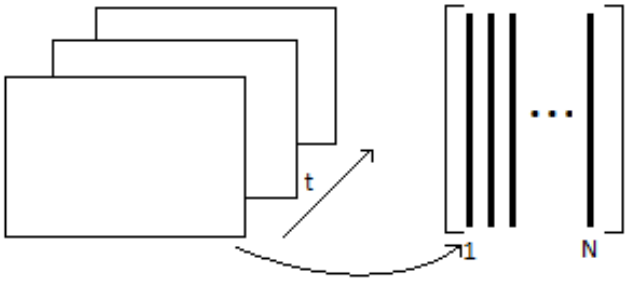
3.1 Funkce pro převod z RGB do stupňů šedi

Vzhledem k tomu, že řešení úlohy bylo uvažováno pro video soubory ve stupních šedi, bylo potřeba natočené testovací video sekvence převézt. K tomuto účelu byla vytvořena funkce *prevod.m*. Která se po načtení z příkazové řádky pomocí uživatelského prostředí dotáže na vybrání požadovaného souboru, který po načtení přehraje. Poté video převede do požadovaných stupňů šedi a uloží nově konvertované video do kořenové složky s názvem $NewVid.avi$. A před končením programu jej také přehraje.

3.2 Načtení video souboru

Nejdůležitější částí programu je načtení video souboru. Načtení je věnována samostatná sekce, jelikož je využité i při hledání řídkého řešení.

Po vyvolání funkce, načtení libovolného video souboru pomocí uživatelského rozhraní, dojde ke čtení a zapsání video souboru a jeho přehrání. Obr.(3.1) blíže ukazuje zápis videa. Jak již bylo několikrát řečeno, video je sekvencí snímků jdoucích rychle za sebou. Program si v první fázi zapsání video souboru vytvoří prázdnou matici $\mathbf{M}$, do které zapisuje jednotlivé framy/snímky videa. Po vytvoření matice, si do zvolené proměnné načte program jednotlivé framy/snímky videa, které potom zapisuje



Obr. 3.1: Demonstrace zápisu video souboru

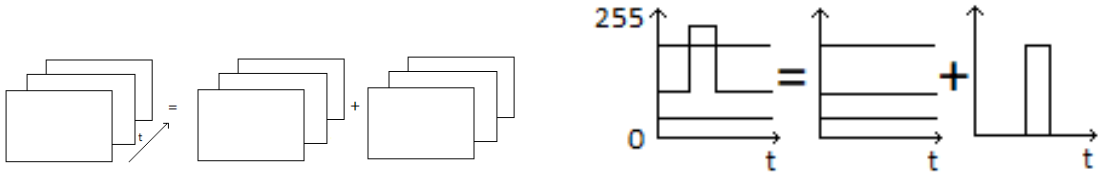
.

do prázdné matice **M**. Jak ukazuje i obrázek, jednotlivé snímky představují sloupce matice **M**, dochází tedy k vektorizaci jednotlivých snímků. Výsledná matice **M** tedy obsahuje 1 až N počet sloupců, kde N je celkový počet snímků videa. Konkrétně, jestliže jsme načítali konkrétní video soubor s rozlišením 480p, kde rozlišení jednoho snímku je 480×854, při snímkové rychlosti $25 fps$ a celkového počtu snímku 65. Při zapsání jednotlivých snímků jako sloupců do matice **M** je výsledná velikost matice $M^{409920\times65}$. Toto platí, pokud bude video ve stupních šedi, tedy jeden snímek je reprezentován 1-D maticí. Pokud by byl snímek reprezentován barevným prostorem RGB, měl by trojnásobnou velikost, jelikož každý jednotlivý snímek by obsahoval tři složky r, g a b o velikosti 480×854.

Během plnění matice **M** videosnímky se vykreslí graf obrázku s škálou barev, dle hodnot jednotlivých pixelů. Na tomto grafu budou vidět vykreslené pohyblivé objekty na neměnném pozadí. Před samotným výpočtem je ještě důležitým krokem zjištění velikosti z proměnné načítající jednotlivé snímky, díky čemuž bude možné načíst video s jakýmkoliv rozlišením a exportovat výsledek právě dle vstupního video souboru.

3.3 Výpočet mediánu

Načtené video snímky lze s trochou představivosti uvažovat jako součet snímků s neměnným pozadím a snímky pouze s pohybujícími se objekty na černém pozadí (což je způsobeno nulovými hodnotami). Jak demonstruje obr.(3.2). Samotný výpočet mediánu (tj. prostřední hodnoty) pomocí funkce median(), která je v matlabu již definovaná. Je nastaven tak, aby získal medián po řádcích, čímž porovná každý jednotlivý pixel snímků jdoucích za sebou. Po získání mediánu máme tedy vektory o velikosti jednoho sloupečku původní matice **M**. Jednotlivé hodnoty tohoto vektoru představují jednotlivé mediány po řádcích, neboli prostřední „neměnné“ hodnoty.



(a) (b)

Obr. 3.2: Představa načtených video snímků

Tento vektor nám představuje již získaný snímek neměnného pozadí, pro jeho zobrazení je potřeba jej „přeskládat“/odvektorizovat zpět do původní velikosti snímku videa, k čemuž využijeme funkci reshape().

Abychom však mohli vektor mediánu odečíst od původní matice **M** a získat tím pohyblivé popředí, je zapotřebí tento vektor (tj. jeden snímek) naskládat za sebe do stejného počtu snímků/sloupců, jako má matice **M**, k čemuž lze využít funkce repmat(). V této chvíli odečteme od sebe dvě matice stejného rozměru. Od matici vstupní odečteme získané pozadí, čímž získáme novou matici, kde většina hodnot bude nulových, či blízkých nule a větších hodnot budou nabývat pouze pohyblivé objekty.

Výstupními soubory jsou tedy obrázek separovaného pozadí, což je zmiňovaný medián. Funkce se nás opět pomocí uživatelského rozhraní zeptá na název a místo uložení tohoto obrázku a následně jej zobrazí. Pokud bychom chtěli uložit místo obrázku pozadí video snímek neměnného pozadí, lze jednoduše využít doplněné matice funkcí repmat(). V odevzdaných funkcích je uložení této video sekvence zakomentováno, stačí jej pouze odkomentovat a lze uložit i video pozadí. Druhým výstupním souborem je video sekvence proměnného popředí, kdy se nás opět funkce zeptá na název a uložení, poté se video také přehraje.

3.4 Výsledky

Pomocí mediánové statistiky bylo dosaženo dobrých výsledků. Na obr.(3.3) jsou výsledná dvě pozadí získaná pomocí mediánové metody. První z nich budovy VUT FEKT na adrese Technická 10 a druhá na Technická 12. Mírně nepříjemný „rozmazaný“ efekt vykazuje první pozadí (a), uprostřed snímku, což ovšem není způsobeno použitou metodou, ale odrazem obrazu od oken budovy, což lze vidět i na pravé části budovy.

(a) Budova VUT FEKT Technická 10

(b) Budova VUT FEKT Technická 12

Obr. 3.3: Výsledná pozadí získaná pomocí mediánové metody

3.5 Testovací video soubory

Pro potřeby testování byly vytvořeny dva krátké video soubory. Oba dva video soubory byly demonstrativně pořízeny před budovami VUT, Fakulty elektrotechniky a komunikačních technologií, první z nich na Technické 12 a druhé na Technické 10. Jedno z videí obsahuje procházející lidi před budovou a druhé taktéž, navíc s projíždějícím automobilem.

Obě videa byla pořízena digitální zrcadlovkou Nikon D7500, v rozlišení 720p, se snímkovou rychlostí 30fps. Videa byla následně zpracována v programu Adobe Premiere, který je komerčně využíván pro práci s videem. Videa byla sestřižena na necelé 3 vteřinové sekvence a pro potřeby testování byla zmenšena („downscalována“) na rozlišení 480p a rozlišení 284x160 se snímkovou rychlostí 25fps, z důvodu pozdějšího rychlejšího zpracování. Obě tyto rozlišení jsou uložena v příloze.

4 ODDĚLENÍ POHYBUJÍCÍCH SE OBJEKTŮ OD POZADÍ POMOCÍ ŘÍDKÝCH REPREZENTACÍ

Pro řešení pomocí analýzy hlavních komponent PCP vypadá úloha takto:

$$minmalizovat \ \|L\|_* + \lambda \|S\|_1 \ vzhledem \ k \ L + S = M, \tag{4.1}$$

kde $\|\cdot\|_*$ je nukleární norma, která nám zaručuje nízkou hodnost a $\|\cdot\|_1$ je l_1-norma zaručující řídkost. Z důvodů špatné škálovatelnosti dřívějších řešení pomocí metod vnitřních bodů, bylo dle [5] vyvinuto řešení, které bylo inspirováno metodami prvního řádu a využitím analogie s iteračními prahovacími algoritmy pro l_1-minimalizaci. Vznikl algoritmus, který minimalizuje nukleární normu opakovaným měkkým prahováním singulárních hodnot vhodné matice, která snižuje složitost při každé iteraci.

Pokud k řešení využijeme ADMM metodu (4.2) s Augmented Lagragianem, pro odvození dvou posledních členů, kde $r = M - L - S$ a $u = \mathbf{Y}\rho$, potom:

$$\begin{aligned} &\frac{\rho}{2}\left\|\mathbf{M}-\mathbf{L}-\mathbf{S}+\rho^{-1}\mathbf{Y}\right\|_2^2 - \frac{1}{2\rho}\|\mathbf{Y}\|_2^2 = \\ &= \frac{\rho}{2}\left\langle \mathbf{M}-\mathbf{L}-\mathbf{S}+\rho^{-1}\mathbf{Y}, \mathbf{M}-\mathbf{L}-\mathbf{S}+\rho^{-1}\mathbf{Y}\right\rangle - \frac{1}{2\rho}\langle \mathbf{Y},\mathbf{Y}\rangle = \\ &= \frac{\rho}{2}\|\mathbf{M}-\mathbf{L}-\mathbf{S}\|_2^2 + \langle \mathbf{Y}, \mathbf{M}-\mathbf{L}-\mathbf{S}\rangle . \end{aligned} \tag{4.2}$$

Potom celkově odpovídá

$$L_\rho(\mathbf{L},\mathbf{S},\mathbf{Y}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle \mathbf{Y}, \mathbf{M}-\mathbf{L}-\mathbf{S}\rangle + \frac{\rho}{2}\|\mathbf{M}-\mathbf{L}-\mathbf{S}\|_F^2 . \tag{4.3}$$

Obecný Lagrangerův algoritmus by vyřešil PCP opakovaným nastavováním $(\mathbf{L}_i, \mathbf{S}_i) = arg\min_{L,S} l(\mathbf{L},\mathbf{S},\mathbf{Y}_i$ a poté aktualizoval Lagrangeovu matici $\mathbf{Y}_{i+1} = \mathbf{Y}_k + \rho(\mathbf{M} - \mathbf{L}_k - \mathbf{S}_k)$.

Pro nalezení nízko hodnostního a řídkého řešení, namísto konvexní optimalizace, můžeme využít proximálních operátorů pro nuklární normu a l_1-mormu. Kde $S_\tau : \mathbb{R} \to \mathbb{R}$ značí operátor měkkého prahování $S_\tau[x] = sgn(x)max(|x| - \tau, 0)$a aplikuje se na každý prvek matice, minimalizací podle S získáme

$$arg\min_S l(\mathbf{L},\mathbf{S},\mathbf{Y}) = S_{\lambda/\rho}(\mathbf{M} - \mathbf{L} + \rho^{-1}\mathbf{Y}). \tag{4.4}$$

Obdobně pro matici $\mathbf{X}$, $D_\tau(\mathbf{X})$ značí měkké prahování pro singulární hodnoty daný $D_\tau(\mathbf{X}) = \mathbf{U}S_\lambda(\sum)\mathbf{V}^*$, kde $X = \mathbf{U}\sum\mathbf{V}^*$ je SVD

$$arg\min_L l(\mathbf{L},\mathbf{S},\mathbf{Y}) = D_{1/\rho}(\mathbf{M} - \mathbf{S} + \rho^{-1}\mathbf{Y}). \tag{4.5}$$

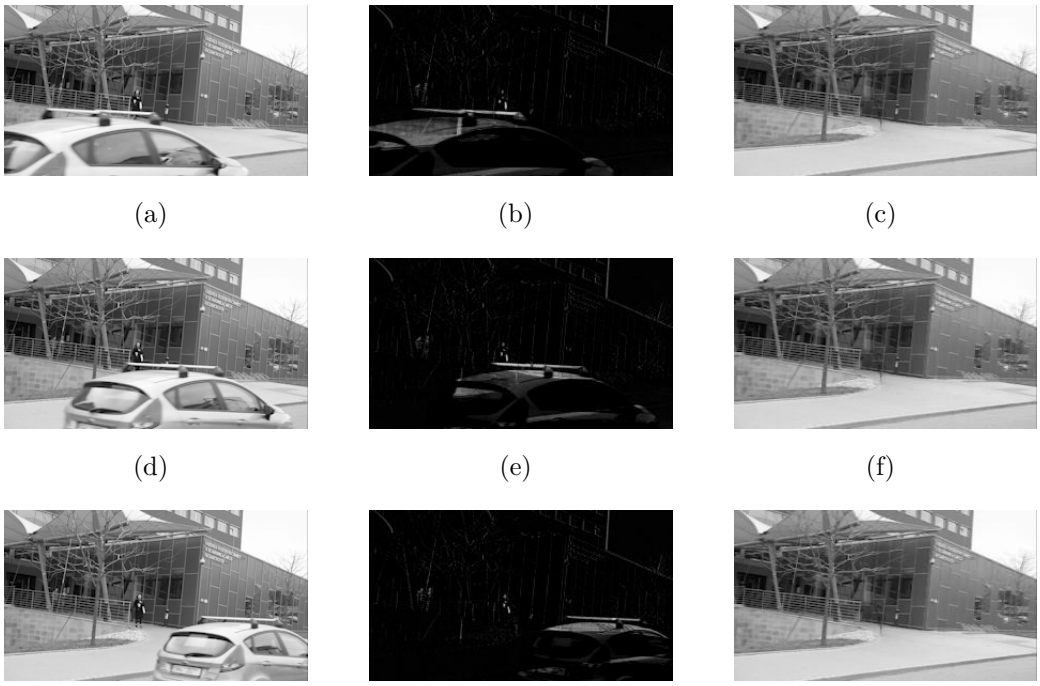
Jeví se jako praktické minimalizovat l vzhledem k $\mathbf{L}$(vázané na $\mathbf{S}$), poté minimalizovat l vzhledem k $\mathbf{S}$(vázané na $\mathbf{L}$), nakonec aktualizovat Lagrangeovu multiplikační matici $\mathbf{Y}$ na základě $\mathbf{M} - \mathbf{L} - \mathbf{S}$. Algoritmus pracuje v následujících krocích:

1. inicializace $S_0 = Y_0 = 0, \rho > 0$
2. pokud nekonverguje
3. počítá $\mathbf{L}_{i+i} = D_{1/\rho}(\mathbf{M} - \mathbf{S} + \rho^{-1}\mathbf{Y})$;
4. počítá $\mathbf{S}_{i+i} = S_{\lambda/\rho}(\mathbf{M} - \mathbf{L} + \rho^{-1}\mathbf{Y})$;
5. počítá $\mathbf{Y}_{i+1} = \mathbf{Y}_k + \rho(\mathbf{M} - \mathbf{L}_{i+1} - \mathbf{S}_{i+1}$;
6. konec cyklu
7. výstup **L**, **S**.

Nejdůležitějšími detaily implementace algoritmu je volba ρ a kritérium zastavení.

Funkce *pcp.m* načítá video soubory stejným způsobem, jako bylo popsáno v předchozí kapitole. Pomocí uživatelského rozhraní se dotáže na načtení souboru, který je obdobným způsobem převeden do matice **M**. Před samotným výpočtem je definice proměnných, konkrétně váhovací parametr λ, počáteční a maximální počet iterací, podmínka konvergence, Frobeniovu normu matice **M**, z důvodů výpočtu parametru ρ. Po samotném cyklu výpočtu proběhne uložení **L** a **S** složky do pracovního adresáře programu, tentokrát již bez uživatelského prostředí. Parametr λ byl nastaven na hodnotu $\lambda = 1/\sqrt{max(m,n)}$, dle doporučení článku[5], taktéž podmínka konvergence $error = 1e^{-}7$. Nastavení těchto dvou parametrů má zásadní vliv na celkový výsledek.

Na snímcích níže již můžeme vidět výsledek pro dvě testovaná videa. V levém sloupci jsou původní snímky video souboru, v prostředním pak řídká složka popředí a v pravém sloupci nízko hodnostní složka pozadí.[5]

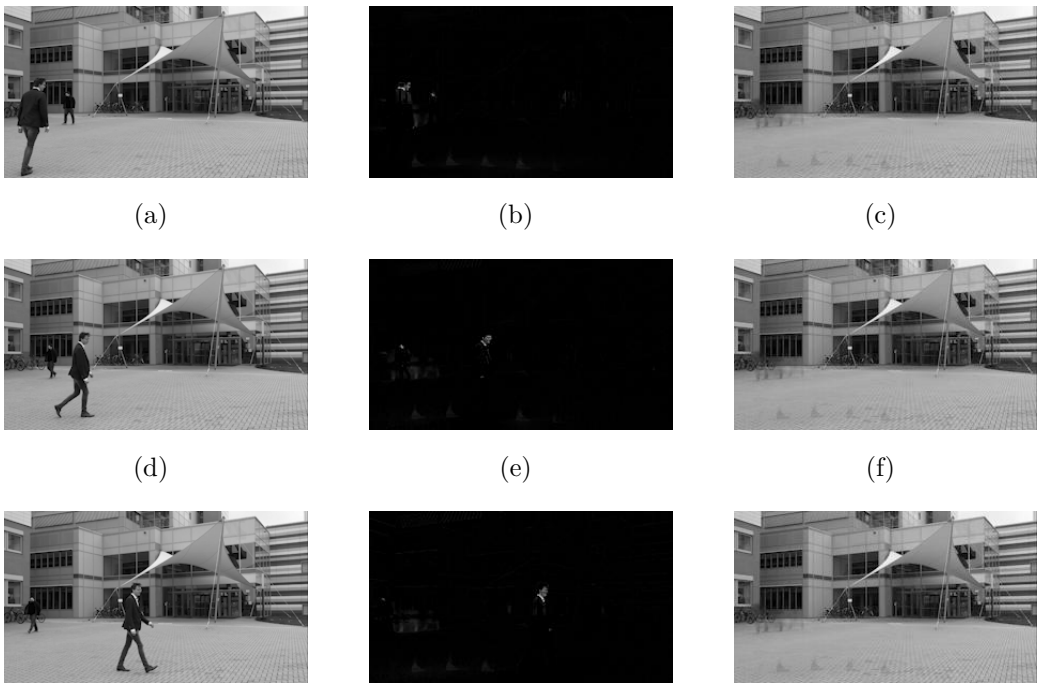


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Obr. 4.1: Výsledná sekvence snímků pomocí PCP



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Obr. 4.2: Výsledná sekvence snímků pomocí PCP

5 POROVNÁNÍ A VYHODNOCENÍ

Metoda separace pomocí mediánu pracuje vcelku rychle a pro námi testovaná videa dala velmi uspokojivý výsledek, během pár sekund. Samozřejmě výpočet pro šedotónové video trval kratší dobu než pro barevné. K této metodě je nutné dodat zásadní poznámku, metoda fungovala správně, protože jsme měli k dispozici dvě testovaná videa s ne příliš zarušenými daty, pokud bychom měli mnohem více pohyblivých objektů s ještě například měnící trajektorií, metoda by správně nefungovala. Tento fakt je dán samotnou podstatou metody. Jelikož medián porovnává jednotlivé pixely snímků jdoucích za sebou a vybírá prostřední hodnotu z řady hodnot. Pokud bychom měli velmi mnoho špičkových hodnot, pozorovali bychom na výsledných separovaných pozadích stejné artefakty, jako jsou na snímcích v úvodní kapitole, při separaci pomocí Adobe PS, který neměl pro svůj výpočet dostatek snímků.

U PCP metody hrála obrovskou roli volba parametrů, především podmínky konvergence *error* a λ. Tato volba měla zásadní vliv nejen na výsledek separace, ale také počtu iterací a době zpracování. Při nastavení hodnoty $error = 1e^{-}7$ a $\lambda = 1/\sqrt{(10 * max(m,n))}$ se dosáhlo dobrého výsledku pozadí bez artefaktů, nicméně počet iterací byl na maximální hodnotě (1000). Při snížení ρ (resp. zvětšení) na hodnotu $error = 1e^{-}5$ a $\lambda = 1/\sqrt{(10 * max(m,n))}$ bylo dosaženo výsledku za 178 a v druhém případě za 279 iterací, ovšem na výsledných separovaných snímcích byly již pozorovatelné velmi mírné artefakty. Pro ověření byla nastavena ještě jedna extrémní hodnota $error = 1e^{-}4$ a $\lambda = 1/\sqrt{(10 * max(m,n))}$, kdy byl výsledek zpracován během 68 iterací, nicméně s poměrně rozsáhlými artefakty. Nejlepší výsledek, co do doby trvání a kvality výstupu, byl při nastavení $error = 1e^{-}6$ a $\lambda = 1/\sqrt{(10 * max(m,n))}$, což proběhlo během 686 iterací.

6 ZÁVĚR

V této diplomové práci bylo dosaženo kladných výsledků, podařilo se separovat pozadí z narušené video sekvence, jak klasickou metodou, tak pomocí řídkého zpracování.

Pro samotné testování metod byla vytvořena dvě různá videa. Obě s cílem demonstrovat podstatu separace. Na snímcích byly zachyceny dvě budovy, před kterými se vyskytovaly rušivé elementy, jako procházející lidé nebo projíždějící automobil.

Tyto video soubory byly použity na vysvětlení oddělení pozadí od popředí v úvodní kapitole, především však pro mediánovou statistiku v programu Adobe Photoshop, která pak následně v návaznosti k této části byla vytvořena a implementována v programu Matlab.

Byla vytvořena mediánová metoda pro separaci pozadí ze zarušené video sekvence. Pro potřeby zpracování videa byla vytvořena funkce pro převod videa z RGB do šedotónových složek, která je také součástí práce. Později však byla tato metoda implementována i pro RGB videa, bez nutnosti převodu, i když tento krok si zadání nevyžadovalo. Součástí práce a v přílohách je uvedena funkce *medianmethod.m* a *medianmethodRGB.m*, což se ve výsledku jeví jako zbytečné, protože druhá ze jmenovaných načte video soubory o libovolném rozlišení, jak RGB videa, tak šedotónová videa. Její funkčnost a využitelnost se ukazuje totožná a shodná se statistikou v programu Adobe PS, což ji činí atraktivním, více zajímavé by možná bylo vytvoření podobné funkce jako samostatnou aplikaci např. do chytrých telefonů, případně vylepšit rychlost výpočtu využitím nejen CPU, ale i GPU. Mediánová metoda je velmi jednoduchá, lehce implementovatelná, její výpočetní čas není taktéž velice náročný, avšak nehodila by se pro video sekvence s velkým počtem skokových hodnot.

Druhou metodou separace, využitím řídkého zpracování signálů, byla vytvořena aplikace PCA, která dává také uspokojivé výsledky, ačkoliv za cenu delšího času zpracování. Bylo ověřeno, že nastavením zásadních parametrů se může velice lišit celková doba výpočtu, počtu iterací, ale také výsledné separace. Což platí i pro předchozí metodu. Pokud se separace nezdařila, pozorovali jsme na separovaných pozadích artefakty - „duchy“, či rozmazaná nebo tmavší místa.

Pro testování a ověření funkčnosti metod byly vytvořeny dvě krátké video sekvence. Drobným nedostatkem pro samotné testování byl ten, že oba snímky byly podobného zarušení. Pokud by byl k dispozici jeden videosnímek, který by byl oproti druhému extrémněji zarušen, mohla by práce přinést některé další poznatky.

Součástí elektronických příloh nejsou všechna videa, z důvodu přesahu velikosti. Je zde přiložen soubor s textem kde je odkaz na cloudové úložiště, kde jsou veškeré soubory, které jsou na přiloženém CD u tištěné verze.

LITERATURA

[1] TEKALP, M. A.: *Digital Video Processing.* University of Rochester, 1995. Dostupné z URL: https://www.researchgate.net/publication/200132428_Digital_Video_Processing

[2] KEITH, J.: *Video Demystified: A Handbook for the Digital Engineer* 4th edition, 2005, ISBN: 0-7506-7822-4

[3] RAJMIC, P., Daňková, M.: *Úvod do řídkých reprezentací signálů a komprimovaného snímání.* Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 68 s.

[4] CANDES, E. J., WAKIN, M. B.: *An introduction to cmpressive sampling.* IEEE Signal Processing Magazine, ročník 25, č.2, 2008. s. 21-30., ISSN 1053-5888.

[5] CANDES, E. J., LI, X., MA, Y., WTIGHT, J.: *Robust principal component analysis?* J. ACM 58, 3, Article 11(May 2011), 37 s. Dostupné z URL: https://dl.acm.org/citation.cfm?doid=1970392.1970395

[6] MANGOVÁ, M.: *Zvýšení rozlišení perfúzního zobrazování magnetickou rezonancí pomocí komprimovaného snímání.* Brno, 2018, 104s Dizertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

[7] ZÁVIŠKA, P., MOKRÝ, O., RAJMIC, P.: *S-SPACE Done Right: Detailed Study of the Sparse Audio Declipper Algorithms.* October 31, 2018. Dostupné z URL: https://arxiv.org/pdf/1809.09847.pdf

[8] Wikipedia [online]: https://cs.wikipedia.org/wiki/Digit%C3%A1ln%C3%AD_film

[9] GONZALEZ, C. R., WOODS, E. R.: *Digital Image Processing.* 2nd edition, University of Tennessee, MedData Interactive, Dostupné z URL: http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/Digital_Image_Processing_2ndEd.pdf

[10] BAŘINA, D.: *Videokodek - komprese videosekvencí.* Brno: Vysoké učení technické v Brně, Fakulta informačních technologií, 2009. 58 s. Vedoucí diplomové práce doc. RNDr. Pavel Smrž, Ph.D., Dostupné z URL: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=116858

[11] Wikipedia [online]: https://en.wikipedia.org/wiki/Video

[12] EASTON, L. R. Jr.: *Fundamentals of Digital Image Processing.* 22 November 2010, Dostupné z URL: https://www.cis.rit.edu/class/simg361/Notes_11222010.pdf

[13] Teorie digitálního videa [online]: http://bitgoo.cz/teorie-digitalniho-videa.php

[14] OUJEZDSKÝ, D.: *Digitální video.* Ostrava: Vysoká škola báňská - Technická universita Ostrava, 2011. 76s.http://bitgoo.cz/teorie-digitalniho-videa.php

[15] Wikipedia [online]: https://cs.wikipedia.org/wiki/Lidsk%C3%A9_oko

[16] Encyklopedie fyziky [online]: http://fyzika.jreichl.com/main.article/view/719-zareni-absolutne-cerneho-telesa

[17] CHADIM, P.: *Detekce pohybu ve video sekvenci.* Brno: Vysoké učení technické v Brně, Fakulta informačních technologií, 2008. 41 s. Vedoucí diplomové práce ing. Miroslav Švub, Dostupné z URL: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=116195

[18] DAŇKOVÁ, M.: *Aplikace lineární algebry a optimalizace ve zpracování obrazů.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 35 s. Vedoucí bakalářské práce Mgr. Pavel Rajmic, Ph.D.

[19] RAJMIC, P.: *Základy počítačové sazby a grafiky.* Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 161 s.

[20] BENEZETH, Y., JODOIN, M. P., EMILE, B., LAURENT, H., ROSENBERGER, CH.: *Comparative study of background substraction algorithms.* Journal of Electronic Imaging, SPIE and IST, 2010, 31s, 19, 10.1117/1.3456695. inria-00545478 Dostupné z URL: https://hal.inria.fr/inria-00545478/document

[21] BOYD, S. P., VANDENBERGHE, L.: *Convex Optimization.* Cambridge University Press, 2004, ISBN 0521833787. Dostupné z URL: https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

[22] Wikipedia [online]: DSLR https://en.wikipedia.org/wiki/Digital_single-lens_reflex_camera

[23] ZÁVIŠKA, P., MOKRÝ, O., RAJMIC, P.: *S-SPADE Done Right: Detailed Study of the Sparse Audio Declipper Algorithms.* Brno: University of Technology, May 7, 2019, Dostupné z URL: https://arxiv.org/pdf/1809.09847.pdf

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

HTC	High Tech Computer Corporation
HDCAM	High-definition video digital recording videocassette
CCD	Charge-coupled device
PAL	phase alternating line
SECAM	Séquentiel couleur à mémoire
NTSC	National Television System(s) Committee
CRT	Cathode Ray Tude
bps	bits per second
CBR	Constant Bitrate
VBR	Variable Bitrate
PSNR	Peak signal-to-noise ratio
ITU-R	International Telecommunication Union - Radiocommunication Sector
A/D	Analog to Digital
DC	Direct current
DC	Digital count
RGB	Red Green Blue
YIQ	Y - luma, I,Q - chrominance information
YUV	Y - jasová složka, U,V - barevné složky
YCbCr	Y - luma, Cb,Cr - blue/red-difference chroma
HSI	Hue Saturation Intensity
HSV	Hue Saturation Value
DSLR	Digital single-lens reflex camera
VST	Visual Studio Technology
VLC	VLC Media Player
PC	Personal Computer
NSP	Null Space Property
CS	Compressed sensing/compressive sampling
SVD	Single Value Decomposition
IPM	Interior-point method
MP	Matching Pursuit
OMP	Ortoghonal Matching Pursuit
BP	Basic Pursuit
LARS	Least Angle Regresion, homotopy method
IRLS	Iterative Reweighted Least Square
FOCUS	ROCal Underdetermined System Solver
OMP	Orthogonal Matching Pursuit
PS	Photoshop

fps	frame per seconds
GUI	Graphical User Interface/Grafické uživatelské rozhraní
VUT	Vysoké učení technické v Brně
Fekt	Fakulta elektrotechniky a komunikačních technologií
RPCPA	Robust Principal Component Pursuit Analysis
PCA	Principal Component Analysis
LSI	Latent Semantic Indexing
ADMM	Alternating Direction Method of Multipliers
CPU	Central Processing unit
GPU	Graphics processing unit

SEZNAM PŘÍLOH

A ZDROJOVÉ KÓDY

A.1 Převod RGB do stupňů šedi

```
1  function [filename, pathname] = prevod()
2  %function PREVOD is for conversion any loaded video from RGB to
       ↪ GRAYSCALE
3  %1. function asked you to load video via GUI, 2. make convert to GS
       ↪ , 3.
4  %fucnction ask you for new name and location for save
5
6  %% GUI for loading video file
7  % show supported video formats
8  formats = VideoReader.getFileFormats();
9  % specification of supported formats
10 filterSpec = getFilterSpec(formats);
11 % read video file via GUI
12 [filename,pathname] = uigetfile(filterSpec);
13 % read video
14 reader = VideoReader(filename);
15 % playing readed video
16 implay(filename)
17
18 %% converting to GrayScale and saving new converted video
19 writer = VideoWriter('NewVid.avi', 'Grayscale␣AVI');
20 writer.FrameRate = reader.FrameRate;
21 open(writer);
22 %reading all frames, convert to grayscale and save as a new
23 M = [];
24 while hasFrame(reader)
25     img = rgb2gray(readFrame(reader));
26     writeVideo(writer, img);
27     M = [M img(:)];
28 end
29 close(writer);
30 implay('NewVid.avi');
31 end
```

A.2 Mediánová metoda

```
1  function [frg, bgr,imgdata, filename, pathname] = medianmethod ()
2  %% Median method - for background substraction from video sequence
       ↪ with moving objects
3  %function is for separation static background picture from
       ↪ GRAYSCALE video sequence with
4  % moving objects via MEDIAN METHOD
5  % You can load any video with any resolution, the code is set for
       ↪ exporting
6  % in the same setup as INPUT data
7  % 1. loading video, 2. filling video to big matrix frame by frame,
       ↪ 3. count
8  % median and create matrix from it 4. Count L and S component, 5.
       ↪ saving
9  % video foreground with moving objects and picture of foreground
10 %% Importing video sequnce
11 formats = VideoReader.getFileFormats();
12 % specification of supported formats
13 filterSpec = getFilterSpec(formats);
14 %read video file via GUI
15 [filename,pathname] = uigetfile(filterSpec);
16 %read video
17 reader = VideoReader(filename);
18 % playing readed video
19 implay(filename)
20
21 %% reading video frames
22 M = [];
23 while hasFrame(reader)
24     img = readFrame(reader);
25     M = [M img(:)];
26     imagesc(M)
27 end
28 %size of img for exporting video and picture
29 [x,y,z]= size(img);
30 %size of M matrix for repeating median column
31 [m,n]=size(M);
32 %% median method for separation - counting
```

```
%median of each rows
Med = median(M,2);
% backgraound - L
% function repmat - creat matrix with median
bg = repmat(Med,1,n);
figure(1)
imagesc(bg)
% foreground - S
fo= M - bg;
figure(2)
imagesc(fo)

%% export only FOREGROUND
% specification for saving video via GUI
% show supported video formats
formats = VideoReader.getFileFormats();
% specification of supported formats
filterSpec = getFilterSpec(formats);
%write video file via GUI
[filename,pathname] = uiputfile(filterSpec);

vid = VideoWriter(filename);
open(vid)
[r,c]=size(fo);
for k = 1:c
    frg = reshape(fo(1:end,k), x, y);
    writeVideo(vid,frg);
end
close(vid)

%% export only Background picture
% writing photo file via GUI
[filename,pathname] = uiputfile('.jpg');
imgdata = reshape(Med, x, y);
imwrite(imgdata,filename);
%imfinfo('background.jpg');
imshow(imgdata);
end
```

A.3 Mediánová metoda RGB

```
1  function [frg, bgr,imgdata, filename, pathname] = medianmethodRGB ()
2  %% Median method - for background substraction from video sequence
       ↪ with moving objects
3  %function is for separation static background picture from video
       ↪ sequence with
4  % moving objects via MEDIAN METHOD
5  % You can load any video with any resolution, the code is set for
       ↪ exporting
6  % in the same setup as INPUT data
7  % 1. loading video, 2. filling video to big matrix frame by frame,
       ↪ 3. count
8  % median and create matrix from it 4. Count L and S component, 5.
       ↪ saving
9  % video foreground with moving objects and picture of foreground
10 %implemented in Matlab versi
11
12 %% Importing video sequnce
13 formats = VideoReader.getFileFormats();
14 % specification of supported formats
15 filterSpec = getFilterSpec(formats);
16 %read video file via GUI
17 [filename,pathname] = uigetfile(filterSpec);
18 %read video
19 reader = VideoReader(filename);
20 % playing readed video
21 implay(filename)
22
23 %% reading video frames
24 M = [];
25 while hasFrame(reader)
26     img = readFrame(reader);
27     M = [M img(:)];
28     imagesc(M)
29 end
30
31 %size of img for exporting video and picture
32 [x,y,z]= size(img);
```

```
33 %size of M matrix for repeating median column
34 [m,n]=size(M);
35
36 %% median method for separation
37 %median of each rows
38 Med = median(M,2);
39 % backgraound - L component
40 %create matrix with median
41 bg = repmat(Med,1,n);
42 figure(1)
43 imagesc(bg)
44 % foreground - S component
45 fo= M - bg;
46 figure(2)
47 imagesc(fo)
48
49 %% export only FOREGROUND
50 % specification for saving video via GUI
51 formats = VideoReader.getFileFormats();
52 % specification of supported formats
53 filterSpec = getFilterSpec(formats);
54 %write video file via GUI
55 [filename,pathname] = uiputfile(filterSpec);
56
57 vid = VideoWriter(filename);
58 open(vid)
59 [r,c]=size(fo);
60 for k = 1:c
61     frg = reshape(fo(:,k), x, y, z);
62     writeVideo(vid,frg);
63 end
64 close(vid)
65
66 %% export only Background picture
67 % writing photo file via GUI
68 [filename,pathname] = uiputfile('.jpg');
69 imgdata = reshape(Med, x, y, z);
70 imwrite(imgdata,filename);
71 %imfinfo('background.jpg');
```

```
imshow(imgdata);
end
```

A.4 Principal component analysis

```
function [S, L] = pcp ()
% min nuclear_norm(L) + lambda*||W(S)||_1 s.t. L+S=M

% show supported video formats
formats = VideoReader.getFileFormats();
% specification of supported formats
filterSpec = getFilterSpec(formats);
%read video file via GUI
[filename,pathname] = uigetfile(filterSpec);
%read video
reader = VideoReader(filename);
% playing readed video
implay(filename)

%new matrix for filling from video sequence
M = [];
while hasFrame(reader)
    img = double(readFrame(reader));
    M = [M img(:)];
    imagesc(M)
end
%size of loaded video for export
[x, y, z] = size(img);
whos M
[m,n]=size(M);

%% INPUTS parametrs
lambda = 1/sqrt(max(m,n));
% convergence condition
error = 1e-7;
% starting number of iterations
iter = 0;
% maximum number of iteration
```

```
34 iter_max = 1000;
35
36 S = zeros(m,n);
37 L = zeros(m,n);
38 Y = zeros(m,n);
39
40 % Frobenius norm of matrix
41 norm_m = norm(M, 'fro');
42 % choise of of parametr micro - treshold
43 mu = (n*n)/(4*norm_m);
44 invMu = 1/mu;
45
46 %algorythm
47 while true
48     iter = iter + 1;
49
50     % nuclear norm
51     L = Delta(invMu, M - S + (invMu * Y));
52     % sparsity l1 norm
53     S = es(lambda/mu, M - L + (invMu * Y));
54     %Lagrange multiplier matrix
55     Z = M - L - S;
56     Y = Y + mu*(Z);
57
58     disp((norm(Z, 'fro')/norm_m));
59
60     if ((norm(Z, 'fro')/norm_m) <= error) || (iter >= iter_max)
61
62         break;
63     end
64 end
65
66 pocet=iter
67 %% video writing
68
69 % S-comp Sparse component - moving objects
70 compS = VideoWriter('S_comp.avi');
71 open(compS)
72 for k = 1:n
```

```
73      frg = reshape(uint8(S(:,k)), x, y, z);
74      writeVideo(compS,frg);
75  end
76  close(compS)
77  implay('S_comp.avi')
78
79  %L-comp Low-rank component - static background
80  compL = VideoWriter('L_comp.avi');
81  open(compL)
82  for l = 1:n
83      bgr = reshape(uint8(L(:,n)), x, y, z);
84      writeVideo(compL,bgr);
85  end
86  close(compL)
87  implay('L_comp.avi')
88  end
89
90  %The Shrinkage operator - extend it to matrices by applying to each
        ↪ elemen
91  function [E] = es(tau, x)
92      E = sign(x) .* max(abs(x)-tau,0);
93  end
94  % Singular value thesholding operator - singular value
        ↪ decomposition
95  function [D] = Delta(t, x)
96      [U,S,V] = svd(x, 'econ');
97      D = U*es(t, S)*V';
98  end
```

B OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje funkci *prevod.m*, *medianmethod.m*, *medianmethodRGB.m* a *pcp.m*. Všechny funkce jsou spustitelné vyvoláním z příkazové řádky. Byly implementovány a ověřeny v programu Matlab ve verzi R2017b.

Složka $photoshop_separation$ obsahuje sekvenci snímků a pozadí zpracovaných pomocí Adobe Photoshop. Složka $median_vysledky$ obsahuje soubory získané pomocí funkce median. Složka $pcp_vysledky$ obsahuje výsledky dosažené pomocí funkce pcp. Složka *prevedenavidea* obsahuje video soubory prevede pomoci funkce prevod a složka *zdrojovavidea* obsahuje originální video soubory v rozlišení 480p a 284×160.