

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra radioelektroniky



Bakalářská práce

Aplikace metod segmentace popředí
Application of foreground segmentation methods

Anastasia Kozitsyna

Studijní program: Komunikace, multimédia a elektronika

Studijní obor: Multimediální technika

Vedoucí práce: Ing. Stanislav Vítek, Ph.D.

Praha, 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická
katedra radioelektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Anastasia Kozitsyna**

Studijní program: Komunikace, multimédia a elektronika
Obor: Multimediální technika

Název tématu: **Aplikace metod segmentace popředí**

Pokyny pro vypracování:

Cílem bakalářské práce je realizace systému pro sledování návštěvníků interaktivní multimediální instalace.

1. Seznamte se s metodami segmentace popředí z videosekvencí pořízených nepohyblivou kamerou.
2. Vybrané algoritmy implementujte v programovém prostředí vhodném pro zpracování obrazu v reálném čase.
3. Navrhněte a implementujte způsob získávání doplňkové informace, např. informace o natočení hlavy.
4. Diskutujte funkcionalitu celého systému (výpočetní náročnost, vliv nasvícení scény na jednotlivé algoritmy apod.)

Seznam odborné literatury:

- [1] Bouwmans, Thierry. "Recent advanced statistical background modeling for foreground detection-a systematic survey." *Recent Patents on Computer Science* 4.3 (2011): 147-176.
- [2] Benezeth, Yannick, et al. "Comparative study of background subtraction algorithms." *Journal of Electronic Imaging* 19.3 (2010): 033003-033003.
- [3] Aghajan, Hamid, and Andrea Cavallaro, eds. *Multi-camera networks: principles and applications*. Academic press, 2009.

Vedoucí: Ing. Stanislav Vítek, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

doc. Mgr. Petr Páta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 2. 2016

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 27.5.2016

Anastasia Kozitsyna

Poděkování

Na tomto místě bych chtěla poděkovat vedoucímu bakalářské práce Ing. Stanislavu Vítkovi, Ph.D. za odborné vedení, cenné rady a optimismus.

Abstrakt

Tato bakalářská práce se zabývá studiem a aplikací algoritmů segmentace popředí z videosekvencí pořízených nepohyblivou kamerou. První část práce je rozbor pojmů použitých ve výkladu. Druhá část je věnována popisu algoritmů segmentace a jejich porovnání. Vybrané algoritmy jsou implementovány v jazyce Python s použitím knihovny OpenCV a v prostředí MATLAB. Třetí část popisuje tvorbu systému pro sledování návštěvníků interaktivní multi-mediální instalace.

Klíčová slova

Segmentace popředí, zpracování obrazu, OpenCV, sledování pohybu, interaktivní instalace, barevná segmentace

Abstract

This bachelor thesis deals with the study and application of background subtraction algorithms for videos from static cameras. The first part focuses on theoretical description of terms, which are used in the document. The aim of the second part is to explain and compare segmentation algorithms. Some of algorithms are implemented by using Python with OpenCV library and MATLAB. The third part describes the formation of a visitor monitoring system for interactive multimedia installation.

Key words

Background subtraction/segmentation, image processing, OpenCV, tracking, interactive installation, color segmentation

Obsah

1	Úvod	1
2	Vybrané pojmy	2
2.1	Matematika	2
2.2	Barevné prostory	4
2.3	Filtry rozostření	5
2.4	Morfologické operace	6
2.5	Obrazové momenty	8
2.6	Kalmanův filtr	8
3	Metody segmentace popředí	10
3.1	Základní (Basic)	11
3.2	Statistické (Statistical)	13
3.2.1	Gaussovské (Gaussian)	13
3.2.2	Podprostorové (Subspace learning)	17
3.3	Shlukové (Clusters)	20
4	Porovnání algoritmů segmentace	22
5	Instalace	28
5.1	Popis instalace	28
5.2	Obrazová část	30
5.3	Zvuková část	34
6	Závěr	36
7	Obsah příloženého CD	37

Seznam použitých zkratek

1G Running Gaussian Average 13, 23, 27

Adaptive Running Average 12, 23, 27

CB Codebook 20, 23, 26, 27

FD Frame Difference 11

FN False Negatives 22

FP False Positives 22

GT Ground Truth 10, 22, 37

Mean Mean Filter 11

MOG Mixture Of Gaussian 15, 23, 27

PCA Principal Components Analysis 17, 23, 27

Static Static Frame Difference 11, 23, 27, 37

TN True Negatives 22

TP True Positives 22

Seznam obrázků

1	Gaussovo rozdělení, křivka	3
2	RGB prostor, model	4
3	Rozklad obrázku na jednotlivé kanály: červený, zelený, modrý	4
4	HSV prostor, model	5
5	Rozklad obrázku na jednotlivé složky: odstín H, sytost S, jas V	5
6	Princip mediánového filtru	5
7	Mediánový filtr, 5x5	6
8	Gaussovský filtr, (13,13)	6
9	Příklady strukturních elementů 5x5: čtverec, elipsa, kříž	6
10	Originál obrázku	7
11	Vliv morfologických operací pro strukturní element elipsa 5x5 a) dilatace b) eroze c) otevření d) uzavření	7
12	Příklad činnosti Kalmanova filtru, zelená tečka odpovídá odhadu	9
13	Vstupní snímek a maska, odpovídající vjemu člověka	10
14	Pozadí - první snímek, výsledky	11
15	Pozadí - předchozí snímek, výsledek	11
16	Pozadí - průměr z 10 předchozích snímků, výsledky	12
17	Pozadí - průměr z 50 předchozích snímků, výsledky	12
18	Aktualizované pozadí s parametrem $\alpha=0.1$, výsledky	12
19	Aktualizované pozadí s parametrem $\alpha=0.01$, výsledky	12
20	Příklady chybné detekce popředí	13
21	Příklad parametrů pro jeden konkrétní pixel videosekvence	14
22	Výsledek segmentace pro metodu 1G, implementace pomocí knihovny [33]	14
23	MOG, tvorba modelu pozadí	16
24	MOG, schéma algoritmu [21]	17
25	Výsledek segmentace pro metodu MOG	17
26	Vlastní tváře, sada obrázků na vstupu (vlevo) a vlastní vektory (vpravo) [22]	18
27	PCA, tvorba vlastního prostoru	19
28	PCA, detekce popředí	19
29	Výsledek segmentace pro metodu PCA	19
30	Struktura kódové knihy	20
31	Výsledek segmentace pro metodu CB, implementace pomocí [26]	22
32	Výsledky segmentace pro videosekvenci Kampus	24
33	Výsledky segmentace pro videosekvenci Světla	25
34	Výsledky segmentace pro videosekvenci Obchod	25
35	Výsledky segmentace pro videosekvenci Maskování	26
36	Instalace, schématické znázornění	29
37	Video z místnosti, výsledky	30
38	Ukázka umístění míček	31
39	Příklad kalibrace rybího oka	32
40	Příklad nastavení parametrů kamery	33
41	Příklad barevné segmentace v HSV prostoru	33
42	Výpočet úhlu na příkladu víček	33
43	Ukázka výstupu programu	34
44	Zjednodušené schéma obrazové části programu. Příkazy v zelených blocích mají být aplikovány na oba míčky	35

Seznam tabulek

1	Parametry algoritmů, kde T je práh, α je učící konstanta, LF je počet snímku pro trénování, n je použitý počet Gaussových rozdělení, HS je velikost historie, Dim je počet dimenzí, N je počet trénovacích snímků	24
2	Kampus, výsledky pro 10 dvojic snímků 160x128	24
3	Světla, výsledky pro 10 dvojic snímků 160x128	25
4	Obchod, výsledky pro 10 dvojic snímků 320x256	26
5	Maskování, výsledky pro dvojici snímků 160x120	26
6	Celková úspěšnost algoritmů	27
7	Souhrnný popis algoritmů, kde N , K , M , N jsou počty snímků, Gaussových rozdělení, vlastních vektorů a kódových slov, α je učící konstanta - parametry mající vliv na zatížení algoritmů.	36

1 Úvod

Segmentace popředí neboli detekce pohybu je první krok pro následné zpracování dat: trasování, analýzu chování, detekci podezřelých předmětů.

Budeme vycházet z předpokladu, že se scéna skládá ze statických objektů - pozadí (**Background**) a pohybujících se objektů – popředí (**Foreground**). Změny hodnot pixelů pozadí v průběhu pozorování jsou buď malé, nebo žádné. Popředí je naopak charakterizováno rychlou změnou hodnot pixelů. Výstupem segmentace jsou zpravidla oblasti odpovídající objektům v pohybu (například postavě nebo vozidlu), s nimiž následně pracujeme.

Bezchybné oddělení objektů zájmu od pozadí ve videosekvenci je triviální úloha pro člověka. Avšak ne pro počítač, který reaguje pouze na změny. K chybnému zařazení pixelů do popředí dochází například v případě nehomogenního pozadí: hladina vody, stromy ve větru, pohyblivé schody nebo odrazy slunce. Další zdroje chyb jsou změny v osvětlení scény a automatické úpravy kamery (vyvážení bílé, automatické ostření). Může také dojít k opačnému problému: objekty ve skutečnosti patřící popředí budou chybně zahrnuty do pozadí. Například když se objekt na nějakou dobu zastaví. Tudíž, úkolem algoritmů je segmentovat snímek co nejlépe, jak by to udělal člověk.

Existuje velké množství přístupů, které se liší způsobem modelování pozadí, výpočetní složitostí, nároky na paměť, robustností. Klasifikace algoritmů segmentace není pevně dána. V literatuře můžeme nalézt klasifikace metod například podle rychlosti, přesnosti a nároků na paměť [30], nebo podle vlastností algoritmů, které mohou být rekurzivní či nerekurzivní. [14] V této práci budu používat klasifikaci navrhnoutou Thierry Bouwmansem [13], a to podle matematické podstaty metod. Zaměřím se především na klasické přístupy, protože z nich vychází většina pokročilejších metod.

Bakalářskou práci lze rozdělit na několik částí. První část je věnována krátkému rozboru pojmů použitých ve výkladu, a to jak matematického aparátu, tak i pojmů z oblasti zpracování obrazů. Přístupy jsou použity i k tvorbě obrázků z této sekce. Následuje popis algoritmů segmentace a jejich porovnání. Vybrané algoritmy jsou implementovány v jazyce Python s použitím knihovny OpenCV [5] a v prostředí MATLAB [2]. Poslední část popisuje tvorbu systému pro sledování návštěvníků interaktivní multimediální instalace.

2 Vybrané pojmy

Tato část slouží k vysvětlení (nebo připomenutí) některých pojmů a nástrojů, o kterých je zmíněno v následujícím výkladu. Pro informace týkající se oblasti zpracování obrazu jsou také uvedeny poznámky ohledně specifičnosti činnosti příslušných funkcí knihovny OpenCV. Doporučuju se na tuto část obracet jen v okamžiku, kdy se v textu objeví odkaz.

2.1 Matematika

Informace z této sekce jsou nezbytné pro pochopení statistických přístupů k segmentaci pozadí 3.2. Čerpáno z [28].

Střední hodnota

Často potřebujeme znát "průměrné" hodnoty náhodných veličin¹, obzvlášť když pracujeme s velkými soubory dat. Pro diskrétní náhodnou veličinu X nabývající hodnot x_1, x_2, \dots , $P(X = x_i) = p_i$, je *střední hodnota* definována následujícím způsobem:

$$EX = \sum_{i=1}^{\infty} x_i p_i \quad (1)$$

Jedná se o vážený průměr možných hodnot veličiny X s vahami odpovídajícími jednotlivým pravděpodobnostem. $P(x)$ je pravděpodobnostní funkce. Dále v textu pro střední hodnotu bude použito značení μ . Ve většině případů počítáme jako aritmetický průměr.

Rozptyl

Rozptyl neboli *disperze* (**Variance**) je mírou variability náhodné veličiny. Je to střední hodnota kvadrátu odchylky od střední hodnoty.

$$DX = E((X - EX)^2) \quad (2)$$

Značíme σ^2 .

Pokud pracujeme s náhodnými vektory $\mathbf{X} = (X_1, \dots, X_n)$, střední hodnota a rozptyl jsou vektory hodnot pro jednotlivé složky: $E\mathbf{X} = (EX_1, \dots, EX_n)$ a $D\mathbf{X} = (DX_1, \dots, DX_n)$

Směrodatná odchylka

Směrodatná odchylka (**Standard deviation**) je střední kvadratická odchylka od střední hodnoty.

$$\sigma = \sqrt{DX} \quad (3)$$

Kovariance

Kovariance vyjadřuje závislost dvou náhodných veličin X a Y , je definována jako střední hodnota součinu odchylek X a Y od jejich střední hodnoty.

$$\text{cov}(X, Y) = E((X - EX)(Y - EY)) = E(XY) - EXEY \quad (4)$$

Kovarianční matice

Kovarianční matice náhodného vektoru $\mathbf{X} = (X_1, \dots, X_n)$ je čtvercová, symetrická ($A^T = A$), na diagonále má rozptyly, mimo diagonálu kovariance. Je definována následujícím způsobem:

¹Ve sféře zpracování obrazu náhodné veličiny jsou hodnotami intenzit pixelů

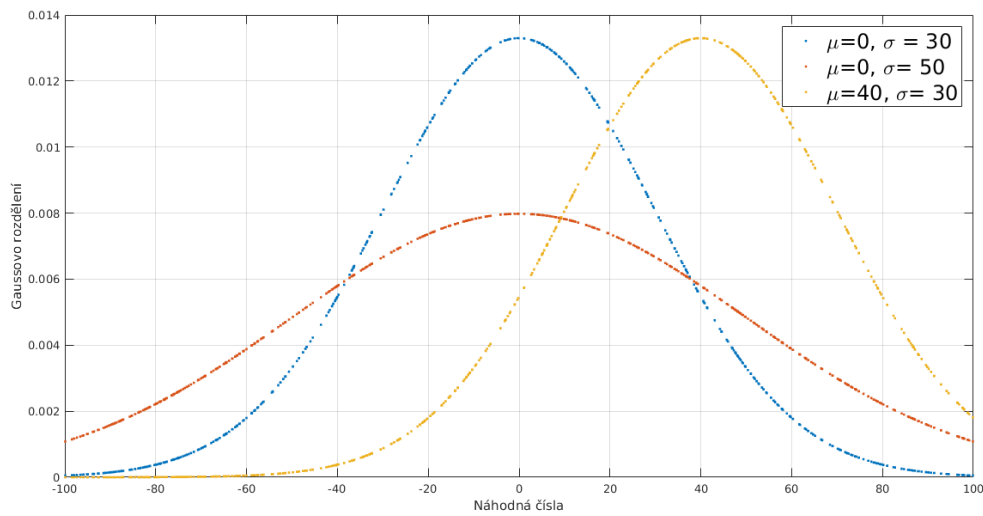
$$\Sigma = \begin{pmatrix} \text{cov}(X_1, X_1) & \text{cov}(X_1, X_2) & \cdots & \text{cov}(X_1, X_n) \\ \text{cov}(X_2, X_1) & \text{cov}(X_2, X_2) & \cdots & \text{cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \text{cov}(X_n, X_2) & \cdots & \text{cov}(X_n, X_n) \end{pmatrix}$$

Gaussovo rozdělení pravděpodobnosti

Gaussovo neboli *Normální rozdělení* $N(\mu, \sigma^2)$ je definováno střední hodnotou μ a rozptylem σ . Hustota je:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty \quad (5)$$

Jedná se o nejčastěji používané rozdělení.



Obrázek 1: Gaussovo rozdělení, křivka

Vlastní čísla a vlastní vektory matice

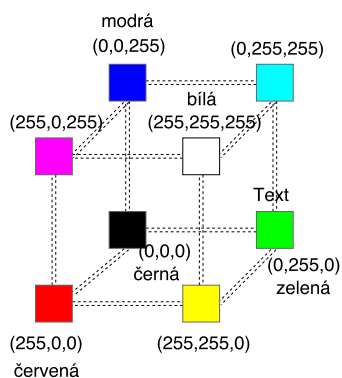
Číslo α je *vlastní číslo* čtvercové matice \mathbf{A} , jestli existuje nenulový vektor \mathbf{x} takový, že $\mathbf{Ax} = \alpha\mathbf{x}$. \mathbf{x} nazýváme *vlastním vektorem* odpovídajícím vlastnímu číslu α . Vlastní vektory příslušné navzájem různým vlastním číslům jsou lineárně nezávislé.

2.2 Barevné prostory

Barevný prostor definuje základní barvy a pravidla jejich míchání.

RGB

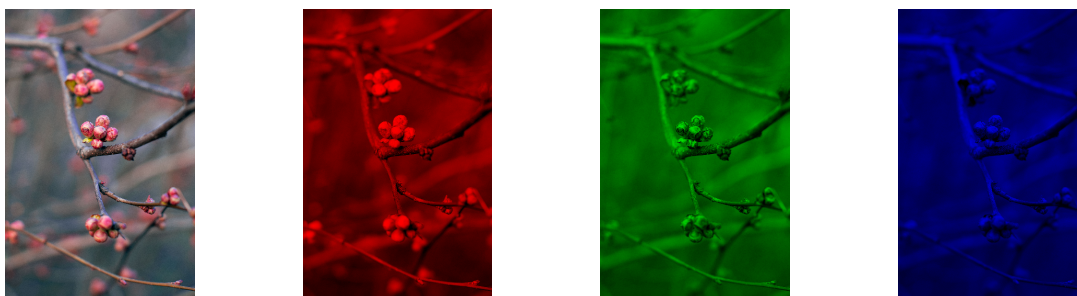
Barevný prostor *RGB* se skládá ze tří barev: červené (**R**ed), zelené (**G**reen) a modré (**B**lue). Každá barva má rozsah od 0 (není zastoupena) do 255 (maximální intenzita). Složením všech složek s maximální hodnotou intenzity dostaneme bílou barvu, s minimální barvu černou.



Obrázek 2: RGB prostor, model

Lidské oko není stejně citlivé na každou barvu, proto se k výpočtu jasu používá vztah:

$$I = 0.299R + 0.587G + 0.114B \quad (6)$$



Obrázek 3: Rozklad obrázku na jednotlivé kanály: červený, zelený, modrý

Z historických důvodů knihovna OpenCV pracuje s prostorem BGR místo RGB. [27]

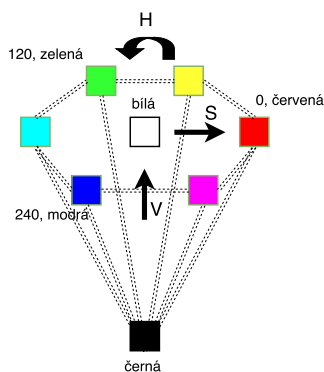
HSV

HSV je barevný model, který nejvíce odpovídá vjemu a intuitivnímu popisu barev člověkem. Skládá se ze tří složek:

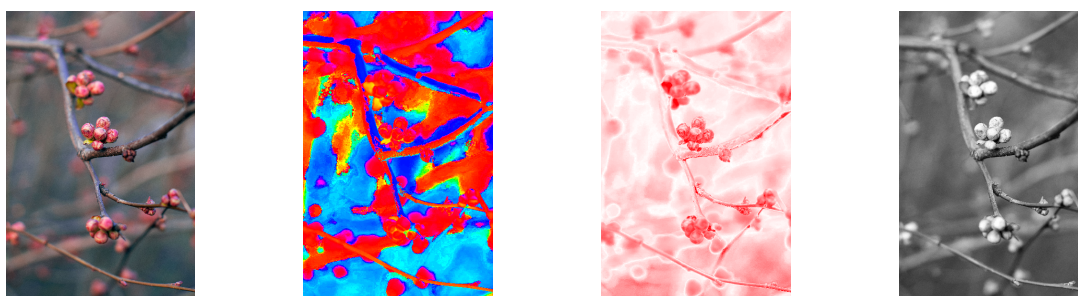
Odstín (**H**ue) představuje převládající barevný tón (červený, zelený...). Měří se jako poloha na barevném kole (0° až 360°).

Sytost (**S**aturation) určuje čistotu barvy neboli příměs šedi. Mění se v rozmezí od 0% (šedá) do 100% (plně sytá). Na barevném kole vzrůstá sytost od středu k okrajům.

Jas (**V**alue) vyjadřuje relativní světlost nebo tmavost barvy a má rozsah 0% (černá) až 100% (bílá).



Obrázek 4: HSV prostor, model



Obrázek 5: Rozklad obrázku na jednotlivé složky: odstín H, sytost S, jas V

Výhodou HSV prostoru je oddělení jasu od barvy, tím pádem menší citlivost na změny v osvětlení. Tento prostor se často používá v oblasti zpracování obrazu, a to například k účelu detekce konkrétní barvy (kupříkladu pleťové).

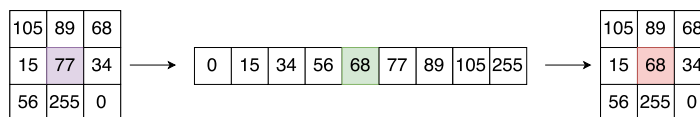
Pro knihovnu OpenCV rozsahy jednotlivých parametrů jsou následující: $H[0,179]$, $S[0,255]$, $V[0,255]$.

2.3 Filtry rozostření

Medián

Medián je široce používaný filtr v oblasti zpracování obrazu. Využívá se tehdy, kdy potřebujeme odstranit šum z obrázku (především šum s velkými rozdíly v jasu, tzv. sůl a pepř šum), ale zachovat hrany objektů.

Pod pojmem mediánový filtr si můžeme představit čtvercové okénko, které je tvořeno lichým počtem pixelů. Tohle okénko se postupně posouvá podél pixelů obrázku. Každý pixel nacházející se ve středu okénka nahrazujeme mediánem z hodnot sousedních pixelů. [16]



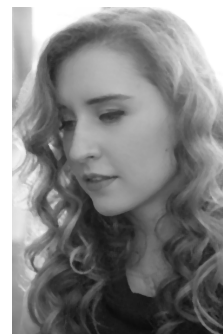
Obrázek 6: Princip mediánového filtru



originální obrázek



obrázek s šumem sůl a pepř



obrázek po aplikaci filtru

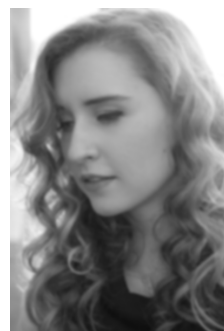
Obrázek 7: Mediánový filtr, 5x5

Gauss

Gaussovský filtr je rozšíření obyčejného průměrování (Filtr průměrování pracuje na stejném principu jako medián, ale místo mediánu počítá průměr z hodnot okolních pixelů), ale používá při tom Gaussovo rozdělení. 2.1 Dochází tedy ke konvoluci každého obrazového bodu s Gaussovým jádrem. Je užitečný pro vyhlazování obrazu, odstranění šumu a detailů. [16]



originální obrázek

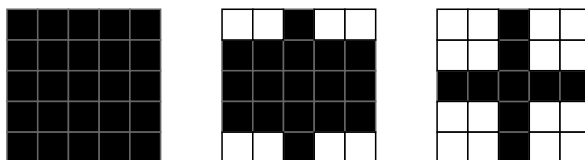


obrázek po aplikaci filtru

Obrázek 8: Gaussovský filtr, (13,13)

2.4 Morfologické operace

Matematická morfologie je nelineární přístup ke zpracování obrazu, nejčastěji je aplikována na binární obraz. Používá se k odstranění šumu, zjednodušení tvarů a segmentaci obrazu, a to především tam, kde je požadován krátký čas zpracování. [20]



Obrázek 9: Příklady strukturálních elementů 5x5: čtverec, elipsa, kříž

Obrázky lze modelovat pomocí bodových množin. Eukleidovský prostor E^2 je definiční obor pro popis rovinných útvarů. [19] Morfologická transformace je relace množiny bodů

obrazu s menší bodovou množinou neboli *strukturním elementem*, který může mít různý tvar podle potřeby. Transformaci lze představit jako systematický pohyb strukturního elementu po obrazu a vyhodnocení odezvy podle typu operace.

Existuje několik základních operací:

Dilatace (Dilation) množiny bodů obrázku I strukturním elementem K je definována jako vektorový součet:

$$I \oplus K = \{d \in E^2 : d = i + k, \forall i \in I, \forall k \in K\} \quad (7)$$

Slouží k zaplnění malých děr a úzkých zálivů v objektech, zvětšuje původní velikost objektu.

Eroze (Erosion) je duální operace k dilataci, představuje rozdíl vektorů:

$$I \ominus K = \{e \in E^2 : e + k, \text{ pro } \forall k \in K\} \quad (8)$$

Používá se pro zjednodušení struktury objektů (objekty spojené tenkými čarami se rozdělí na několik samostatných, objekty menší než strukturní element zmizí), k detekci hran (odečtením erodovaného obrazu od originálu).

Otevření (Opening) je eroze následovaná dilatací.

$$I \bullet K = (I \oplus K) \ominus K \quad (9)$$

Rozděluje objekty, zjednodušuje strukturu objektů, odstraňuje šum.

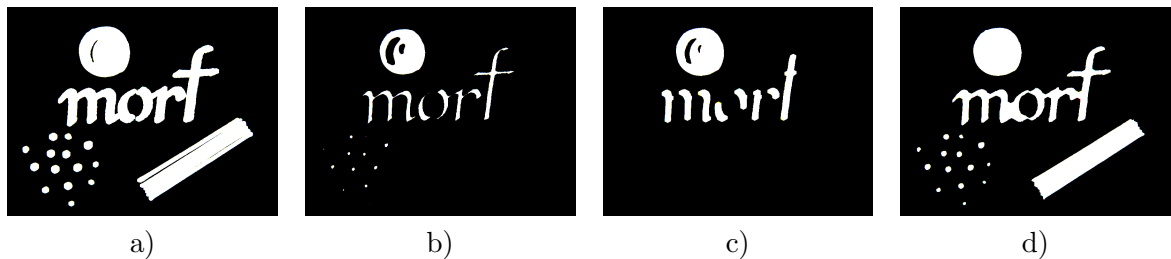
Uzavření (Closing) je dilatace následovaná erozí.

$$I \circ K = (I \ominus K) \oplus K \quad (10)$$

Spojuje blízké objekty, zaplňuje díry, vyhlazuje obrysy.



Obrázek 10: Originál obrázku



Obrázek 11: Vliv morfologických operací pro strukturní element elipsa 5x5 a) dilatace b) eroze c) otevření d) uzavření

Knihovna OpenCV vyžaduje černé pozadí a bílý objekt, pracuje s bílými oblastmi.

2.5 Obrazové momenty

Obrazový moment (Image moment) je "vážený průměr" intenzit pixelů. Používá se pro popis vlastností objektů po segmentaci, například tvaru a středu. [23] Dále budeme předpokládat, že pracujeme s binárními obrázky.

Geometrické momenty (Spatial (raw) moments) se počítají podle:

$$m_{ji} = \sum_{x,y} (I(x,y)x^j y^i) \quad (11)$$

$I(x,y)$ je jasová (obrazová) funkce.

Moment m_{00} je "hmotnost", pro binární obraz představuje plochu objektu. Souřadnice středu objektu ("těžiště") neboli centroid spočítáme jako podíly momentů prvního řádu a plochy:

$$x_c = m_{10}/m_{00} \quad (12)$$

$$y_c = m_{01}/m_{00} \quad (13)$$

Další používané momenty jsou *Centrální momenty (Central moments)*. K výpočtu potřebujeme hodnoty centroidu:

$$mc_{ji} = \sum_{x,y} (I(x,y)(x - x_c)^j (y - y_c)^i) \quad (14)$$

Výhodou centrálních momentů je to, že jsou invariantní vůči transformaci posunutí (**translation**). Používají se pro popis tvaru objektu.

Momenty obrysů jsou definovány stejně, ale výpočet se provádí pomocí Greenové věty. [6]

2.6 Kalmanův filtr

Kalmanův filtr (Kalman filter) je rekurzivní filtr používaný k modelování stavů dynamického lineárního systému. Využívá měření zatížená šumem (s nulovou střední hodnotou a Gaussovým rozdělením) a z nich vypočítává odhad potřebné hodnoty. Informace jsem čerpal z [37].

Mezi výhody filtru patří rychlost a malý nárok na paměť, což umožňuje používat filtr v aplikacích pracujících v reálném čase. Kalmanův filtr se používá v mnoha oborech, například v letectví, námořnictví, radiokomunikaci, bioinženýrství. Lze ho využít i v oblasti počítačového vidění k účelu trasování objektů.

Řekneme, že máme vektor \mathbf{x} , hodnoty kterého nemůžeme přímo změřit, ale chceme znát, a to v každém časovém okamžiku. Pak máme k dispozici jiný vektor veličin daného systému, avšak zatížený šumem - vektor pozorování neboli vektor měřených veličin \mathbf{z} . Filtrace znamená, že na základě znalostí o vektoru \mathbf{z} můžeme odhadnout vektor \mathbf{x} .

Systém lze popsat rovnicemi:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

První rovnice je *stavová*, popisuje vývoj stavového vektoru \mathbf{x} . Druhá rovnice je rovnice *pozorování*, vyjadřuje vztah mezi vektorem pozorování \mathbf{z} a stavovým vektorem \mathbf{x} . \mathbf{w} je šum

systému a \mathbf{v} je šum měření. Předpokládáme, že šумы jsou vzájemně nezávislé. Matice \mathbf{A} vyjadřuje přenos neboli vztah mezi aktuálním a budoucím stavem. Matice \mathbf{H} popisuje vztah mezi měřenými veličinami a stavem systému. Index k je čas pro příslušný snímek, když se jedná o videosekvenci.

Kalmanův filtr pracuje ve dvou fázích: predikce a korekce, které tvoří cyklus. V predikční fázi se na základě předchozího měření odhaduje budoucí stav systému. Ve fázi korekce se s použitím hodnot současného měření a předchozí predikce odhaduje současný stav.

Rovnice popisující predikční fázi:

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \\ \mathbf{P}_k^- &= \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}\end{aligned}$$

Vektor $\hat{\mathbf{x}}_k^-$ je počáteční hodnota stavového vektoru, stříška označuje odhad. Matice \mathbf{P} je kovarianční matice chyby odhadu vektoru $\hat{\mathbf{x}}_k^-$, \mathbf{Q} je kovarianční matice šumu systému. Matice \mathbf{B} popisuje vliv řídicího vektoru \mathbf{u} na stav \mathbf{x} .

Rovnice korekce:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-\end{aligned}$$

Matice \mathbf{R} je kovarianční matice šumu měření. Matice \mathbf{K} je Kalmanovo zesílení, které funguje jako váha a určuje, jaké množství informace z měření bude akceptováno.

Pokud není měření k dispozici (jednotlivé objekty se překrývají, došlo ke ztratě objektu), můžeme použít odhad.



Obrázek 12: Příklad činnosti Kalmanova filtru, zelená tečka odpovídá odhadu

3 Metody segmentace popředí

Algoritmy segmentace pracují na principu modelování pozadí a následném porovnání vstupního snímku s modelem. Nalezené rozdíly mohou odpovídat pohybujícímu se objektu čili popředí. Pro objekt platí vzorec: $d(I_t, B) > T$, kde T je prah určující jak velká musí být změna pro přiřazení daného pixelu do popředí; I_t - hodnota pixelu současného snímku (skalár pro jasové složky, vektor pro barevné), B je model pozadí. d je funkce vzdálenosti, která může být vypočítána různými způsoby, například: [11]

Rozdíl hodnot

$$d_0 = |I_t - B_t| \quad (15)$$

Euklidovská vzdálenost, kde R,G,B 2.2 jsou červená, zelená a modrá složka:

$$d_2 = (I_t^R - B_t^R)^2 + (I_t^G - B_t^G)^2 + (I_t^B - B_t^B)^2 \quad (16)$$

Mahalanobisova vzdálenost, kde μ_t a Σ_t odpovídají střední hodnotě 2.1 barvy pixelu a kovarianční matici 2.1:

$$d_M = |I_t - \mu_t| \Sigma_t^{-1} |I_t - \mu_t|^T \quad (17)$$

Podle [11] d_0 je o něco méně efektivní, protože nepracuje s barvou.

Výstupem algoritmů segmentace jsou oblasti označené zpravidla bílou barvou, odpovídající objektům v pohybu - *maska*. Chceme, aby nalezené oblasti byly co nejvíc plynulé a maximálně odpovídaly vjemu člověka. Vstupní snímek a snímek po správné segmentaci neboli **Ground Truth (GT)** lze pozorovat na obrázku.



Obrázek 13: Vstupní snímek a maska, odpovídající vjemu člověka

Úskalí spočívá v tom, že počítač nedokáže vnímat scénu podobným způsobem, jako člověk. Proto se do masky občas započítává šum, změny barev způsobené světelnými podmínkami, stíny, pohyby na pozadí. A naopak se nezapočítávají oblasti, co mají stejnou barvu jako pozadí a nebo objekty, které se na nějakou dobu zastavily. Lepších výsledků lze dosáhnout použitím filtrů rozostření (Medián, Gauss 2.3) aplikovaných buď na každý vstupní snímek (v případě že systém nevyžaduje precizní výsledky a rozostřené obrázky na vstupu budou dostačující), nebo na výslednou masku. Tyto filtry odstraní šum a zmírní přechody mezi oblastmi s odlišnými barvami. Další možné vylepšení kvality masky je použití různých kombinací morfologických operací. 2.4

Hlavní rozdíl v algoritmech segmentace je způsob, jakým je modelováno pozadí. V následujících sekcích budou popsány různé přístupy k tomuto problému.

3.1 Základní (Basic)

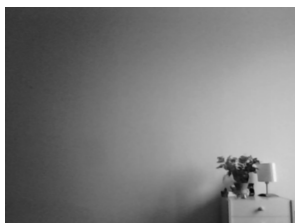
Základní metody jsou založeny na vytváření modelu pozadí (referenčního snímku) z existujících snímků a následném porovnávání každého pixelu referenčního a současného snímku.

Rozdíly metod spočívají ve způsobu nalezení referenčního obrázku - B . Může to být:

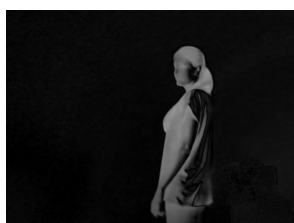
1. *Snímek scény bez pohybu*, často - první snímek ve videosekvenci (**Static Frame Difference (Static)**).

+ jednoduchý, intuitivní způsob

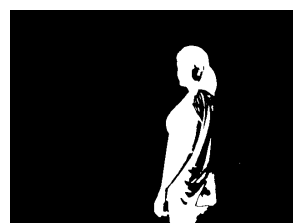
- model pozadí je statický, nedokáže se přizpůsobit k měnícím se podmínkám; velká citlivost na nejmenší změny; ne vždy máme takový snímek k dispozici



první snímek



rozdíl snímků



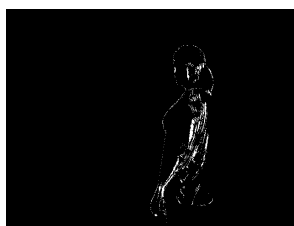
maska

Obrázek 14: Pozadí - první snímek, výsledky

2. *Předchozí snímek $B = I_{t-1}$* (**Frame Difference (FD)**)

+ adaptace na změny; objekty, které se zastavily nejsou nadále detekovány

- objekt popředí je tvořen jenom hranicemi



Obrázek 15: Pozadí - předchozí snímek, výsledek

3. *Průměr z N předchozích snímků* (**Mean Filter (Mean)**) [30] Aktualizace probíhá tak, že odstraníme nejstarší snímek a přidáme současný.

+ řeší problém, kdy nemáme snímek bez popředí k dispozici

- nárok na paměť (v každý okamžik potřebujeme $N * \text{velikost snímku}$); vznik falešných objektů



Obrázek 16: Pozadí - průměr z 10 předchozích snímků, výsledky



Obrázek 17: Pozadí - průměr z 50 předchozích snímků, výsledky

4. *Aktualizovaný model pozadí (Running Average (Adaptive))* [18] podle

$$B_{t+1} = (1 - \alpha)B_t + \alpha I_t \quad (18)$$

kde α je koeficient (0 až 1 [11]), který určuje rychlost aktualizace pozadí. Čím menší je α , tím menší vliv na model pozadí mají náhle změny.

- + dokáže reagovat na změny
- vznik falešných objektů pro velké hodnoty α



Obrázek 18: Aktualizované pozadí s parametrem $\alpha=0.1$, výsledky



Obrázek 19: Aktualizované pozadí s parametrem $\alpha=0.01$, výsledky

Výše uvedené algoritmy byly implementovány v jazyce Python pomocí knihovny OpenCV. Pracovala jsem s obrázky odstínů šedi. Ve svých experimentech jsem použila Gaussovský filtr s jádrem (5,5) 2.3, aplikovaný na každý vstupní snímek a následné zpracování masky morfologickými operacemi. Hodnota prahu (**Threshold**) je 25 (empirické nastavení, kompromis: objekt je co nejvíc vidět, není šum na pozadí). Použitá vzdálenost: d_0 .

Základní metody jsou velice citlivé na změny v obraze, a to i na nejmenší změny v osvětlení, občas i neviditelné okem (stíny, blikání žárovek). Jsou závislé na hodnotě parametrů: prahu, počtu snímků pro tvorbu referenčního obrázku a dalších. Může dojít například k "zapamatování" pozice objektu, který zůstal na stejném místě delší dobu a pak zase začal měnit polohu. A nebo také naopak k rychlému "zapomenutí" předchozí polohy, tím pádem k detekci jen části objektu, která je v daném okamžiku v pohybu.



Obrázek 20: Příklady chybné detekce popředí

Výše uvedené metody se zpravidla nepoužívají v základním provedení, ale v kombinaci s jinými metodami. Avšak mohou být dostačující pro ideální podmínky: vnitřní scéna s konstantním nebo pomalu se měnícím osvětlením.

3.2 Statistické (Statistical)

3.2.1 Gaussovské (Gaussian)

Running Gaussian Average (1G) Modelování pozadí pomocí snímků (přístupy z předchozí sekce) vyžaduje konstantní scénu bez šumu a artefaktů. V reálné situaci takový stav je těžko dosažitelný, proto někteří autoři navrhují modelovat každý pixel pozadí pomocí Gaussova rozdělení pravděpodobnosti 2.1 pro několik za sebou jdoucích snímků. [38] Pixel s nízkou hodnotou pravděpodobnosti (mimo Gaussovou křivku) je považován za popředí.

Grafy v této sekci znázorňují historii hodnot intenzity konkrétního pixelu videosekvence, histogramu a Gaussova rozdělení pro tyto hodnoty. Je vidět, že nejčastěji zastoupená hodnota intenzity je přibližně 150, Gaussovo rozdělení má vrchol právě v bodě kolem 150 (střední hodnota rozdělení).

Pro každý pixel ze série N obrázků počítáme střední hodnotu a rozptyl. 2.1 S každým novým snímkem parametry jsou aktualizovány podle následujících vzorců: [12]

$$\mu_t = (1 - \alpha)\mu_{t-1} + \alpha I_t \quad (19)$$

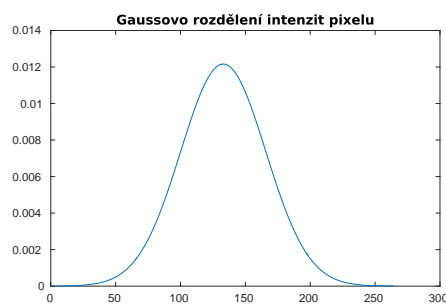
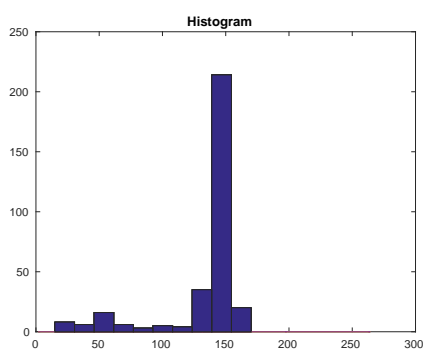
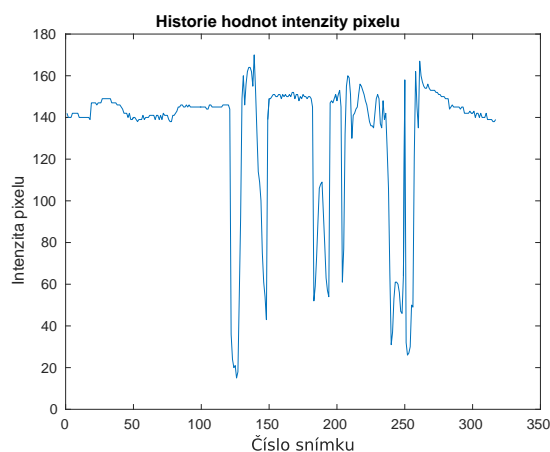
$$\sigma_t^2 = (1 - \alpha)\sigma_{t-1}^2 + \alpha(I_t - \mu_t)(I_t - \mu_t)^T \quad (20)$$

kde α je učící konstanta, I_t je současná hodnota pixelu.

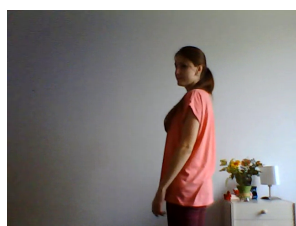
Následně probíhá kontrola podle:

$$|I_t - \mu_t| > T = k\sigma_t \quad (21)$$

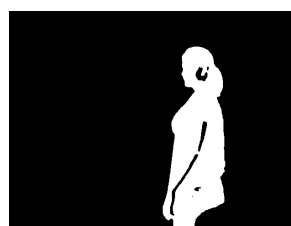
T je práh a počítá se jako násobek odchylky.



Obrázek 21: Příklad parametrů pro jeden konkrétní pixel videosekvence



vstupní snímek



maska

Obrázek 22: Výsledek segmentace pro metodu 1G, implementace pomocí knihovny [33]

Metoda má menší nárok na paměť. V každý okamžik potřebuje ukládat jenom 2 hodnoty: střední hodnotu μ_t a odchylku σ_t místo N předchozích hodnot daného pixelu. [30] Používá se zpravidla ve vnitřních scénách s pomalu se měnícími světelnými podmínkami. [12]

Možné vylepšení metody je použití HSV-barevného prostoru, který je více odolný proti pomalým změnám v osvětlení, protože odděluje od sebe barvu a jas. [12]

Mixture Of Gaussian (MOG) V předchozí metodě bylo použito jedno rozdělení. 2.1 Pro účel detekce vozidel Friedman a Russel [15] tento počet zvětšili a použili 3 předem určené oddělené rozdělení: pro barvu silnice, barvu vozidel a barvu stínů. Nejtmavší odpovídalo stínům, rozdělení s větším rozptylem 2.1 vozidlům, zbylé silnici. Místo modelování hodnot pixelů s použitím konkrétních rozdělení, Stauffer a Grimson [4] navrhli modelovat každý pixel obrázku samostatně za pomoci směsi gausiánů (**MOG**). Autoři tvrdí, že přístup umožňuje rychlé přizpůsobení modelu pozadí k měnícím se světelným podmínkám, opakujícím se pohybům elementů scény a ke změnám v geometrii scény (například parkování aut).

Podle hodnot rozptylu gausiánů a četnosti výskytu pixelů lze posoudit, který z nich odpovídá pozadí. Pokud rozptyl je nízký a pixely se objevují s vysokou četností, jedná se o pozadí, barva pixelu je uvnitř křivky rozdělení. Naopak pixely mimo oblasti gausiánů můžeme považovat za popředí. Metoda je náročnější na paměť, protože pro každý pixel potřebujeme více parametrů. Zatížení je úměrné počtů použitých gausiánů.

Zápis $\{X_1 \dots X_t\} = \{I(i) : 1 \leq i \leq t\}$ popisuje takzvaný ”**pixel proces**” - posloupnost hodnot určitého pixelu za čas t . X_i je skalár, když se jedná o obraz v barvách šedi nebo vektor pro barevný obraz. I je intenzita pixelu.

Minulé hodnoty pixelu jsou modelovány pomocí K (většinou 3 až 5 [4]) gausiánů, pravděpodobnost pozorování pixelu lze vyjádřit vztahem:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (22)$$

kde K je počet rozdělení, ω je váha: určuje, jak velkou část dat obsahuje konkrétní i -ty gausián (váhy jsou normovány, jejich součet je roven 1), μ je střední hodnota i -tého gausiánu směsi, Σ je kovarianční matice 2.1 i -tého gausiánu, η je funkce hustoty pravděpodobnosti:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (23)$$

n je dáno počtem barevných složek, pro RGB model 2.2 je 3.

Pro zjednodušení výpočtu vycházíme z předpokladu, že červený, zelený a modrý kanál jsou na sebe nezávislé a mají stejný rozptyl. Proto kovarianční matici můžeme považovat za diagonální, ve tvaru:

$$\Sigma_{k,t} = \sigma^2 I \quad (24)$$

kde I je jednotková matice.

Hodnota intenzity každého pixelu nového snímku je porovnávána s existujícími K rozděleními. Kontrola se provádí podle:

$$|X_t - \mu| < 2.5\sigma \quad (25)$$

Pro barevný obraz se kontrola provádí pro všechny tři složky. Pokud se jedná o pozadí, musí podmínka platit pro všechny barevné kanály. Jestli podmínka platí a rozdělení, do kterého byl pixel přidělen odpovídá pozadí, znamená to, že pixel patří do pozadí. Když rozdělení není pozadím, pixel patří do popředí.

Jestli podmínka neplatí ani pro jeden z K gausiánů, je vytvořeno nové rozdělení se střední hodnotou odpovídající hodnotě aktuálního pixelu X_t , s velkou hodnotou rozptylu a nízkou váhou. Protože v každý okamžik máme přesně K gausiánů, nejméně pravděpodobné

rozdělení (to, co má nejnižší váhu a největší rozptyl) je odstraněno. Váhy se přepočítávají podle:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \quad (26)$$

kde α je učící konstanta (**learning rate**), definuje rychlost změn parametrů rozdělení), $M_{k,t}$ je 1 pro případ shody a 0 pro zbývající.

Hodnoty μ a σ se přepočítávají pouze u gausiánu, u kterého došlo ke shodě v aktuální hodnotě pixelu:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (27)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (28)$$

kde

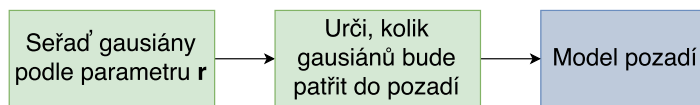
$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \quad (29)$$

To znamená, že když dojde k výskytu objektu nepatřícího do pozadí, model se pozmění tak, že nový objekt bude vyjádřen jedním z gausiánů. Pokud se objekt bude nacházet v určitém místě delší dobu, bude postupně zahrnout do pozadí.

Na základě parametrů rozptylu a četnosti výskytu jednotlivých gausiánů lze posoudit, které z gausiánů patří do pozadí a které do popředí. Jak již bylo uvedeno, pozadí je charakterizováno co nejmenší hodnotou rozptylu a co největší hodnotou četnosti výskytu. Pokud seřadíme gausiány podle parametru $r = \omega/\sigma$, dostaneme řadu, kde gausián s největší pravděpodobností patřící do pozadí bude na začátku, nejméně pravděpodobný na konci. Kolik gausiánů bude patřit do pozadí určíme pomocí následujícího vztahu:

$$B = \operatorname{argmin}_b \left(\sum_{k=1}^b \omega_k > T \right) \quad (30)$$

T je práh určující dělicí rovinu pro gausiány modelu pozadí. V případě volby malé hodnoty prahu T bude daná soustava schopna reagovat pouze na malé změny (šum, pomalá změna osvětlení).

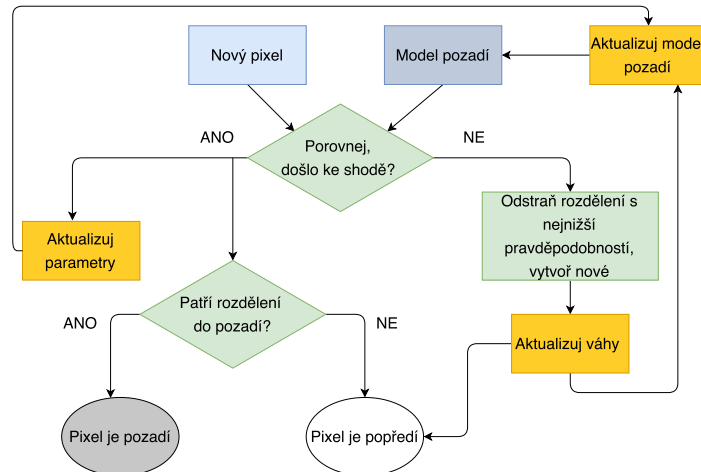


Obrázek 23: MOG, tvorba modelu pozadí

Výhodou metody je, že existující model pozadí nebude zničen, když dojde je změně. Barva ve směsi zůstává tak dlouho, dokud rozdělení nebude mít nejnižší pravděpodobnost a nebude nahrazeno novým.

Přístup může být použit ve venkovních scénách, kdy dochází k pomalým změnám v pozadí.

Bylo navrženo množství vylepšení daného algoritmu na úrovni parametrů, barevných modelů, způsobů detekce popředí atd. [12] Například, knihovna OpenCV (verze 3.0) má implementovanou vylepšenou verzi algoritmu. Výhodou modifikace je to, že algoritmus určuje potřebné množství rozdělení pro každý pixel, což zajišťuje lepší adaptaci k měnícím se podmínkám. Dokáže detekovat i stíny, které budou označeny šedou barvou.



Obrázek 24: MOG, schéma algoritmu [21]



Obrázek 25: Výsledek segmentace pro metodu MOG

3.2.2 Podprostorové (Subspace learning)

Analýza hlavních komponent (**Principal Components Analysis (PCA)**) je metoda vícerozměrné analýzy proměnných. Byla zavedena Karlem Pearsonem v roce 1901 a zobecněna Haroldem Hotellingem v roce 1933. [10]

Cílem analýzy hlavních komponent je transformace dat z původních proměnných do menšího počtu umělých dekokorelovaných proměnných - *hlavních komponent*. Komponenty jsou vzájemně nezávislé lineární kombinace původních proměnných. Hlavní komponenty vyjadřují rozptyl (proměnlivost) původních dat. Jsou seřazeny dle důležitosti: od největšího rozptylu k nejmenšímu. To znamená, že první hlavní komponenta obsahuje většinu informací o rozptylu původních dat, druhá pak menší množství a tak dále. Nejméně informací je soustředěno v poslední komponentě. Pro dostatečné vysvětlení variability původních proměnných stačí brát v úvahu jenom několik prvních hlavních komponent. Nedochozí tedy k velké ztrátě informací, protože nevyužité komponenty obsahují malé množství informace a odpovídají spíše šumu (jejich rozptyl je malý).

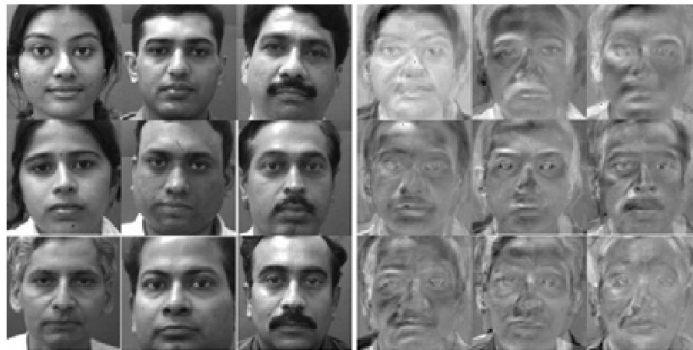
Analýza hlavních komponent se používá v oblasti rozpoznávání tváří a v oblasti komprese obrazů.

Obecný postup: [32]

1. získat vstupní data, spočítat jejich střední hodnotu; 2.1
2. odečíst střední hodnotu od každého datového souboru pro každou dimenzi (tak získáme data se střední hodnotou rovnou nule – normalizovaná data);

3. spočítat kovarianční matici 2.1 (je vždy čtvercová, $n \times n$, kde n odpovídá počtu dimenzí);
4. z kovarianční matice vypočítat vlastní čísla (n) a vlastní vektory (n) 2.1 (mají být jednotkové);
5. seřadit vlastní vektory podle hodnot vlastních čísel: od největšího (významného) k nejmenšímu (nevýznamnému). Vlastní vektor s nejvyšší hodnotou vlastního čísla je právě první hlavní komponenta daného souboru;
6. vybrat si jen m ($m < n$) prvních hodnot, to znamená, že budeme mít m -dimenzionální data. Dále zformovat vektor rysů (**Feature Vector**), což je matice, jejíž jednotlivé sloupce jsou vlastními vektory;
7. extrahovat nový datový soubor, normalizovat vstupní data (krok 2). Výsledná data získáme vynásobením vektoru rysů a matice nových normalizovaných dat. Původní osy (například x , y , z) mohou být nahrazeny vlastními vektory, procházející směrem maximálního rozptylu.

Dobrou vizualizací metody je reprezentace lidské tváře pomocí tzv. vlastních tváří (**Eigen Faces**). [35] Ze sady obrázků dle postupu spočítáme vlastní vektory. Když tyto vektory zobrazíme, zjistíme, že připomínají obličej člověka, proto se místo pojmu "vektor" používá "tvář". Nový obličej na vstupu lze poskládat z různě zastoupených jednotlivých vlastních tváří.



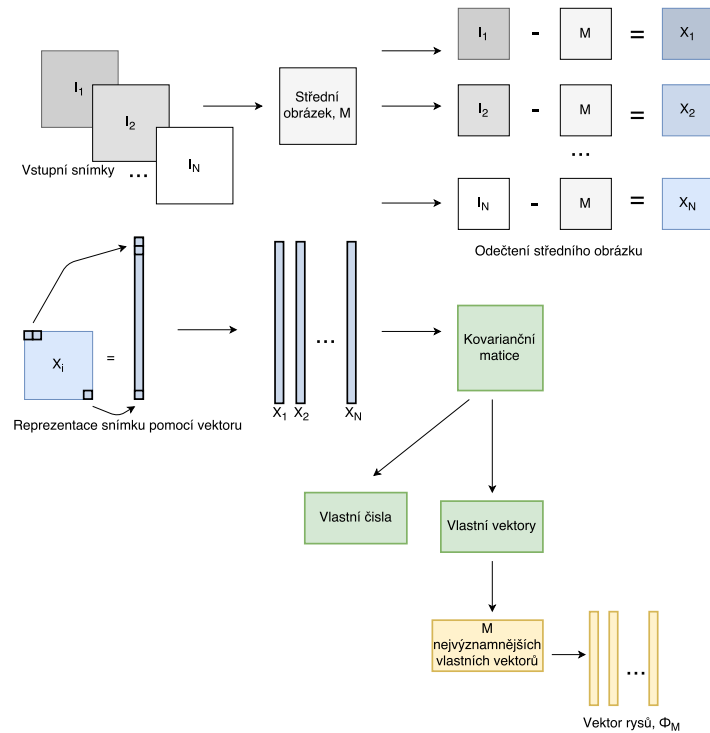
Obrázek 26: Vlastní tváře, sada obrázků na vstupu (vlevo) a vlastní vektory (vpravo) [22]

Analýzu hlavních komponent lze použít i k modelování pozadí. Oliver [29] konstruuje vlastní prostor (**Eigenspace**), který modeluje pozadí způsobem, popsaným výše, to znamená:

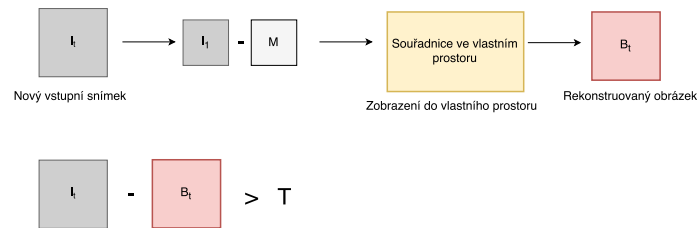
Vezmeme N vstupních snímků a spočítáme jejich střední hodnotu μ .

Od každého vstupního obrázku odečteme střední hodnotu, tím získáme normalizovaný vektor $X = [X_1 \dots X_N]$, kde $X_i = I_i - \mu$. Dále spočítáme kovarianční matici Σ , ze které dekompozicí získáme vlastní vektory a vlastní čísla a necháme si pouze M největších hodnot. Následně zformujeme vektor rysů Φ_M - matici obsahující M vlastních vektorů. Pomocí vektoru rysů budeme modelovat pozadí.

Jakmile máme nový vstupní snímek I_t , odečteme od něj střední hodnotu μ a zobrazíme do prostoru vytvořeného vlastními obrázky pozadí. Tím dostaneme souřadnice ve vlastním prostoru: $W_t = (I_t - \mu)^T \Phi_M$. Následně rekonstruujeme pozadí: $B_t = \Phi_M W_t^T + \mu$. Detekce popředí probíhá tak, že spočítáme vzdálenost mezi vstupním obrázkem a zobrazeným obrázkem. To znamená, že detekujeme to, co se přidalo "navíc". Objekt patří do popředí, když $|I_t - B_t| > T$.



Obrázek 27: PCA, tvorba vlastního prostoru



Obrázek 28: PCA, detekce popředí

Pohybující se objekty jsou typicky malé a neobjevují se na stejném místě v N snímcích, proto jejich příspěvek ve výsledném modelu není podstatný. To znamená, že vlastní prostor dobře modeluje statické pozadí, špatně pohybující se předměty popředí.



Obrázek 29: Výsledek segmentace pro metodu PCA

Daný algoritmus je postaven na maticových výpočtech, proto je implementován v prostředí MATLAB. K tvorbě vlastního prostoru jsem použila kolem 120 snímků a 3 vlastní vektory.

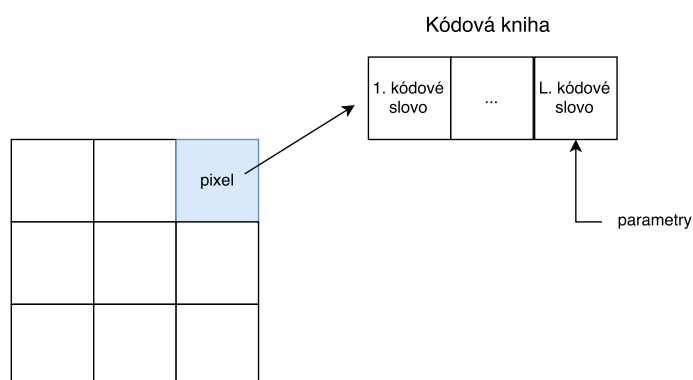
Výše popsána metoda má určitá omezení [17]: objekty popředí musejí být malé a ne-

vyskytovat se na stejném místě během tvorby vlastního prostoru; výpočetní náročnost aktualizace modelu; omezení na obrázky odstínů šedi (barevné obrázky vyžadují větší počet dimenzí); problém s měnícími se světelnými podmínkami. Lze použít ve venkovních scénách s malými objekty popředí.

3.3 Shlukové (Clusters)

Shlukové modely obecně jsou založeny na tom, že vstupní data lze na základě podobných vlastností roztrždit do jednotlivých skupin – *shluků*. Data s podobnými vlastnostmi by měly být co nejbliž k sobě, jednotlivé shluky co nejdál. Podobnost je určena pomocí vzdáleností jednotlivých objektů.

Codebook Princip metody spočívá v tom, že se pro každý pixel obrazu tvoří tzv. *kódová kniha* (**Codebook (CB)**), do které se ukládají *kódová slova* (**Codeword**) – hodnoty, které charakterizují pixel. [24]



Obrázek 30: Struktura kódové knihy

Máme trénovací sekvenci $X = \{X_1, X_2, \dots, X_N\}$ pro pixel, kde $X_1 \dots X_N$ jsou RGB vektory. $C = \{C_1, C_2, \dots, C_L\}$ je kódová kniha, $C_1 \dots C_L$ jsou kódová slova. Pro každý pixel může mít kódová kniha různou délku. Pixel ze stabilní oblasti snímku lze vyjádřit jedním kódovým slovem (jednou barvou). Zatímco pro pixel, který se nachází například v oblasti blízko stromu ve větru, potřebujeme tři barvy: zelenou pro listí, modrou pro oblohu, hnědou pro kůru.

Každé kódové slovo se skládá z RGB vektoru (zastoupení jednotlivých barevných složek, tvořících výslednou barvu) $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ a 6 parametrů charakterizujících kódové slovo $aux_i = \langle \check{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i \rangle$:

- \check{I}, \hat{I} : minimální, maximální hodnoty jasu, které mělo;
- f : frekvence, s jakou se objevovalo;
- λ : (**Maximum negative run-length**) maximálně dlouhý časový interval během kterého se neopakovalo;
- p, q : první a poslední časový okamžik kdy se objevilo.

Barevné a jasové složky jsou oddělené, aby si algoritmus byl schopen poradit se změnami v osvětlení.

Během trénování je každý pixel porovnán se současnou kódovou knihou. Tím určujeme, které kódové slovo mu odpovídá (jestli takové existuje).

Zařazení pixelu do popředí nebo pozadí se provádí na základě barevné vzdálenosti a jasů (jas se musí nacházet uvnitř přijatelných hranic). Pokud pixel nesplňuje tyto dvě podmínky (není v kódové knize), patří do popředí a je přidán do kódové knihy. Pokud odpovídá některému kódovému slovu, znamená to, že patří do pozadí.

Proces konstrukce kódové knihy vypadá následovně: [24]

1. $L \leftarrow 0$, $C \leftarrow$ prázdná množina (\leftarrow znamená přiřazení)
2. **for** $t = 1$ to N **do**
 - (a) $X_t = (R, G, B)$, $I \leftarrow R + G + B$
 - (b) Najdi kódové slovo C_m z $C = \{C_i | 1 \leq i \leq L\}$ odpovídající X_t podle podmínek:
 - 1) barevné zkreslení $(X_t, v_m) \leq \epsilon_1$
Barevné zkreslení se měří podle: $\delta = \sqrt{\|x_t\|^2 - \rho^2}$, kde $x_t = (R, G, B)$ je vstupní pixel, c_i je kódové slovo, $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$
 $\|x_t\|^2 = R^2 + G^2 + B^2$
 $\|v_i\|^2 = \bar{R}_i^2 + \bar{G}_i^2 + \bar{B}_i^2$
 $\langle x_t, v_i \rangle^2 = (\bar{R}_i R + \bar{G}_i G + \bar{B}_i B)^2$
 $\rho^2 = \frac{\langle x_t, v_i \rangle^2}{\|v_i\|^2}$
 - 2) jas $(I, \langle \hat{I}_m, \hat{I}_m \rangle) = \mathbf{true}$
 - (c) Jestli C je prázdná nebo nedošlo ke spojení, $L \leftarrow L + 1$. Vytvoř nové kódové slovo C_L pro které bude platit $v_L \leftarrow (R, G, B)$, $aux_L \leftarrow \langle I, I, 1, t - 1, t, t \rangle$.
 - (d) V jiném případě aktualizuj odpovídající kódové slovo sestavené z $v_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ a $aux_m = \langle \hat{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m \rangle$ nastavením:
 - 1) $v_m \leftarrow (\frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m + G}{f_m + 1}, \frac{f_m \bar{B}_m + B}{f_m + 1})$
 - 2) $aux_m \leftarrow \langle \min\{I, \hat{I}_m\}, \max\{I, \hat{I}_m\}, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \rangle$
3. Pro každé kódové slovo $C_i, i = 1 \dots L$ nastav $\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i - 1)\}$.

Detekce popředí pro aktuální pixel probíhá podle:

1. $x = (R, G, B)$, $I \leftarrow R + G + B$
2. Pro všechna kódová slova najdi kódové slovo C_m odpovídající X podle:
 - 1) barevná vzdálenost $(X, v_m) \leq \epsilon_2$, kde ϵ_2 je práh pro detekci;
 - 2) jas $(I, \langle \hat{I}_m, \hat{I}_m \rangle) = \mathbf{true}$
3. **popředí**, když nejsou spojení;
pozadí - v jiném případě.

Výhodou metody je menší nárok na paměť, model pozadí je kompaktní. Metoda může být použita ve venkovních scénách i s mírně dynamickým pozadím.

Příklady vylepšení algoritmu [13]: adaptivní aktualizace kódové knihy; dvouvrstvý model kódové knihy (jedna vrstva obsahuje současné snímky pozadí, druhá se používá k modelování nových snímků pozadí); použití jiných barevných prostorů.



Obrázek 31: Výsledek segmentace pro metodu CB, implementace pomocí [26]

4 Porovnání algoritmů segmentace

V předchozí sekci byly vysvětleny principy činnosti vybraných algoritmů, uvedeny ukázky výstupů na příkladu jedné videosekvence, stručně diskutovány silné a slabé stránky každé metody (většinou podle autorů).

Tohle však nestačí k relevantnímu hodnocení algoritmů. Ve většině reálných situací nemáme stálé pozadí: osvětlení se mění (a to jak postupně, tak i náhle); pozadí může obsahovat pohyb (listí stromů, pohyblivé schody); geometrie pozadí se postupně mění (parkování aut). Další problém je, že dochází k situaci, kdy část popředí má stejnou barvu, jako pozadí za ní (například kus oblečení postavy stejné barvy jako podlaha, skříň...). Potíže působí šum (pro videa, pořízená nekvalitní kamerou) a stíny, které mění svoji polohu spolu s předmětem. Algoritmus by měl být schopen si poradit i s takovými případy.

K hodnocení kvality algoritmů se zpravidla používají snímky obsahující správně označené popředí - **GT**. Každý pixel snímku se po segmentaci nějakým algoritmem porovnává s odpovídajícím pixelem snímku po správné segmentaci. Rozdíly se započítávají do příslušných parametrů:

- **True Positives (TP)** - pixely, které byly zahrnuty do popředí, ve skutečnosti patří popředí;
- **True Negatives (TN)** - pixely, které byly zahrnuty do pozadí, ve skutečnosti patří pozadí;
- **False Positives (FP)** - pixely, které byly zahrnuty do popředí, i když ve skutečnosti jsou pozadím;
- **False Negatives (FN)** - pixely, které nebyly zahrnuty do popředí, i když ve skutečnosti jsou popředím.

Představme si, že máme dva binární obrázky (s hodnotami pixelů 0-pozadí a 255-popředí): náš snímek (v tomto případě snímek po segmentaci nějakým algoritmem) - F a správný - GT . Princip zařazení pixelu do jedné z kategorií popsaných výše je zřejmý z následující tabulky:

		F	
		255	0
GT	255	TP	FN
	0	FP	TN

Jakmile máme všechny hodnoty, můžeme vypočítat následující parametry:

- *Přesnost* (**Precision**) neboli počet správně detekovaných pixelů popředí dělený celkovým počtem pixelů detekovaných jako popředí:

$$Precision = \frac{TP}{TP + FP} \quad (31)$$

- *Výtěžnost, úplnost* (**Recall**) neboli počet správně detekovaných pixelů popředí dělený počtem pixelů popředí, který by měl být vrácen (správný počet):

$$Recall = \frac{TP}{TP + FN} \quad (32)$$

- *Úspěšnost* (**Accuracy**) neboli počet správně detekovaných pixelů dělený celkovým počtem pixelů:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (33)$$

Existuje i řada dalších, používaných například statistické parametry, specifčnost vnímání, rozložení chyb: F-score, PSNR (Peak Signal–Noise Ratio), SSIM (Structural Similarity), D-Score. [34] Avšak v následujícím hodnocení budou použity tři výše uvedené.

K výpočtům jsem si vybrala 4 různá videa z veřejně dostupných databází: "Perception Test Images Sequences" [7] a "Wallflower Test Images Sequences" [9]. Výhodou je, že kromě samotné videosekvence soubory obsahují i několik obrázků správné segmentace popředí. Zvolila jsem si videa se scénáři, dělající algoritmům problémy: pohybující se listí stromů; náhlé změny v osvětlení; shodné barvy předmětů popředí a pozadí; objekty popředí, které se zastavily; video, neobsahující "čisté" pozadí.

Na videa jsem aplikovala 6 různých algoritmů:

- Základní:
 - Statické pozadí - první snímek videosekvence (Static) 1;
 - Aktualizované pozadí (Adaptive) 4;
- Gaussovské:
 - Jeden gausián (1G) 3.2.1;
 - Směs gausiánů (MOG) 3.2.1;
- Podprostory: Vlastní pozadí (PCA) 3.2.2;
- Shlukové: Kódová kniha (CB) 3.3.

K segmentaci popředí jsem použila program *UMD Background Subtraction* od autorů [26] (pro metodu kódová kniha) a knihovnu *BGSlibrary* [33] (pro zbytek) a to z důvodu větší relevantnosti výsledků a zamezení vlivu chyby programátoru (většina algoritmů je implementována v jednom prostředí, jediným stylem).

Príslušné parametry pro algoritmy byly nastaveny podle autorů: [34], [25].

Dále jsem postupně ve dvojicích porovnávala výstupy algoritmů s obrázky správné segmentace. Následně, pro každou videosekvenci jsem vypočítala příslušné parametry, a to podle postupů uvedených výše.

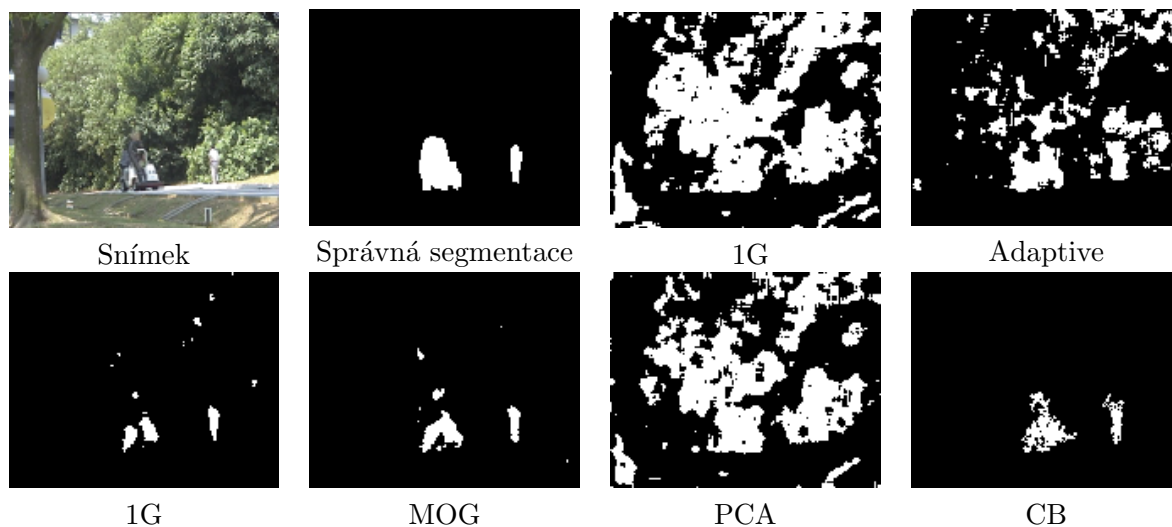
V následujících odstavcích uvedu příklady výstupů.

	Static	Adaptive	1G	MOG	PCA	CB
<i>parametry</i>	$T = 15$	$T = 15$ $\alpha = 0.5$	$T = 12.15$ $LF = 30$ $\alpha = 0.05$	$T = 9$ $\alpha = 0.05$ $n = 3$	$T = 255$ $HS = 10$ $Dim = 10$	$T = N/2$

Tabulka 1: Parametry algoritmů, kde T je práh, α je učící konstanta, LF je počet snímku pro trénování, n je použitý počet Gaussových rozdělání, HS je velikost historie, Dim je počet dimenzí, N je počet trénovacích snímků

1. *Kampus* (Campus with wavering tree branches)

Máme video silnice, po které jezdí auta a chodí chodci. Podél silnice rostou stromy. Stromy ve větru jsou klasickým příkladem dynamického pozadí, které působí problém během detekce popředí.



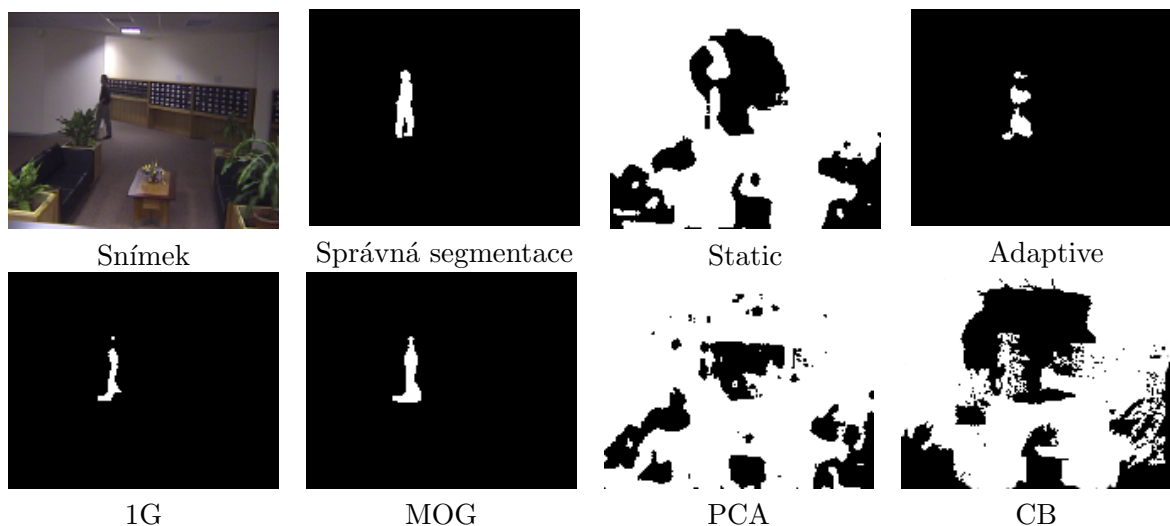
Obrázek 32: Výsledky segmentace pro videosekvenci Kampus

	Static	Adaptive	1G	MOG	PCA	CB
<i>TN</i>	119638	189297	196801	196281	138303	189745
<i>TP</i>	6592	5576	4395	5371	6484	4024
<i>FP</i>	78264	8605	1101	1621	59599	8157
<i>FN</i>	306	1322	2503	1527	414	2874
<i>Precision</i>	<i>0.07768</i>	<i>0.3932</i>	<i>0.79967</i>	<i>0.76816</i>	<i>0.09812</i>	<i>0.33035</i>
<i>Recall</i>	<i>0.95564</i>	<i>0.80835</i>	<i>0.63714</i>	<i>0.77863</i>	<i>0.93998</i>	<i>0.58336</i>
<i>Accuracy</i>	<i>0.61636</i>	<i>0.95153</i>	<i>0.9824</i>	<i>0.98463</i>	<i>0.70697</i>	<i>0.94614</i>

Tabulka 2: Kampus, výsledky pro 10 dvojic snímků 160x128

2. *Světla* (Lobby in an office building with switching on / off lights)

Další video je použito pro demonstraci vlivu změny světelných podmínek. Dochází tedy k vypínání a následnému rozsvícení světel.



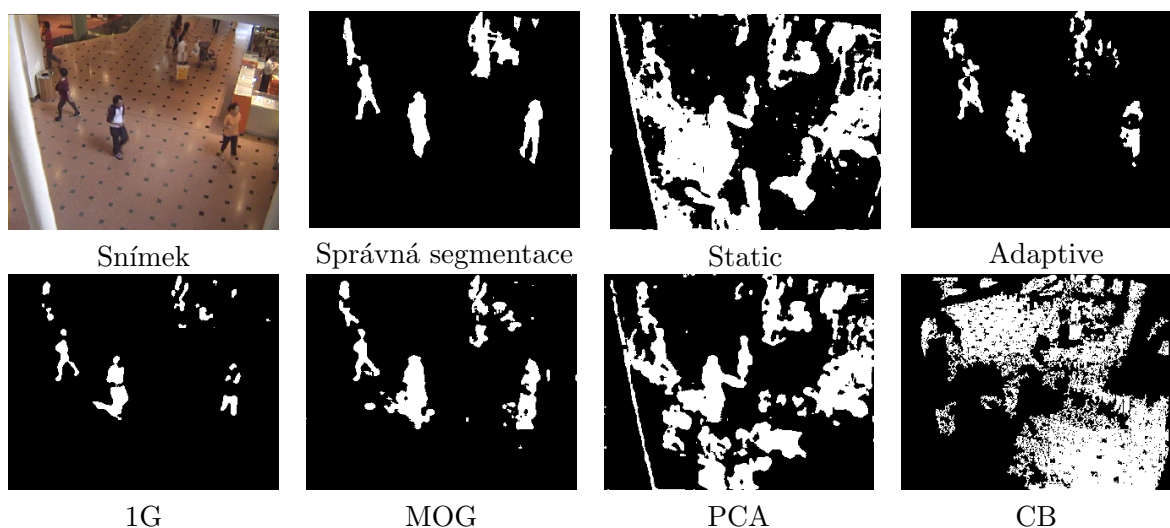
Obrázek 33: Výsledky segmentace pro videosekvenci Světla

	Static	Adaptive	1G	MOG	PCA	CB
<i>TN</i>	167226	199245	195086	199425	167226	78954
<i>TP</i>	3154	748	772	1352	3154	3596
<i>FP</i>	32993	974	5133	794	32993	121265
<i>FN</i>	1427	3833	3809	3229	1427	985
<i>Precision</i>	<i>0.08725</i>	<i>0.43438</i>	<i>0.13074</i>	<i>0.63001</i>	<i>0.08725</i>	<i>0.0288</i>
<i>Recall</i>	<i>0.6885</i>	<i>0.16328</i>	<i>0.16852</i>	<i>0.29513</i>	<i>0.6885</i>	<i>0.78498</i>
<i>Accuracy</i>	<i>0.83193</i>	<i>0.97653</i>	<i>0.95634</i>	<i>0.98036</i>	<i>0.83193</i>	<i>0.40308</i>

Tabulka 3: Světla, výsledky pro 10 dvojic snímků 160x128

3. Obchod (Shopping center)

V následující videosekvenci máme hodně objektů popředí. Některé objekty velice pomalu mění svoji polohu, zůstávají na jednom místě delší dobu. Další problém je, že video neobsahuje snímky "čistého" pozadí neboli pozadí bez pohybu.



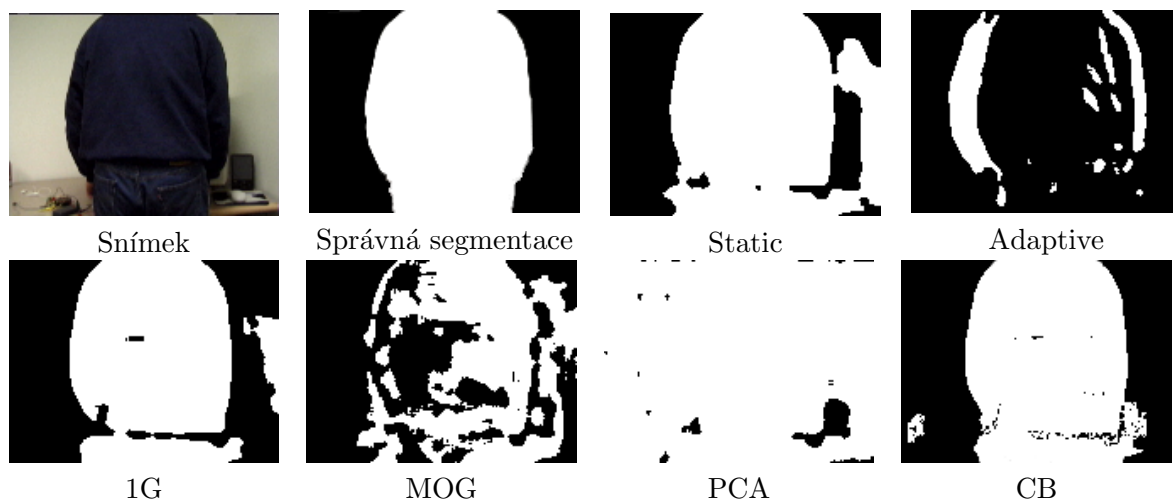
Obrázek 34: Výsledky segmentace pro videosekvenci Obchod

	Static	Adaptive	1G	MOG	PCA	CB
<i>TN</i>	535773	748580	765971	749164	616310	492405
<i>TP</i>	42754	23319	22472	31009	42468	20082
<i>FP</i>	232421	19614	2223	19030	151884	275789
<i>FN</i>	8252	27687	28534	19997	8538	30924
<i>Precision</i>	<i>0.15537</i>	<i>0.54315</i>	<i>0.90998</i>	<i>0.6197</i>	<i>0.21851</i>	<i>0.06787</i>
<i>Recall</i>	<i>0.83822</i>	<i>0.45718</i>	<i>0.44058</i>	<i>0.60795</i>	<i>0.83261</i>	<i>0.39372</i>
<i>Accuracy</i>	<i>0.70621</i>	<i>0.94226</i>	<i>0.96245</i>	<i>0.95236</i>	<i>0.80417</i>	<i>0.62559</i>

Tabulka 4: Obchod, výsledky pro 10 dvojic snímků 320x256

4. Maskování (Camouflage)

Následující videosekvence znázorňuje maskování. Vidíme blikající obrazovku počítače, před kterou se postaví muž.



Obrázek 35: Výsledky segmentace pro videosekvenci Maskování

	Static	Adaptive	1G	MOG	PCA	CB
<i>TN</i>	6528	7203	7820	6596	531	7924
<i>TP</i>	9998	930	9656	6314	10168	9281
<i>FP</i>	2107	1432	815	2039	8104	711
<i>FN</i>	270	9338	612	3954	100	987
<i>Precision</i>	<i>0.82594</i>	<i>0.39373</i>	<i>0.92217</i>	<i>0.7559</i>	<i>0.55648</i>	<i>0.92884</i>
<i>Recall</i>	<i>0.9737</i>	<i>0.09057</i>	<i>0.9404</i>	<i>0.61492</i>	<i>0.99026</i>	<i>0.90388</i>
<i>Accuracy</i>	<i>0.87425</i>	<i>0.43025</i>	<i>0.92451</i>	<i>0.68296</i>	<i>0.56599</i>	<i>0.91017</i>

Tabulka 5: Maskování, výsledky pro dvojici snímků 160x120

Provedené hodnocení mělo ryze demonstrační charakter. Dosažené výsledky nejsou postačující pro relevantní analýzu robustnosti algoritmů. Úplné hodnocení by mělo být prováděno na mnohem větším počtu různě kvalitních a dlouhých videosekvencí, znázorňujících rozmanité situace, které mohou nastat v reálném životě. Navíc metoda CB je implementována ve vlastním programu. Avšak precizní hodnocení nebylo cílem této práce. Výsledky kompletních hodnocení zájemci najdou například v člancích [34], [13], [11].

Nicméně, lze dospět k některým závěrům. Je zřejmé, že parametry *Precision*, *Recall* a *Accuracy* pro kvalitní algoritmus by měly být v každé situaci co největší. Teď se podíváme podrobněji na každou videosekvenci.

Video "Kampus" mělo znázornit vliv dynamického pozadí na vybrané algoritmy. Nejlepších výsledků dosáhly statistické přístupy: 1G a MOG. Další video "Světla" obsahuje náhlé změny v osvětlení. Pomocí tohoto videa lze hodnotit schopnost přístupů přizpůsobit se měnícím se podmínkám. Z pouhého pozorování a kvantitativních výsledků lze říct, že nejrychlejší adaptaci prokázaly algoritmy Adaptive a MOG. Následuje scéna z obchodu, kde nemáme k dispozici snímky pozadí bez pohybu a dochází k zastavení objektů popředí. Další potíž působí stíny a lesklá podlaha. Zvítězily statistické metody. Poslední situaci je maskování. Muž se postaví před obrazovkou počítače. Nejlepších výsledků dosáhly metody 1G a CB. Překvapivě jsem zjistila, že relativně komplikovaná metoda PCA nedosáhla citelně lepších výsledků, než základní metody. Mohlo to být způsobeno nevhodným nastavením parametrů.

Často rozhodujícím parametrem při výběru algoritmu je jeho výpočetní rychlost. Byl zjištěn čas v ms, potřebný pro zpracování každého snímku videosekvence. Static je pochopitelně nejrychlejší metoda. Nejpomalejší je MOG, čas ke zpracování je několikanásobně větší než u ostatních přístupů. Je důležité podotknout, že výsledky platí pouze pro parametry uvedené výše.

Potřebné množství paměti pro modelování pozadí je závislé na parametrech konkrétního algoritmu. Například MOG potřebuje K krát víc než 1G, kde K je počet použitých rozdělení. Velikost paměti pro PCA je úměrná počtu použitých vlastních vektorů, pro CB - počtu kódových slov. Základní metody jsou nejméně náročné na paměť.

Citelně lepších výsledků lze dosáhnout precizním nastavováním parametrů algoritmu pro každou konkrétní situaci. Příklady vylepšení po zvolení vhodných parametrů lze vidět v [34].

	Static	Adaptive	1G	MOG	PCA	CB
<i>Precision</i>	<i>0.28656</i>	<i>0.44112</i>	<i>0.69064</i>	<i>0.69344</i>	<i>0.24009</i>	<i>0.33897</i>
<i>Recall</i>	<i>0.86401</i>	<i>0.37985</i>	<i>0.54666</i>	<i>0.57416</i>	<i>0.86284</i>	<i>0.66648</i>
<i>Accuracy</i>	<i>0.75719</i>	<i>0.82514</i>	<i>0.95643</i>	<i>0.90008</i>	<i>0.72727</i>	<i>0.72125</i>

Tabulka 6: Celková úspěšnost algoritmů

5 Instalace

5.1 Popis instalace

Cílem interaktivní instalace *"Hear me"* je zprostředkovat jejím návštěvníkům kontakt napříč časem, pocit z prožitku něčeho už uplynulého a možnost nevědomě zanechat svoji "stopu" pro další návštěvníky. Projekt je výsledkem spolupráce několika lidí: Anastasia Kozítsyna (autorka této bakalářské práce, ČVUT FEL - sledování návštěvníků), Karel Antonín (bývalý student ČVUT FEL - tvorba prostorového zvuku), Jiří Nižník (FAMU - nahrávání zvuku).

Instalace je založena na interakci jednoho návštěvníka s imaginární postavou (i postavami), vytvořenou pouze zvukem, v uzavřeném prostoru. Imaginární postava bude simulována pomocí prostorového zvuku, k její ztvárnění bude využito záznamu kroků a dalších projevů předchozího účastníka. Zvuk bude následně k návštěvníkovi přenášen přes prostorová sluchátka. Zajímá nás reakce člověka, jehož smyslové vjemy budou silně omezeny, na takovou interakci.

Všechno se bude odehrávat v uzavřené místnosti velikosti přibližně 4,5x4,5m s jedním vchodem (dveře). Místnost bude relativně tmavá, jen na jedné její straně bude umístěn slabý zdroj světla.²

Před vchodem do místnosti na návštěvníka budou nasazeny nezbytné senzory:

- výstroj na hlavu – *helma*, sloužící několika účelům:
 1. umístění sluchátek a zvukové techniky pro snímání zvuku (uvnitř helmy) a systému pro sledování pohybu (seshora helmy);
 2. omezení pohybu hlavy návštěvníka pro snadnější sledování;
 3. omezení vidění;
 4. psychologický efekt. Předpokládáme, že se návštěvníci budou kvůli těmto omezením pohybovat pomalu a obezřetně.
- chránič určitého typu se zvukovou záznamovou technikou, který bude upevněn u chodidla.

Po nasazení výše uvedených zařízení návštěvník dostane pouze jednu instrukci: "Jdi za světlem". Následně mu bude pokynuto, aby vešel do místnosti, a zavřou se za ním dveře. Po celou dobu v místnosti (2-3 minuty) budou snímány a zaznamenány:

- poloha a úhel natočení hlavy návštěvníka;
- zvuk kroků - mikrofonom na noze, případné další reakce (hlasitý dech, smích, polknutí...) – mikrofonom uvnitř masky.

Informace se využijí pro ztvárnění další imaginární postavy, se kterou bude interagovat následující účastník.

²Původní představa se lišila: chtěli jsme mít nasvícenou místnost (z důvodu možnosti použití levnější, méně kvalitní kamery), ale úplně omezit vizuální vjem člověka, a to prostřednictvím přední části helmy (o helmě - v dalším textu). Avšak, po několika pokusech se zjistilo, že by se člověk v cizím prostoru bez možnosti vidět necítil pohodlně, navíc by neměl žádný orientační bod. Interaktivní instalace je věc zaměřená na návštěvníka, jako provozovatelé musíme v první řadě zajistit pohodlí a bezpečí. Z těchto důvodů bylo rozhodnuto upustit od původní představy a vydat se jiným směrem. Tak vznikla současná podoba průběhu instalace.

Prvních několik desítek sekund nebude spuštěn žádný zvukový výstup. Cílem je nechat člověka přizpůsobit se prostředí, navyknout si na místnost tak, aby polevila počáteční nervozita. Ne však až tak dlouho, aby se začal nudit.

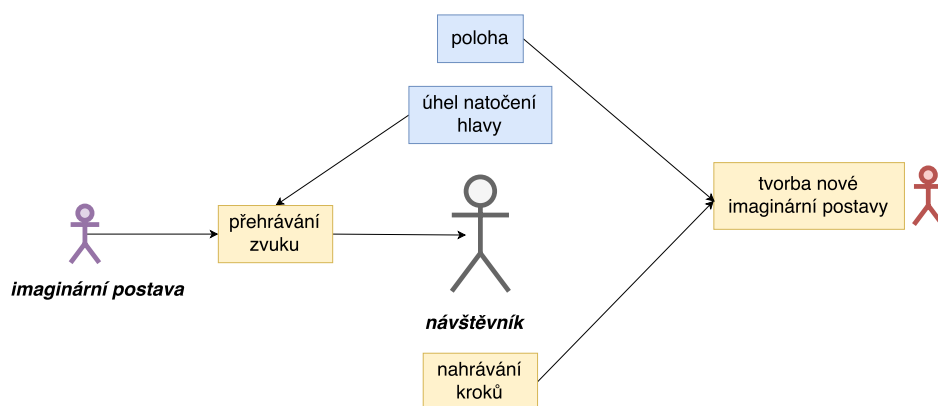
Začátek další části bude oznámen spuštěním zvuku otevření dveří (aniž by se tyto dveře v místnosti opravdu otevřely). To bude znamenat, že do místnosti "vpustíme" imaginární postavu - stopu předchozího návštěvníka. Návštěvník bude mít nyní možnost sledovat její pohyb prostřednictvím naslouchání zvuku v prostoru. Teď už je na něm, jak se zachová. (Jestli se bude snažit postavě vyhnout a nebo naopak: kráčet ve šlépějích, sledovat její pohyb.) Po určitém, stále se zmenšujícím časovém intervalu přidáváme další postavy. V poslední fázi se k postavám přidává i dronový zvuk, který graduje a postupně se přibližuje k posluchači, až se dostává na téměř neúnosnou hranici. V tomto momentu všechna reprodukce i záznam utichá a nakonec otevřeme dveře.

Jinými slovy: Po odchodu návštěvníka z místnosti bude i jeho stopa (pozice a zvuk, nahrávaný během jeho návštěvy) přidán do paměti instalace, takže nyní bude i on jeden z mnoha návštěvníků, kteří se budou neviditelně procházet onou místností.

Shrnutí:

Průběh jedné iterace instalaci lze popsat v několika bodech:

1. *Příprava* (před vstupem do místnosti) - umístění potřebných zařízení, sdělení pokynů;
2. *Vpuštění návštěvníka do místnosti* - není spuštěn žádný signál, návštěvník si navyká na místnost;
3. *"Vpuštění" imaginární postavy* - interakce návštěvníka se zvukovou stopou;
4. *Ukončení* - ukončení zvukového výstupu, otevření dveří.



Obrázek 36: Instalace, schématické znázornění

5.2 Obrazová část

Popis řešení Sledování návštěvníka v místnosti je jedním z kroků, potřebných ke správnému fungování instalace. V každý okamžik výstupy jsou:

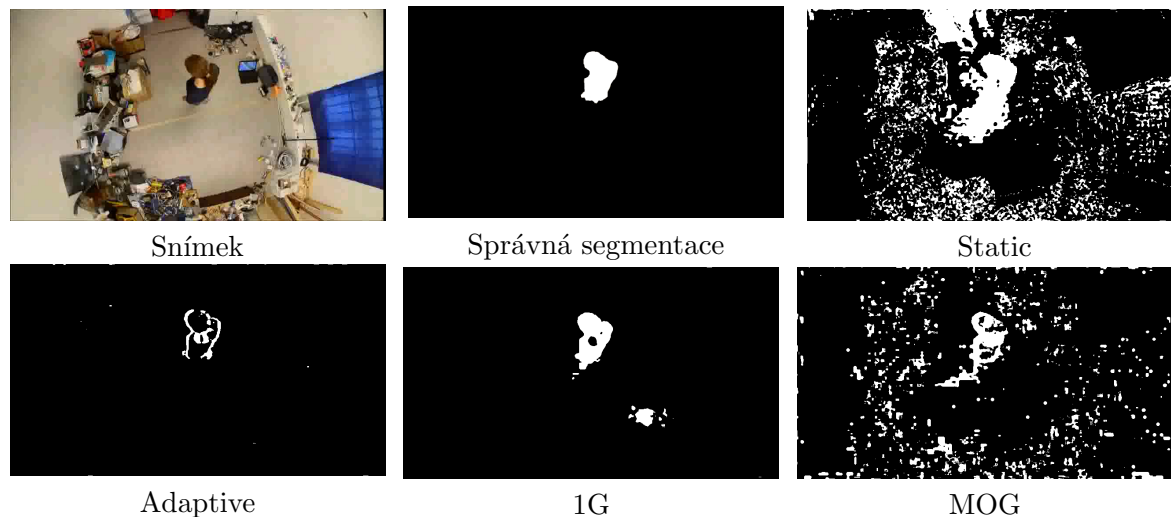
- poloha člověka (x,y souřadnice);
- úhel natočení hlavy

Tyto údaje jsou využity k výpočtu prostorového zvuku.

Zpracování zvuku a obrazu zajišťuje jediný program, napsaný v jazyce C++ s použitím externích knihoven. Nejpodstatnější požadavek kladený na celý systém, je fungování v reálném čase, bez zpoždění. Malé zpoždění by nebylo problémem při pomalém pohybu návštěvníka v místnosti. Avšak když dojde k náhlému otočení (například, návštěvník uslyší zvuk otevírajících se dveří, přirozenou reakcí je podívat se do směru, odkud zvuk přichází), zpožděný zvuk by nepůsobil věrohodně. Obrazová část by měla zatěžovat co nejmíň. Zároveň musí být stabilní a nenáchylná k možným malým změnám v osvětlení.

Jak již bylo uvedeno v popisu instalace, původní představa o vzhledu místnosti se lišila od finální: místo relativně tmavé, místnost měla být konstantně nasvícena. Možným řešením pro trasování bylo použití algoritmů segmentace popředí z předchozích sekcí a následný výpočet souřadnic návštěvníka. Získání doplňkové informace v podobě úhlu natočení hlavy by mohlo být vyřešeno pomocí pokrývky hlavy se speciálním vzorem, jednoznačně určujícím stranu obličeje a zátylku.

První část experimentu se týkala aplikace algoritmů segmentace na video pořízené v místnosti. Níže jsou představeny výsledky pro několik algoritmů.



Obrázek 37: Video z místnosti, výsledky

To, že výsledky nejsou uspokojivé, lze posoudit pouhým pohledem. První potíž spočívá v tom, že v místnosti jsou nainstalovány zářivky, jejichž blikání není vidět okem, ale lze pozorovat na videozáznamu. Toto blikání působí periodické změny hodnot intenzit pixelů, proto dochází k falešné detekci popředí. Další nesnáze přináší způsob pohybu návštěvníka: pomalu mění svoje polohu, delší dobu zůstává na stejném místě. Často proto přestává být detekován a je zařazen do pozadí.

Vzhledem k provedeným změnám v koncepci instalace a po dohodě s vedoucím práce byl zvolen jiný způsob segmentace objektů zájmu, a to podle barvy. Ukázalo se, že tento

přístup odpovídá požadavkům, uvedeným výše: je rychlý a stabilní. Máme ideální podmínky: známé prostředí, možnost manuálního ovládní kamery, možnost nastavovat a průběžně kontrolovat potřebné parametry.

Předložené řešení je popsáno v následujících odstavcích.

Vybavení:

- Kamera Nikon 7000D, objektiv Sigma rybí oko 10mm
- Hardware pro sledování návštěvníka
- USB-prodlužovací kabel, 5m

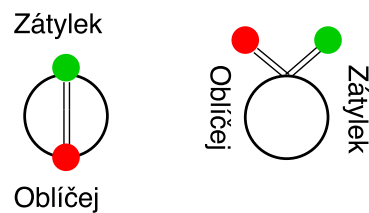
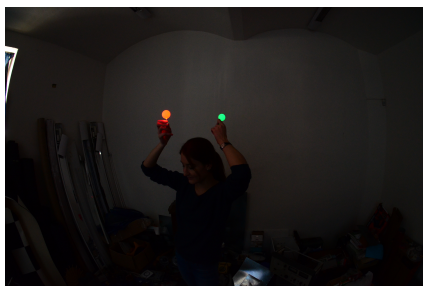
Proces získávání potřebných informací lze rozdělit do jednotlivých kroků.

0. Výroba hardware³

Materiály:

- Diody červená (2.1V; 50mA) a zelená (3.2V; 25mA)
- Odpor 33R Ohm a 47R Ohm
- Krabička pro baterky 2xAA + vypínač
- Dráty, bužírka
- Míčky pro ping-pong

Každá dioda bude umístěna uvnitř bílého pingpongového míčku, sloužícího k rozptylu světla, tím získáme 2 různobarevné konstantně nasvícené body pro trasování objektu. Svítící míčky na nevelkých tyčkách budou umístěny seshora helmy v konstantní vzdálenosti, tak, že zelená barva bude označovat zadní stranu hlavy a barva červená obličej.



Obrázek 38: Ukázka umístění míčků

1. Kalibrace kamery

K zachycení obrazu se používá zrcadlová kamera s objektivem rybí oko (**Fish eye**), připevněná na strop uprostřed místnosti. Dané umístění je výhodné z toho důvodu, že se celá místnost nachází v zorném poli kamery, což znamená, že není potřeba snímat obraz dvěma kamerami a vytvářet stereo nebo uměle omezovat prostor pro návštěvníka. Dále oproti umístění kamery v rohu místnosti nemusíme řešit prostorové transformace obrazu. Kamera je přes USB připojena k počítači. Software *SparkoCam* [8] umožňuje použití kamery pro přenos videa v reálném čase (zrcadlová kamera funguje jako web kamera).

³Bude probíhat současně s výrobou helmy. Zatím zkusíme s "mobilní" verzí: diody uvnitř míčků jsou napájeny z 3V baterek.

Kladnou stránkou rybího oka je velké zorné pole, avšak tyto objektivy nezachycují obraz tak, jak ho vidí člověk. Vady, které způsobují nepřesnosti zobrazení nazýváme *distorze* neboli zkreslení objektivu. Pro rybí oko jsou hodnoty zkreslení velké, pro přesnější výsledky je potřeba tyto vady odstranit a nebo aspoň zmírnit - zkalibrovat.

Distorze má dvě složky: radiální a tangenciální. *Radiální* distorze je posun polohy zobrazeného bodu od správné. Velikost odchylky se mění s radiální vzdáleností od středu objektivu. Kvůli radiálnímu zkreslení se rovné přímky stávají zakřivené. *Tangenciální* zkreslení souvisí s centrací čoček, běžně se neuvažuje, předpokládá se, že u kvalitních objektivů je jeho vliv zanedbatelný. [36]

Model objektivu rybího oka lze popsat rovnicemi. Zkreslení je ve tvaru:

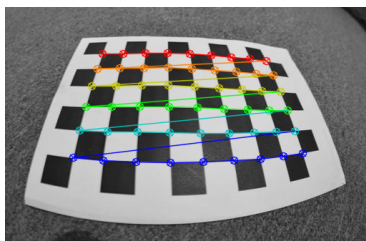
$$\theta_d = \theta(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8) \quad (34)$$

kde k_i jsou koeficienty distorze, $\theta = \text{atan}(r)$, $r = \sqrt{(x/z)^2 + (y/z)^2}$. x, y, z jsou souřadnice objektu v prostoru. Zkreslené (pozorované) souřadnice $[x'; y']$, kde $x' = (\theta_d/r)x$, $y' = (\theta_d/r)y$.

Ke kalibraci jsem použila standardní skript knihovny OpenCV `camera_calibrate.cpp`. Vstupem jsou snímky šachovnice, pořízené pod různým úhlem a na různých vzdálenostech od kamery a vlastní parametry šachovnice: velikost čtverců, počet rohů v obou směrech. Principem je získávání významných bodů na obrázku (v našem případě jsou to rohy čtverců) a porovnání vzdáleností mezi těmito body v obrazovém a reálném prostoru. Čím víc snímků program zpracuje, tím přesnější bude kalibrace. Výstupem je matice kamery (**camera matrix**) obsahující parametry čoček f_x, f_y a optických center c_x, c_y ve tvaru:

$$\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

a koeficienty distorze (**distortion coefficients**). Tyto hodnoty jsou následně využity jako vstupní parametry funkcí k odstranění zkreslení. [3]



šachovnice s body



původní snímek



snímek po kalibraci

Obrázek 39: Příklad kalibrace rybího oka

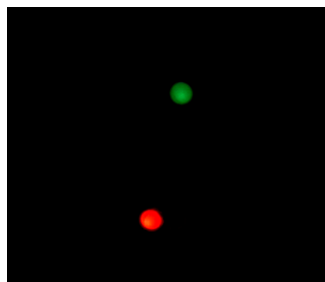
2. Vytváření masky

Použití kvalitní kamery umožňuje manuální nastavení expozice tak, že ve výsledném obrazu budou vidět jen dva zmíněné barevné body: červený a zelený.

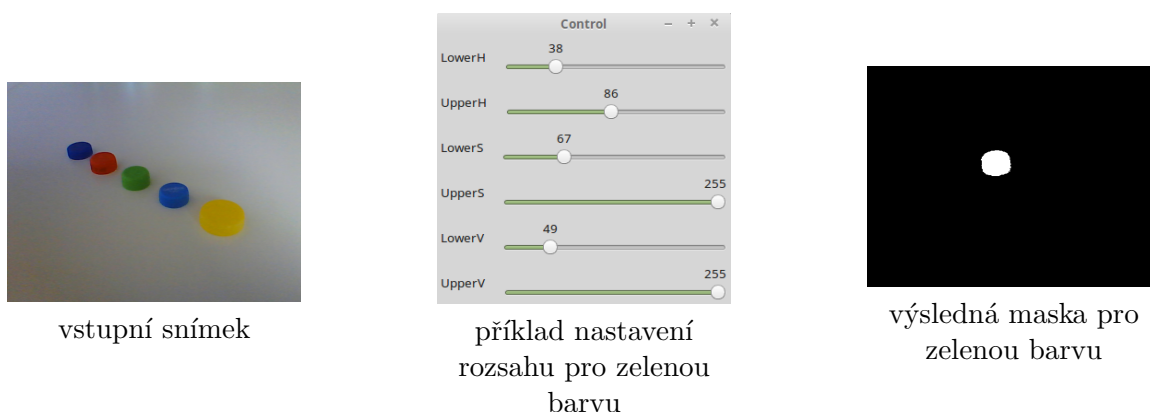
Prvním krokem je převod snímku do HSV prostoru. 2.2

Následuje definice rozsahů barev. K tomuto účelu jsem vytvořila pomocný program, umožňující nastavovat každý parametr zvlášť v reálném čase. Cílem je vytvořit masku, která co nejpřesněji pokrývá objekt zájmu a eliminuje jiné objekty a šum.

Jakmile máme definované rozsahy, pomocí funkcí **inRange** (kontroluje, jestli pixel patří do definovaného rozsahu) můžeme vytvořit masku pohybujícího se objektu. Pixely odpovídající definovanému rozsahu budou mít bílou barvu, zbytek zůstane černý. Masku je potřeba upravit (odstranit šum, zaplnit malé díry) s použitím morfologických operací. 2.4



Obrázek 40: Příklad nastavení parametrů kamery



vstupní snímek

příklad nastavení rozsahu pro zelenou barvu

výsledná maska pro zelenou barvu

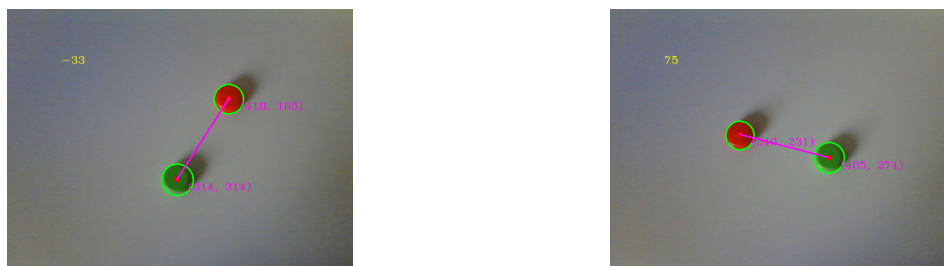
Obrázek 41: Příklad barevné segmentace v HSV prostoru

3. Získávání souřadnic a úhlu

Dále pracujeme jen s vytvořenou maskou. Předpokládáme, že maska má podobu vyplněného kruhu, můžeme vyhledat kontury (**findContours**) a vypočítat obrazové momenty (**moments**). 2.5 Následným výpočtem získáváme x,y souřadnice každého míčku v obrazovém prostoru (480x640px). [31]

Výstup potřebný pro zvukovou část je střed úsečky, ohraničené míčky. Střed úsečky simuluje střed hlavy člověka neboli polohu uší.

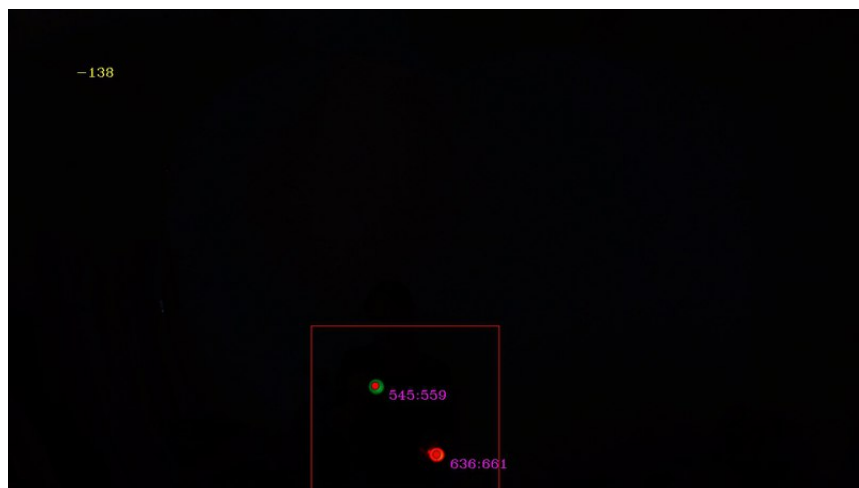
Zbývá vypočítat úhel natočení hlavy. Jeden míček (zelený) bude referenčním, přes tento bod provedeme vertikálu. Hledaný úhel je úhel mezi vertikálou a druhým (červeným) míčkem, přičemž na jednu stranu hodnota úhlu je kladná 0° až 180° , na druhou záporná 0° až -180° . Čím dále od sebe se budou míčky nacházet, tím výraznější bude úhel. Avšak nesmějí být tak daleko, aby rušily návštěvníka.



Obrázek 42: Výpočet úhlu na příkladu víček

Optimalizace a opatření Hledání barev v celém snímku zbytečně zatěžuje program. Místo toho lze definovat oblast zájmu (**Region Of Interest**) - čtverec určité velikosti kolem míčků. Pokud se nepodaří nalézt míčky v tomto okolí, hledání pokračuje v celém snímku.

Program si musí poradit i s nouzovým případem, kdy jeden z míčků (a nebo i oba) zmizí, například, návštěvník zvedne ruku. To může způsobit problém s výpočtem zvuku (zvuk "uletí"). V takovém případě lze použít Kalmanův filtr 2.6, který dokáže predikovat budoucí souřadnice míčků vzhledem k předchozím hodnotám. To znamená, že pokud se podařilo najít střed míčku, úhel a střed budou počítány pomocí skutečných souřadnic. Ale když dojde k chvilkové ztratě, výpočet bude proveden podle predikovaných hodnot.



Obrázek 43: Ukázka výstupu programu

Okomentovaný kód je na CD.

Hodnocení Program funguje dle očekávání. Manuální nastavování parametrů kamery a rozsahů barev umožňuje kvalitní segmentaci objektů zájmu.

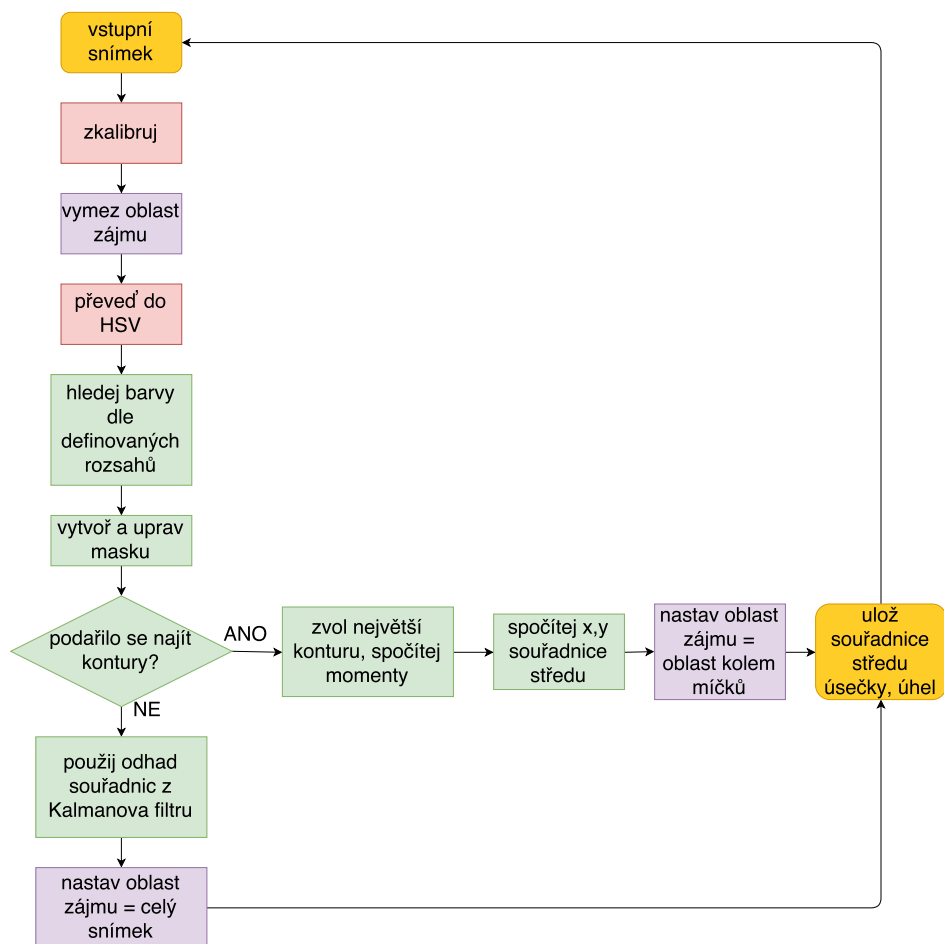
Níže uvedené hodnoty platí pro Intel(R) Core(TM) i5-2450M CPU 2.50GHz, 4GB, x64. Průměrný čas, potřebný k zpracování jednoho snímku je 0,1ms (snímek s použitím ROI, kdy nedošlo ke ztratě míčků) a 0,4ms (pro nouzový případ, kdy došlo ke ztratě, vyhledávání se provádí v celém snímku, k výpočtům slouží predikovaná hodnota). Využití CPU je od 30 do 40%, jak i v případě času parametr je závislý na tom, o jakou situaci se jedná. Využití paměti je kolem 40MB.

5.3 Zvuková část

Vybavení

- Bezdrátové mikrofony Sennheiser EW100 G2 2 kusy
- Prostorová sluchátka Zalman MCH MML 1000 HF Surround Sound Stereo Headphones
- Zvuková karta Steinberg UR22

Zvuková část má dvě fáze: nahrávání reálného zvuku a výpočet prostorového zvuku, neboli tvorba imaginární postavy.



Obrázek 44: Zjednodušené schéma obrazové části programu. Příkazy v zelených blocích mají být aplikovány na oba míčky

K účelu nahrávání slouží 2 bezdrátové všesměrové mikrofony. Jeden je připevněn k noze návštěvníka a zaznamenává kroky, druhý je umístěn v helmě v úrovni krku a je potřebný pro nahrávání projevů emocí.

Výpočet zvukového výstupu zajišťuje knihovna irrKlang [1], která se stará o záznam a přehrávání zvuku, a jeho korektní prostorové umístění. Zvuk je simulován s použitím výstupu obrazové části - souřadnic v 2D prostoru.

Přestavme si, že teď jsme v okamžiku, kdy uplynula první minuta v místnosti. Stojíme v její pravém rohu. V ten samý okamžik (jedna minuta v místnosti) předchozí návštěvník přešlapoval v levém rohu. Ve sluchátkách uslyšíme zvuk kroků, který přichází z tohoto rohu. Když budeme točit hlavou, tak i zvuk se bude měnit.

Pozici posluchače zadáváme přímo, zatímco pozici předchozího návštěvníka dopočítáváme pro aktuální časový okamžik lineární interpolací dvou záznamů s nejmenší časovou vzdáleností od aktuálního uběhnutého času od začátku průchodu.

Výsledný zvuk je posílán do externí zvukové karty, ze které jde do sluchátek na hlavě návštěvníka.

6 Závěr

V první části práce byl proveden teoretický rozbor nezbytných matematických pojmů a přístupů z oblasti zpracování obrazu. Tyto přístupy byly implementovány ve vhodných programovacích prostředích a sloužily pro tvorbu obrázků.

Druhá část práce byla věnována algoritmům segmentace popředí z videosekvencí pořízených nepohyblivou kamerou. Zvolila jsem si několik různých přístupů k problému, každý jsem podrobně popsala. Vybrané algoritmy byly implementovány. Následně bylo provedeno porovnání algoritmů, diskutovány jejich silné a slabé stránky, možnosti použití a vylepšení. Výsledky lze najít v sekci 4. Na tomto místě bych chtěla uvést jen stručnou tabulku.⁴

		<i>Model pozadí</i>	<i>Parametry</i>	<i>Rychlost</i>	<i>Paměť</i>
<i>Základní</i>	<i>Static</i>	První snímek videosekvence	-	***	***
	<i>FD</i>	Předchozí snímek	-		
	<i>Mean</i>	Průměr z N předchozích snímků	N		
	<i>Adaptive</i>	Aktualizované pozadí	α		
<i>Gaussovské</i>	<i>1G</i>	Gaussovo rozdělení pro každý pixel	N	**	***
	<i>MOG</i>	Směs gaussových rozdělení pro každý pixel	K	*	**
<i>Podprostory</i>	<i>PCA</i>	Vlastní podprostor	M	**	*
<i>Shlukové</i>	<i>CB</i>	Kódová kniha pro každý pixel	L	-	**

Tabulka 7: Souhrnný popis algoritmů, kde N , K , M , N jsou počty snímků, Gaussových rozdělení, vlastních vektorů a kódových slov, α je učící konstanta - parametry mající vliv na zatížení algoritmů.

Poslední kapitola se týkala tvorby interaktivní multimediální instalace. Nejdřív se čtenář mohl seznámit s principem fungování a uměleckým významem instalace, následoval podrobný rozbor obrazové části (realizace systému sledování návštěvníků) a krátké seznámení s částí zvukovou. V současné době instalace je ve výstavbě. Plánujeme ji otevřít veřejnosti na konci června 2016.

⁴U metody CB není uvedeno hodnocení rychlosti, protože rychlost výpočtu je hodně závislá na složitosti vstupního videa.

7 Obsah příloženého CD

Ve složce "*Segmentace popředí > Algoritmy*" je vstupní video a zdrojové kódy, jimž je věnována druhá část práce. Nacházejí se v příslušných složkách s názvy podle zkratk, zavedených ve výkladu (například, "*Segmentace popředí > Algoritmy > Static*"). Každá složka obsahuje soubor s implementací a příklad výstupu.

Ve složce "*Segmentace popředí > Porovnání*" jsou materiály, jejichž pomocí bylo provedeno hodnocení algoritmů: obrázky po segmentaci, obrázky GT a příslušný program pro výpočet parametrů.

Implementaci systému pro sledování návštěvníků lze nalézt ve složce "*Instalace > Tracking*", složka "*Instalace*" obsahuje také pomocný program sloužící k nastavení rozsahů barev v HSV prostoru.

Reference

- [1] Ambiera, irrKlang. [vid. 21.5.2016] Dostupné z: <http://www.ambiera.com/irrklang/>.
- [2] Matlab documentation [online]. [vid. 21.5.2016] Dostupné z: <http://www.mathworks.com/help/matlab/>.
- [3] OpenCV documentation. Camera Calibration and 3D Reconstruction. fisheye [online]. [vid. 21.5.2016] Dostupné z: http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#fisheye.
- [4] OpenCV documentation. Motion Analysis and Object Tracking. Background-SubtractorMOG2 [online]. [vid. 21.5.2016] Dostupné z: http://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html#backgroundsubtractormog2.
- [5] OpenCV documentation [online]. [vid. 21.5.2016] Dostupné z: <http://opencv.org/documentation.html>.
- [6] OpenCV documentation. Structural Analysis and Shape Descriptors. moments [online]. [vid. 21.5.2016] Dostupné z: http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=moments#moments.
- [7] Perceptual computing. Statistical modeling of complex background for foreground object detection. [vid. 21.5.2016] Dostupné z: http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html.
- [8] SparkoCam. [vid. 21.5.2016] Dostupné z: <https://sparkosoft.com/sparkocam>.
- [9] Test images for Wallflower paper. [vid. 21.5.2016] Dostupné z: <http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm>.
- [10] Wikipedia. Principal component analysis [online]. [vid. 21.5.2016] Dostupné z: https://en.wikipedia.org/wiki/Principal_component_analysis.
- [11] Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, H elene Laurent, and Christophe Rosenberger. Comparative study of background subtraction algorithms [online]. *Journal of Electronic Imaging*, 2010.
- [12] Thierry Bouwmans. Recent advanced statistical background modeling for foreground detection-a systematic survey [online]. *Recent Patents on Computer Science*, 2011.
- [13] Thierry Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview [online]. *Computer Science Review*, 2014.
- [14] Sen-Ching S Cheung and Chandrika Kamath. Robust background subtraction with foreground validation for urban traffic video [online]. *EURASIP Journal on Advances in Signal Processing*, 2005.
- [15] Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach [online]. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, 1997.
- [16] Rafael C Gonzalez and Richard E Woods. Digital image processing, vol. 2, 2002.

- [17] Charles Guyon, El-hadi Zahzah, and Thierry Bouwmans. Robust principal component analysis for background subtraction: systematic evaluation and comparative analysis [online]. 2012.
- [18] Janne Heikkilä and Olli Silvén. A real-time system for monitoring of cyclists and pedestrians [online]. *Image and Vision Computing*, 2004.
- [19] Václav Hlaváč. Matematická morfologie [online]. [vid. 21.5.2016] Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/71-3MatMorpholBinCz.pdf>.
- [20] Karel Horák. Matematická morfologie [online]. [vid. 21.5.2016] Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/10_Matematicka_morfologie.pdf.
- [21] Michal Jůza. Detection of the unusual behavior from video sequences [online]. Czech Technical University in Prague 2004, Diploma thesis, Faculty of Electrical Engineering. Dostupné z: <http://euler.fd.cvut.cz/~juzam/DiplJuzaA.pdf>.
- [22] Hiremath S. Joshi D.G. Chadda V.K. Bajpai A. Kar, S. A multi-algorithmic face recognition system [online]. [vid. 21.5.2016] Dostupné z: <http://home.iitk.ac.in/~draj/cs676/project/paper.html>.
- [23] Johannes Kilian. Simple image analysis by moments [online]. [vid. 21.5.2016] Dostupné z: <http://breckon.eu/toby/teaching/dip/opencv/SimpleImageAnalysisbyMoments.pdf>.
- [24] Kyungnain Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction [online]. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*. IEEE, 2004.
- [25] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Real-time foreground-background segmentation using codebook model [online]. *Real-time imaging*, 2005.
- [26] Kyongil Yoon Kyungnam Kim, Thanarat Horprasert. Background Subtraction Program [online]. [vid. 21.5.2016] Dostupné z: <http://www.umiacs.umd.edu/~knkim/UMD-BGS/index.html>.
- [27] Satya Mallick. Why does OpenCV use BGR color format? [online]. [vid. 21.5.2016] Dostupné z: <http://www.learnopencv.com/why-does-opencv-use-bgr-color-format/>.
- [28] Mirko Navara. *Pravděpodobnost a matematická statistika*. České vysoké učení technické v Praze, Nakladatelství ČVUT, Praha, 2007.
- [29] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions [online]. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2000.
- [30] Massimo Piccardi. Background subtraction techniques: a review [online]. In *Systems, man and cybernetics, 2004 IEEE international conference on*. IEEE, 2004.
- [31] Adrian Rosebrock. Ball tracking with OpenCV [online]. [vid. 21.5.2016] Dostupné z: <http://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/>.
- [32] Lindsay I Smith. A tutorial on principal components analysis [online]. [vid. 21.5.2016] Dostupné z: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

- [33] Andrews Sobral. BGSLibrary: An OpenCV C++ Background Subtraction Library. In *IX Workshop de Visão Computacional (WVC'2013)*, 2013.
- [34] Andrews Sobral and Antoine Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos [online]. *Computer Vision and Image Understanding*, 2014.
- [35] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces [online]. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE, 1991.
- [36] Ondřej Vala. Využití digitálního fotoaparátu v kartografické reprodukci [online]. ČVUT 2011. Diplomová práce, ČVUT, Fakulta stavební, Obor geoinformatika. Dostupné z: <http://geo.fsv.cvut.cz/proj/dp/2011/ondrej-vala-dp-2011.pdf>.
- [37] Greg Welch and Gary Bishop. An introduction to the Kalman filter. Department of Computer Science, University of North Carolina, 2006.
- [38] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body [online]. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1997.