

## SEPARACE DYNAMICKÉ A STATICKÉ SLOŽKY V SÉRII OBRAZŮ

KAROLÍNA GEBRTOVÁ

**ABSTRAKT.** Článek se zabývá metodami separace statické a dynamické složky videa, respektive oddělení pohybujících se objektů od pozadí v sérii obrazů. První metoda je založena na použití mediánu, druhá metoda je formulována jako konvexní minimalizační úloha. Součástí článku je i předzpracování videa a vykreslení konečných výsledků, které může být problematické. Nakonec jsou metody porovnány dle jejich přesnosti separace a výpočetní náročnosti a dané výsledky jsou ilustrovány na reálných videích.

### 1. ÚVOD

Separace statické a dynamické složky v sérii obrazů se v dnešní době těší velkému množství užití. Aplikace nalezneme především v oblasti monitorovacích video-systémů, protože použitím těchto metod jsme schopni ve videu i přes nepřehledné pozadí detekovat jednotlivé objekty. Další aplikace leží v oblasti rozpoznávání obličejů, kde jsme díky separaci schopni odstranit stíny i jiná zkraslení (dynamickou složku) a získat tak čistý obraz obličeje [1].

Sérii obrazů můžeme reprezentovat jako video a problém separace ilustrovat na příkladu kamery snímající dálnici. Vozovka, svodidla a dopravní cedule zůstávají v průběhu času na stejném místě. Tvoří tedy v případě neměnných osvětlovacích podmínek pozadí reprezentující statickou složku. Projíždějící auta nebo letící ptáci svoji polohu s časem mění, reprezentují tedy dynamickou složku.

Tento článek bude zaměřen na metodu mediánového filtru, která je význačná svoji jednoduchostí implementace a rychlostí výpočtu, a metodu PCP (Principal Component Pursuit). Jedná se o konvexní úlohu založenou na minimalizaci vážené kombinace nukleární normy a  $\ell_1$  normy.

---

2010 MSC. Primární 47Nxx.

*Klíčová slova.* Série obrazů, separace pozadí, statická a dynamická složka, řídké reprezentace, nízkohodnostní struktura, singulární rozklad, mediánový filtr, principal component pursuit, robustní analýza hlavních komponent.

Článek vznikl na základě bakalářské práce autorky v oboru Matematické inženýrství na FSI VUT v Brně. Vedoucím práce byl Pavel Rajmic z Ústavu telekomunikací FEKT VUT v Brně.

## 2. ZÁKLADNÍ MATEMATICKÉ POJMY

Jednotlivé snímky videa se dají reprezentovat jako matice, bude tedy potřeba umět používat známé normy pro vektory i na matice. Toho lze docílit vektorizací matice: Jednotlivé sloupce matice „poskládáme pod sebe“ a získáme jeden dlouhý sloupcový vektor. Poté lze normu matice vypočítat jako  $\|\mathbf{X}\| = \|\text{vec } \mathbf{X}\|$ .

## 2.1. Řídké reprezentace

Nejprve zadefinujeme tzv.  $l_0$  normu vektoru, i když se dle definice o normu nejedná.

**Definice 2.1.**  $l_0$  normou vektoru  $\mathbf{x} \in \mathbb{C}^n$  rozumíme číslo udávající počet nenulových složek daného vektoru.

Vhodným uspořádáním jednotlivých pixelů lze zjistit, že se pozadí chová jako nízkohodnostní struktura, jejíž jedinou změnu působí dynamická složka. Pokud jsou objekty dynamické složky malé a není jich příliš mnoho, je tato změna navíc řídká. Řídkost definujeme následovně.

**Definice 2.2.** Vektor  $\mathbf{x} \in \mathbb{C}^n$  pro  $k \in \mathbb{N}$  nazveme  $k$ -řídkým, platí-li

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

Tedy  $k$ -řídký vektor má maximálně  $k$  nenulových složek. Analogicky pomocí vektorizace definujeme  $k$ -řídkou matici.

## 2.2. Singulární rozklad

**Definice 2.3** ([6]). Nechť  $\mathbf{A} \in \mathbb{C}^{m \times n}$  je obecná matice. Singulárním rozkladem (SVD rozkladem) matice  $\mathbf{A}$  rozumíme rozklad

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*,$$

kde  $\Sigma \in \mathbb{R}^{m \times n}$  je diagonální matice, jejíž diagonální prvky jsou tzv. singulární čísla  $\sigma_i$ . Tato čísla jsou nezáporná a seřazena podle velikosti od největšího. Tedy  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \geq 0$ , kde  $k = \min(m, n)$ . A matice  $\mathbf{U} \in \mathbb{C}^{m \times m}$ ,  $\mathbf{V} \in \mathbb{C}^{n \times n}$  jsou unitární matice tzv. levých a pravých singulárních vektorů,  $\mathbf{V}^*$  je hermitovská transpozice matice  $\mathbf{V}$ .

Singulární rozklad je velmi užitečný pro práci s nízkohodnostními strukturami, protože existují tvrzení (viz [3]) zajišťující, že singulární rozklad existuje pro každou matici  $\mathbf{A} \in \mathbb{C}^{m \times n}$  a zejména že počet nenulových singulárních čísel odpovídá hodnotě této matice.

## 3. METODY SEPARACE

Před uvedením použitých metod je nutné zmínit, v jaké formě jsou očekávána vstupní data.

### 3.1. Předzpracování dat

Vstupní data jsou očekávána ve formátu videa nebo jako série  $m \in \mathbb{N}$  po sobě jdoucích obrazů. V tomto článku se omezíme pouze na data ve stupních šedi.

Každý digitální obraz lze reprezentovat jako matici s rozměry odpovídajícími jeho rozlišení (počtu pixelů). Matici prvního snímku označíme  $\mathbf{M}_1$  a provedeme její vektorizaci. Výsledkem je jeden dlouhý sloupcový vektor reprezentující první snímek. Vektorizujeme i ostatní obrazy a získáme výslednou matici  $\mathbf{M}$ , jejíž sloupce tvoří vektory jednotlivých obrazů  $\mathbf{M} = [\text{vec } \mathbf{M}_1, \text{vec } \mathbf{M}_2, \dots, \text{vec } \mathbf{M}_m]$ .

Pro ilustraci úlohy separace si představme sérii obrazů se stále stejným pozadím a pohybující se malou kuličkou, která jako jediná působí změnu. Vytvoříme-li z daných obrazů matici  $\mathbf{M}$ , zjistíme, že jednotlivé sloupce  $\mathbf{M}$  jsou skoro stejné a liší se pouze v několika málo místech. Tento výsledek je očekávatelný, protože jednotlivé obrazy jsou až na posouvající se kuličku identické. Konkrétní polohu kuličky zjistíme z míst, kde se hodnota na daném řádku liší od většiny ostatních. Z tohoto příkladu můžeme rovnou odvodit formulaci úlohy separace.

Matici  $\mathbf{M}$  chceme vyjádřit jako součet matice  $\mathbf{L}$  s maticí  $\mathbf{S}$ , přičemž  $\mathbf{L}$  má všechny sloupce identické, tzn. každý sloupec reprezentuje statické pozadí a  $\mathbf{S}$  má nenulové hodnoty pouze v místech, kde se hodnota na daném řádku liší od většiny ostatních, tzn. v místech, kde se pohybuje kulička (dynamická složka).

Následující metody představují návod, jak původní matici  $\mathbf{M}$  vhodně rozdělit na matice  $\mathbf{L}$  a  $\mathbf{S}$ .

### 3.2. Mediánový filtr

Mediánový filtr je jednou z nejstarších a nejjednodušších metod pro separaci pozadí z videa [2]. Výhodou této metody je především snadná implementace a nízká výpočetní náročnost i pro videa s vysokým rozlišením. Nevýhodou je, že metoda může selhávat při větším množství pohybujících se objektů.

Medián je hodnota, která dělí konečnou množinu reálných vzestupně seřazených dat na dvě stejně početné poloviny a je předpokládáno, že sloupce matice  $\mathbf{M}$  jsou si podobné a liší se pouze v několika místech odpovídajících složce  $\mathbf{S}$ . Tedy obsahuje-li nadpoloviční počet sloupců na příslušném řádku hodnotu odpovídající hodnotě původního pozadí, medián daného řádku bude odpovídat právě této hodnotě. Obraz, který získáme výpočtem mediánu hodnot všech obrazů, považujeme za dostatečně dobrý model pozadí. Dynamickou složku spočítáme jako rozdíl jednotlivých obrazů a pozadí.

Tedy vezmeme matici vstupních dat  $\mathbf{M}$  a spočítáme  $\mathbf{n}$  mediánů pro jednotlivé řádky. Pokud se některá hodnota liší od daného řádkového mediánu, tuto hodnotu jím nahradíme. Např. vypadá-li první řádek  $\mathbf{M}$  takto  $[5, 1, 5, 5, 5, 25, 103, 5, 5, 0, 5]$ , mediánem tohoto řádku je číslo pět. První řádek matice  $\mathbf{L}$  získaný pomocí mediánového filtru bude stejné délky jako první řádek matice  $\mathbf{M}$ , ale bude obsahovat pouze pětky.

Aplikujeme-li tento postup na všechny řádky, získáme matici pozadí  $\mathbf{L}$ . Matici dynamické složky  $\mathbf{S}$  dostaneme odečtením pozadí od původního obrázku, tj.  $\mathbf{S} = \mathbf{M} - \mathbf{L}$ .

### 3.3. Principal component pursuit

Již novější metodou je Principal Component Pursuit (může být také označována jako Robust Principal Component Analysis RPCA – robustní analýza hlavních komponent). Jak už název napovídá, princip metody je založen na nalezení „hlavních komponent“, které nejvíc ovlivňují daty popsaný jev. Navíc RPCA na rozdíl od klasického PCA dokáže dobře pracovat i s daty obsahujícími chyby a další nepřesnosti způsobené měřením [1].

Jak už bylo naznačeno v části předzpracování, matice  $\mathbf{S}$  má nenulové hodnoty pouze na několika málo místech, je tedy řídká. Matice  $\mathbf{L}$  má všechny sloupce identické, v ideálním případě tedy platí  $\text{rank } \mathbf{L} = 1$ . V reálném případě bude hodnota sice vyšší, ale stále bude poměrně malá. Z tohoto důvodu budeme matici  $\mathbf{L}$  nazývat nízkohodnostní. Cílem je tyto dvě složky odseparovat. Zapišeme-li daný problém formálně, získáme optimalizační úlohu

$$\min_{\mathbf{L}, \mathbf{S}} \text{rank}(\mathbf{L}) + \|\mathbf{S}\|_0 \quad \text{za podmínky} \quad \mathbf{L} + \mathbf{S} = \mathbf{M}. \quad (3.1)$$

Hodnosti matice odpovídá počet nenulových singulárních čísel. Tudíž chceme-li, aby matice  $\mathbf{L}$  měla co nejmenší hodnotu, musíme minimalizovat počet nenulových singulárních čísel dané matice. Toho lze ve většině případů dosáhnout pomocí nukleární normy, protože nukleární norma určuje hodnotu součtu singulárních čísel dané matice.

Pro získání řídké matice je vhodné minimalizovat její  $\ell_0$  normu, která odpovídá počtu nenulových prvků dané matice. Bohužel minimalizace  $\ell_0$  normy je NP-těžký problém. Za jistých podmínek je možné nahradit  $\ell_0$  normu konvexní  $\ell_1$  normou, která již není NP-těžká (viz [4]). Proto se pokusíme úlohu (3.1) přeformulovat pomocí  $\ell_1$  normy a výše zmíněné nukleární normy na již konvexní problém (viz [1])

$$\min_{\mathbf{L}, \mathbf{S}} \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \quad \text{za podmínky} \quad \mathbf{L} + \mathbf{S} = \mathbf{M}. \quad (3.2)$$

Díky [1] zjišťujeme, že pro zajištění přesné dekompozice stačí, aby  $\mathbf{M}$  nebyla nízkohodnostní a řídká zároveň. Tedy potřebujeme zajistit, aby nízkohodnostní prvek  $\mathbf{L}$  nebyl zároveň řídký, což se dá zaručit splněním několika jednoduchých podmínek s parametrem.

Další problém může nastat, pokud řídká matice  $\mathbf{S}$  bude nízkohodnostní. Proto budeme po řídké matici požadovat, aby její nenulové prvky byly rozmístěné náhodně.

Jsou-li výše zmíněné podmínky splněny, tak existuje tvrzení (viz [1]) zaručující, že je řešení úlohy PCP (3.2) s parametrem  $\lambda = 1/\sqrt{n}$  přesné s pravděpodobností téměř jedna. Nejen, že již je zaručena přesnost dekompozice, ale také je z tohoto tvrzení získán návod jak univerzálně volit parametr  $\lambda$ . Toto univerzální  $\lambda$  sice není pro jednotlivé případy ideální, ale bude fungovat ve všech přípustných případech.

Nyní už je potřeba pouze odvodit algoritmus metody PCP. Odvození se provede pomocí metody ADMM (viz [3]) a tzv. rozšířeného Lagrangianu (viz [1]) pro náš

daný problém (3.2) a jsou z něj získány následující iterativní předpisy pro  $\mathbf{L}$  a  $\mathbf{S}$ :

$$\mathbf{L}^{(i+1)} = \text{svt}_{\frac{\lambda}{\mu}} \left( \mathbf{M} - \mathbf{S}^{(i)} + \mathbf{Y}^{(i)} / \mu \right), \quad (3.3)$$

$$\mathbf{S}^{(i+1)} = \text{soft}_{\frac{\lambda}{\mu}} \left( \mathbf{M} - \mathbf{L}^{(i+1)} + \mathbf{Y}^{(i)} / \mu \right), \quad (3.4)$$

kde operátor

$$\text{soft}_{\lambda}(x_i) = \frac{x_i}{|x_i|} \max(|x_i| - \lambda, 0)$$

je tzv. *měkké prahování* (viz [4]), které je aplikováno na matici po složkách a každou hodnotu v absolutní hodnotě menší než parametr  $\lambda$  zobrazí na nulu. Tedy je to výhodný operátor pro hledání řídké matice, protože všechny malé „chybové“ hodnoty zruší a ponechá pouze ty význačné.

Druhý použitý operátor je tzv. *singular value thresholding* odpovídající měkkému prahování singulárních čísel

$$\text{svt}_{\lambda}(\mathbf{X}) = \sum_{i=1}^n \text{soft}_{\lambda}(\sigma_i) \mathbf{u}_i \mathbf{v}_i^*.$$

Tento operátor je vhodný pro hledání nízkohodnostní matice, protože zmenšuje počet nenulových singulárních čísel, tedy zmenšuje hodnotu matice.

Spojením rovnic (3.3) a (3.4) s maticí Lagrangeových multiplikátorů  $\mathbf{Y}$  získáváme výsledný algoritmus pro PCP.

---

**Algoritmus 1:** Principal Component Pursuit (PCP)

---

inicializace:  $\mathbf{S}_0 = \mathbf{Y}_0$ ,  $\mu > 0$ ;  
**while**  $\|\mathbf{M} - \mathbf{L}_{i+1} - \mathbf{S}_{i+1}\|_F > \delta \|\mathbf{M}\|_F$  **do**  
     $\mathbf{L}_{i+1} = \text{svt}_{\frac{\lambda}{\mu}}(\mathbf{M} - \mathbf{S}_i + \mathbf{Y}_i / \mu)$ ;  
     $\mathbf{S}_{i+1} = \text{soft}_{\frac{\lambda}{\mu}}(\mathbf{M} - \mathbf{L}_{i+1} + \mathbf{Y}_i / \mu)$ ;  
     $\mathbf{Y}_{i+1} = \mathbf{Y}_i + \mu(\mathbf{M} - \mathbf{L}_{i+1} - \mathbf{S}_{i+1})$ ;  
**end**  
**Výsledek:**  $\mathbf{L}, \mathbf{S}$

---

#### 4. UKÁZKY VÝSLEDKŮ

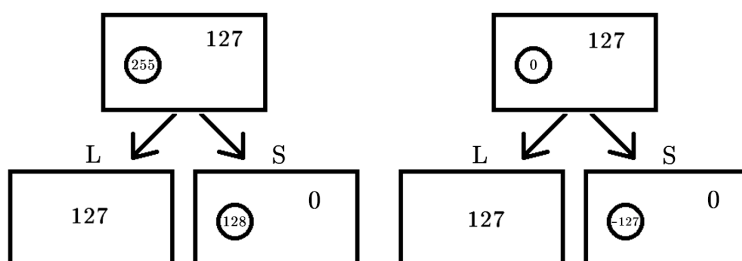
Než přejdeme k samotným ukázkám výsledků, je třeba rozebrat problém vznikající při vykreslování výsledků.

##### 4.1. Vykreslení

Separaci provádíme na klasických 8-bitových obrazech, tedy jednotlivé pixely nabývají hodnot mezi nulou a číslem 255, kde nula odpovídá černé barvě a 255 odpovídá bílé barvě. V důsledku separace bude dynamická složka ve většině případů vykreslena v tmavších odstínech, než byl originál. Dokonce se může stát, že výsledná složka bude v záporných číslech. Jelikož snímky převádíme z formátu *double*

na formát `uint8`, budou všechna záporná čísla zaokrouhlena na nulu. Tato situace může způsobit „zmizení“ celé složky, protože je zobrazována na černém pozadí, tedy na pozadí s hodnotou nula.

Popsaný problém ilustruje obrázek 1. Na levém obrázku je našimi vstupními daty bílá (255) kulička pohybující se po šedém (127) pozadí. Po provedení separace bude barva pozadí zachována, ale barva kuličky ztmavne na šedou (128). To není velký problém, protože kulička je pouze tmavší, ale stále jasně viditelná. Může ale také nastat situace ilustrovaná vpravo. Vstupem je černá (0) kulička na šedém (127) pozadí. Výstupem ze separace je opět správné pozadí, ale kulička bude mít barvu (−127). Pokud snímek převedeme do formátu `uint8`, bude číslo zaokrouhleno na nulu, tedy kulička nebude při vykreslení složky **S** viditelná. Abychom



**Obrázek 1.** Ilustrace ztmavení dynamické složky v důsledku separace.

takovému problému předešli, musíme složku **S** před vykreslením upravit.

Jednou z možností úpravy je přeskálování každého ze snímků zvlášť tak, aby nejmenší hodnota odpovídala nule a největší hodnota odpovídala číslu 255. Tato úprava je vhodná pouze pro data bez barevně rozmanitého pozadí, jako je např. video dálnice. Pro videa, kde se dynamická složka pohybuje přes barevně rozmanitá pozadí, vzniká v důsledku různého ztmavení dynamické složky na jednotlivých snímcích nepříjemný jev „blikání“.

Další možností je přičtení pozadí k nenulovým hodnotám dynamické složky. Tím získáme přesné barvy, ale bohužel budou zvýrazněny i chyby, které vytvoří šum v okolí jednotlivých objektů. Jako nejlepší možnost vyhlazení tohoto šumu se jeví použití adaptivního lineárního filtru, jako je např. Wienerův filtr [7] a následného morfologického uzavření, které vyhledá v dynamické složce „díry“ vzniklé filtrací. Tento způsob se osvědčil i pro obrazy s různorodým pozadím a dynamickou složkou nepravidelného tvaru.

#### 4.2. Porovnání metod na simulovaných datech

Nejprve budou metody porovnány na simulovaných datech obsahujících 90 různých snímků s rozlišením  $160 \times 120$  px. Výpočty probíhají na notebooku s procesorem Intel Core i7-8550U, 1,80 GHz a 8 GB RAM. Výsledky vidíme v tabulce 1, kde *max rozdíl* značí největší rozdíl výsledného a originálního pozadí získaný pomocí

Frobeniovy normy, *prům. rozdíl* značí průměrný rozdíl získaný Frobeniovou normou, *min SSIM* označuje nejmenší naměřený SSIM index<sup>1</sup> a *prům. SSIM* označuje průměrný SSIM index mezi jednotlivými výsledky a originálním pozadím.

Výsledky pro 90 různých snímků					
	čas [s]	max rozdíl	prům. rozdíl	min SSIM	prům. SSIM
Mediánový filtr	0,03	78,82	78,82	0,9987	0,9987
PCP	2,76	78,82	78,82	0,9987	0,9987

**Tabulka 1.** Porovnání metod na simulovaných datech obsahujících 90 různých snímků. Výsledné snímky pozadí jsou porovnány s originálním pozadím a podobnost je zhodnocena pomocí Frobeniovy normy, SSIM indexu a časové náročnosti.

Z tabulky 1 je patrné, že obě metody pozadí oddělily téměř dokonale. Lze vidět že PCP metoda je časově náročnější než mediánový filtr, to je způsobeno počítáním výpočetně náročného singulárního rozkladu v každé iteraci.

Obě metody si dokáží dobře poradit i s objekty dynamické složky, které jsou po část videa statické. To lze pozorovat v tabulce 2.

Výsledky pro 130 snímků se 40 stejnými					
	čas [s]	max rozdíl	prům. rozdíl	min SSIM	prům. SSIM
Mediánový filtr	0,04	78,82	78,82	0,9987	0,9987
PCP	10,21	134,48	98,33	0,9956	0,9976

**Tabulka 2.** Porovnání metod na simulovaných datech obsahujících 130 různých snímků, z nichž 40 je identických. Výsledné snímky pozadí jsou porovnány s originálním pozadím a podobnost je zhodnocena pomocí Frobeniovy normy, SSIM indexu a časové náročnosti.

### 4.3. Výsledky na reálných datech

Nyní metody otestujeme na reálných datech. K tomu bude využito video<sup>2</sup> dálnice, protože obsahuje jak statickou složku (dálnice, tráva, světla), tak dynamickou složku (vozidla na dálnici a vozidla projíždějící v pozadí). Video obsahuje 380 snímků a jeho rozlišení je  $256 \times 144$  px.

Separace pomocí mediánového filtru trvala pouze 0,27 s, zatímco separace pomocí PCP trvala 364,43 s, tedy je již velmi znatelný rozdíl v časové náročnosti obou metod způsobený náročností SVD rozkladu. Tento problém PCP metody je ale vyvážen mnohem lepší kvalitou dynamické složky, u mediánového filtru je tato složka poměrně zašuměná.

Na obrázku 2 je snímek z daného videa dálnice. V pravém horním rohu je vyobrazeno v dálce projíždějící vozidlo. Na tomto snímku můžeme otestovat, zdali jsou

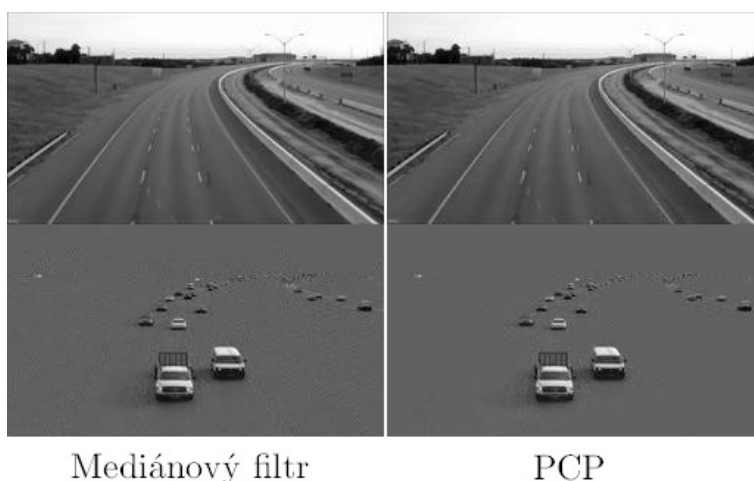
<sup>1</sup>Jedná se o index vyjadřující podobnost obrazů. Může nabývat hodnot z intervalu  $(-1, 1)$ , kde hodnota 1 značí dva zcela identické obrazy a hodnota 0 značí nulovou strukturální podobnost [5].

<sup>2</sup>Dostupné z <https://www.youtube.com/watch?v=PJ5xXXcfuTc>.



**Obrázek 2.** Snímek z videa dálnice. V levém horním rohu je zeleně vyznačeno projíždějící vozidlo.

metody schopny rozpoznat i takto malý pohybující se objekt. Jelikož se vozidla pohybují po téměř stále stejném pozadí, volíme jako způsob vykreslení první způsob škálování. Z obrázku 3 je patrné, že obě metody oddělily dynamickou složku skvěle



**Obrázek 3.** Porovnání škálovaně vykreslených výsledků separace jednotlivých metod. Na horním snímku je znázorněna statická složka a na spodním snímku vidíme dynamickou složku.

včetně v dálce projíždějícího vozidla.

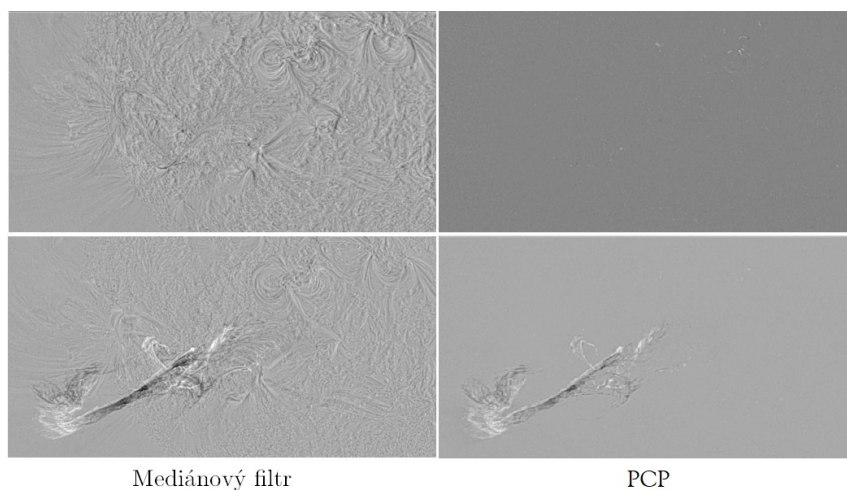
Dále jsme otestovali obě metody na videu<sup>3</sup> sluneční koróny prof. Druckmüllera. Video obsahuje 565 snímků o rozlišení  $2560 \times 1440$  px, jedná se tedy o obrovská data. Z tohoto důvodu výpočty probíhají na počítači s procesorem Intel Xeon CPU E7-4820, 2 GHz a 128 GB RAM.

Na takto velkých datech se ukazuje, že PCP metoda je opravdu časově příliš náročná. Výpočet mediánového filtru trval necelou minutu, zatímco výpočet PCP metody, při kterém proběhlo 2363 iterací, trval 6,5 dne.

<sup>3</sup>Dostupné z [http://www.zam.fme.vutbr.cz/~druck/SD0/Pm-nafe/2012\\_08\\_31/0-info.htm](http://www.zam.fme.vutbr.cz/~druck/SD0/Pm-nafe/2012_08_31/0-info.htm).

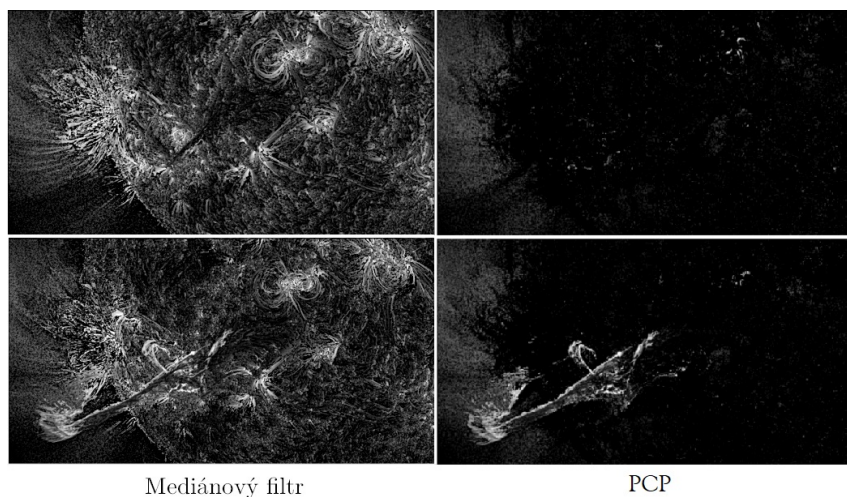


Poté výsledky vykreslíme nejprve pomocí škálování. Již ze škálovaného vykreslení (obrázek 4) je patrné, že mediánový filtr do dynamické složky zařadil i značné



**Obrázek 4.** Porovnání výsledků separace videa slunce získaných mediánovým filtrem a PCP metodou. Jsou zde škálovaně vykresleny dva snímky dynamické složky výsledku.

množství statické složky, zatímco PCP metoda zvládla pozadí velmi dobře oddělit. Podíváme-li se na vykreslení s použitím filtru (obrázek 5), je tento výsledek ještě zřetelnější.



**Obrázek 5.** Porovnání výsledků separace videa slunce získaných mediánovým filtrem a PCP metodou. Jsou zde filtrovaně vykresleny dva snímky dynamické složky výsledku. Jako filtr je použit pouze Wienerův filtr.

## 5. ZÁVĚR

Představili jsme dvě metody pro separaci dynamické a statické složky ve videu. Z výsledků se ukazuje, že metoda mediánového filtru je velmi rychlá, ale není vhodná pro videa s velkým množstvím šumu jako je sluneční koróna. PCP metoda má velmi přesné výsledky i pro videa tohoto typu, ale její výpočetní náročnost je natolik velká, že je vhodná pouze pro práci, při které požadujeme velmi vysokou přesnost výsledků.

## REFERENCE

- [1] E. J. Candès, X. Li, Y. Ma, J. Wright: *Robust principal component analysis?*, Journal of the ACM **58** (2011), No. 3, Article 11, 1–37.
- [2] R. Cucchiara, M. Piccardi, A. Prati: *Detecting moving objects, ghosts, and shadows in video streams*, IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003), No. 10, 1337–1442.
- [3] K. Gebrtová: *Separace dynamické a statické složky v sérii obrazů*, Bakalářská práce, Fakulta strojního inženýrství Vysoké učení technické v Brně, Brno, 2020.
- [4] P. Rajmic: *Řídké a nízkohodnostní reprezentace signálů s aplikacemi*, Habilitační práce, Vysoké učení technické v Brně, Brno, 2014.
- [5] *Structural similarity*, Wikipedia: the free encyclopedia, San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2021-7-22], online [https://en.wikipedia.org/wiki/Structural\\_similarity](https://en.wikipedia.org/wiki/Structural_similarity).
- [6] L. N. Trefethen, D. Bau: *Numerical linear algebra*, Philadelphia: Society for Industrial and Applied Mathematics, 1997.
- [7] *Wiener2*, The MathWorks [cit. 2020-6-14], online <https://www.mathworks.com/help/images/ref/wiener2.html>.

Karolína Gebrtová, Ústav matematiky, Fakulta strojního inženýrství, Vysoké učení technické v Brně, Technická 2, 616 69 Brno, Česká republika,  
*e-mail*: 200875@vutbr.cz