



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Detekce a sledování pohybu osob na základ záznamu z kamerového systému
Student:	Petr Wudi
Vedoucí:	doc. RNDr. Ing. Marcel Ji ina, Ph.D.
Studijní program:	Informatika
Studijní obor:	Teoretická informatika
Katedra:	Katedra teoretické informatiky
Platnost zadání:	Do konce zimního semestru 2017/18

Pokyny pro vypracování

- 1) Seznamte se s úlohou detekce a sledování osob v kamerovém záznamu z více kamer, které jsou umíst ny nad prostorem, kde se osoby pohybují.
- 2) Prove te rešerši metod, které se zabývají touto úlohou, a dále nastudujte metody zpracování obrazu, které jsou využitelné pro danou úlohu.
- 3) Navrhn te postup a díl í metody pro detekci a sledování osob v kamerových záznamech. Zohledn te specifika snímacího procesu.
- 4) Navržené metody implementujte v programovacím jazyku Java s využitím voln dostupných knihoven.
- 5) Implementované metody ov te na nam ených reálných datech, vyhodno te dosaženou p esnost a navrhn te možná zlepšení. Data, se kterými se bude pracovat, jsou již k dispozici v podob kamerových záznam a pochází z již reáln nam ené prodejny.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 24. února 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA TEORETICKÉ INFORMATIKY



Bakalářská práce

Detekce a sledování pohybu osob na základě záznamu z kamerového systému

Petr Wudi

Vedoucí práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

16. května 2017

Poděkování

Rád bych touto formou poděkoval doc. RNDr. Ing. Marcelu Jiřinovi, Ph.D. za čas strávený vedením mé práce a za cenné rady. Také bych chtěl poděkovat své rodině za trpělivost a podporu během psaní práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Petr Wudi. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Wudi, Petr. *Detekce a sledování pohybu osob na základě záznamu z kamerového systému*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato bakalářská práce popisuje algoritmus, který dokáže v kamerovém záznamu detekovat jednotlivé osoby a ty dále sledovat. Omezuje se přitom na případ kamery umístěné nad hlavami těchto osob. Cílem práce je porovnat různé postupy, které se k řešení podobných problémů používají a navrhnout co možná nejpřesnější algoritmus, který by byl schopen běhu v reálném čase.

Klíčová slova detekce osob, sledování osob, trasování osob, počet osob, kamerový záznam, hloubková mapa, pohled shora, Kinect

Abstract

This thesis describes algorithm able to detect people in camera record and to track them. Problem is reduced to case of camera placed above people's heads. Object of the thesis is to compare various approaches and propose accurate real-time algorithm.

Keywords people detection, people tracking, human tracking, human counting, camera record, depth map, top-view, Kinect

Obsah

Úvod	1
1 Cíl práce	3
1.1 Zařízení Kinect	3
2 Základní pojmy	7
2.1 Hlubková mapa	7
2.2 Detekce osob	7
2.3 Sledování osob	7
2.4 Lokální a globální maximum	8
2.5 Morfologický operátor	8
2.6 Segmentace obrazu	8
3 Podobné existující algoritmy	9
3.1 Detekce osob s využitím hlubkové mapy	9
3.2 Detekce osob s využitím barevného záznamu	16
3.3 Hybridní řešení	18
3.4 Zobecnění	20
4 Vybrané metody	21
4.1 Narůstání oblastí	21
4.2 Water filling	22
4.3 Potlačení nemaximálních hodnot	23
4.4 H-maxima	23
4.5 Problém přiřazení	24
4.6 Maďarský algoritmus	25
4.7 Kalmanův filtr	26
4.8 Převod hlubkové mapy na 3D model	29
4.9 Watershed	30

5	Návrh a implementace	31
5.1	Obecný návrh	31
5.2	Preprocessing	33
5.3	Detekce objektů	36
5.4	Úprava výsledku detekce	41
5.5	Segmentace	43
5.6	Příznaky	44
5.7	Klasifikace objektů	46
5.8	Sledování	47
5.9	Shrnutí	55
5.10	Implementace	56
6	Testování	57
6.1	Ukázka výsledků	57
6.2	Metrika	59
6.3	Detekce potenciálních osob	59
6.4	Preprocessing	60
6.5	Úprava výsledků detekce	61
6.6	Klasifikace	62
6.7	Sledování	69
	Závěr	75
	Literatura	77
	A Použité značení	83
	B Obsah příloženého CD	85
	C ROC křivky detekčních metod	87

Seznam obrázků

1.1	Zařízení instalovaná v obchodním domě	4
1.2	Umístění Kinectu	4
1.3	Příklady snímků	5
1.4	Popis zařízení Kinect	5
1.5	Výpočet hloubkové mapy	6
3.1	Oblasti použité ve výpočtu příznaku box text	10
3.2	Paraboly probíhající třemi vybranými body	12
3.3	Osoba označená Haarovým příznakem	14
4.1	Znázornění metody Water Filling na jednorozměrném snímku . . .	23
4.2	Ukázka funkčnosti operátoru h-maxima na jednorozměrném snímku	23
4.3	Princip fungování Kalmanova filtru	27
4.4	Disproporce obrazu osoby v závislosti na jejím umístění na snímku	29
4.5	Převod snímku na 3D model	29
5.1	Celkový návrh algoritmu detekce a sledování osob	32
5.2	Vznik děr	34
5.3	Díry v hloubkové mapě	34
5.4	Různé techniky rozostření	35
5.5	Ukázka průběhu různých implementací hill climbingu	37
5.6	Narůstání oblastí segmentující dvě splývající osoby	39
5.7	Úprava výsledků detekce	42
6.2	Ukázka ohodnocení vybraných výsledků sledování	73

Úvod

Jednou z klíčových vlastností pro úspěch obchodního řetězce je znalost chování jeho zákazníků. Pouze obchod, který ví, o jaké produkty mají zákazníci zájem a za jakou cenu jsou ochotni je kupovat, může na trhu obstát.

K efektivní analýze chování zákazníků či lidí obecně lze použít různé počítačové systémy, existuje velké množství programů určených k tomuto účelu a celý vědní obor – data mining – který se zaměřuje na získávání užitečných informací například z databáze prodaného zboží.

Pro získávání různých znalostí z dat však vždy potřebujeme mít vstupní data – nejlépe ve formátu různých čísel, případně textu.

Různé informace mohou být skryty v datech odlišného formátu: například efektivita rozdělení zboží do regálů může být analyzována z kamerového záznamu.

Abychom mohli použít obrazová data, musí však existovat program, který by na nich byl schopen označit, kolik je na snímku lidí, případně i kde a jakým způsobem se pohybují. Tento program samozřejmě může mít i mnohé další využití.

Proto ve své práci popisuji algoritmus, který bude zpracovávat data z již existujícího systému, který seshora zaznamenává snímek uličky v supermarketu. Snímky jsou dostupné jak v klasickém barevném formátu, kterým zpracovává obraz lidské oko, tak ve formátu hloubkových map.

Hloubkové mapy jsou druhem záznamu dat, kdy je v obrázku pomocí různých stupňů šedi zaznamenávána vzdálenost nějakého objektu od kamery.

Svůj algoritmus budu porovnávat s algoritmem, který již úspěšně osoby detekuje a sleduje.

Současný algoritmus však někdy jako člověka označí i něco jiného, například nákupní vozík. Mívá problémy i při přílišné blízkosti dvou osob.

Mým úkolem je tedy prozkoumat problematiku detekce a sledování osob v hloubkových mapách i v klasických fotografiích a vytvořit systém, který bude detekovat osoby s větší úspěšností.

Cíl práce

Cílem mé bakalářské práce je vytvořit algoritmus, který bude schopen detekovat osoby na kamerovém záznamu a tyto osoby sledovat.

Vstupem pro algoritmus je záznam ze zařízení Microsoft Kinect. Toto zařízení je instalováno nad uličkou v obchodním domě a slouží k analýze chování zákazníků.

Ulička je snímána shora, díky tomu se v záznamu jednotlivé osoby příliš nepřekrývají, jako by tomu bylo při zobrazení z boku.

Kinect je instalován na jednom z regálů vedle uličky (viz obr. 1.1). Kdyby směřoval zcela kolmo k zemi, polovina záběru by zobrazovala pouze regál a nikoli uličku. Proto je mírně natočen směrem do uličky (viz obr. 1.2). Díky tomu jsou osoby snímány mírně z boku a stává se, že na snímku jedna osoba částečně zakryje druhou (viz obr. 1.3d).

Kvůli tomuto řešení navíc není člověk na protější straně uličky vidět celý (obr. 1.3c).

V každé uličce se nachází několik Kinectů, jejich zorná pole se navzájem překrývají. Proto je možné sledovat osobu pohybující se celou uličkou.

Osoby na snímcích se mohou nacházet v různých polohách, nejčastější jsou osoby jdoucí, stojící nebo shýbající se k zemi.

Rozmanité je i jejich oblékání – v záznamu se nacházejí lidé s čepicí, dlouhými vlasy, s batohem, atd. I to výrazně ovlivňuje vzhled člověka na snímcích.

Sledování také ztěžuje fakt, že se v záznamu objevují objekty, jako jsou kočárky, nákupní vozíky či další (obr. 1.3c a 1.3e). Ty by sledovány být neměly.

1.1 Zařízení Kinect

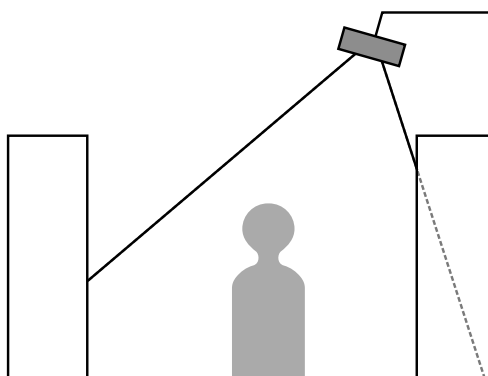
Tato kapitola popisuje zařízení Microsoft Kinect, které bylo použito pro snímání osob.

Kinect pořizuje duální záznam sestávající z klasických barevných snímků ve formátu RGB a hloubkové mapy (udává vzdálenost objektů od kamery,

1. CÍL PRÁCE



Obrázek 1.1: Zařízení instalovaná v obchodním domě



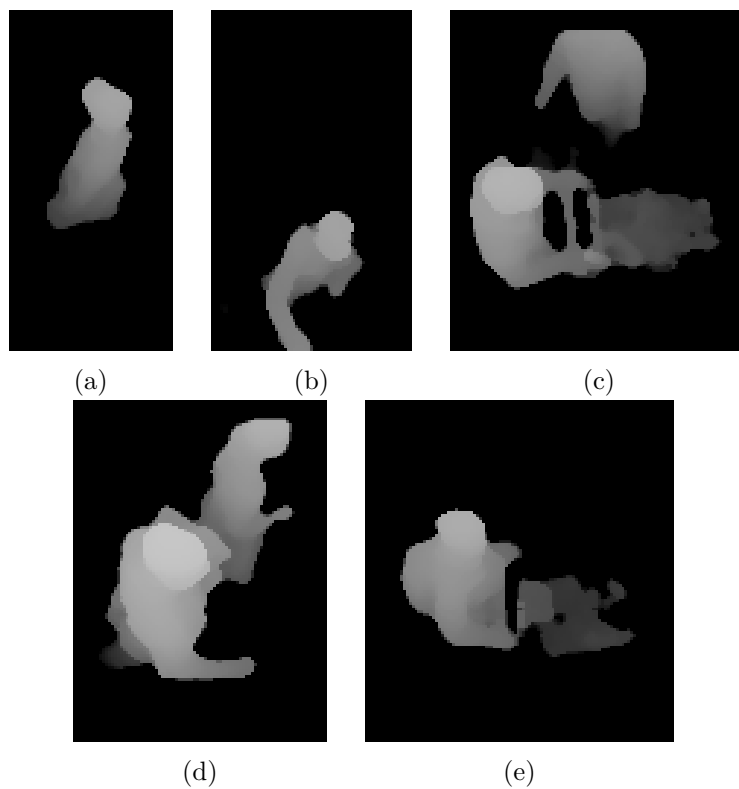
Obrázek 1.2: Umístění Kinectu

více v sekci 2.1). Tyto snímky jsou přibližně synchronizovány a je proto možné jednotlivé RGB snímky a hloubkové mapy spárovat. Přesné spárování pixel na pixel není možné, protože senzor snímající hloubkovou mapu je umístěn 2,5 centimetrů od barevné kamery [2] a snímá tedy trochu jinou oblast. Barevná kamera má navíc menší ohniskovou vzdálenost, takže zabírá větší úhel [2].

Hloubkovou mapu snímá Kinect pomocí infračerveného (IR) senzoru. Zařízení obsahuje infračervený projektor, který promítá do prostoru velké množství bodů. Každý z těchto bodů má trochu jiný tvar, takže se dají navzájem rozlišit. Všechny body jsou promítány pod jiným úhlem a tím pádem dopadají na jiná místa.

Dále Kinect obsahuje IR senzor, který vytváří snímky těchto bodů. Senzor a projektor jsou od sebe vzdáleny 7,5 centimetrů [2].

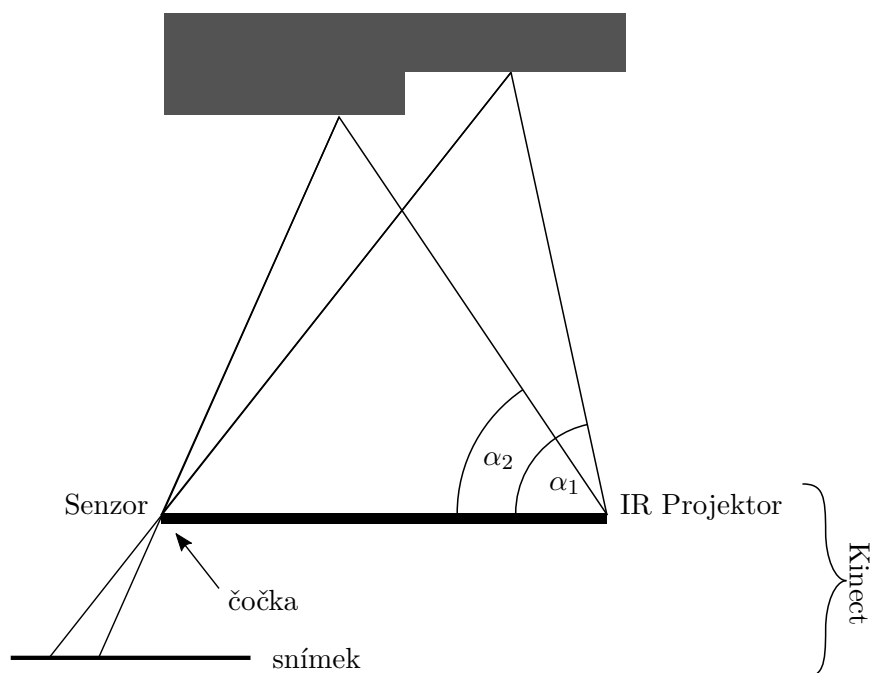
Z těchto bodů je možné vypočítat hloubkovou mapu. U každého zachyceného bodu podle tvaru můžeme zjistit úhel, pod kterým byl promítnut. Z úhlu,



Obrázek 1.3: Příklady snímků



Obrázek 1.4: Popis zařízení Kinect, upraveno z [1]



Obrázek 1.5: Výpočet hloubkové mapy

polohy na infračerveném snímku a ze vzdálenosti mezi senzorem a projekto-rem se počítá poloha a hloubka bodu, kam infračervený paprsek dopadl (viz obr. 1.5).

Optická délka Kinectu odpovídá 580 pixelům pro snímek o velikosti 640×480 pixelů.

Základní pojmy

Tato kapitola představuje vybrané pojmy, se kterými v textu své práce pracuji.

2.1 Hloubková mapa

Hloubková mapa je obrázek obsahující informaci o vzdálenosti snímaného tělesa od snímacího zařízení.

Kódování hloubkové informace může být různé. V některých případech jsou pro různé vzdálenosti použity různé barevné odstíny [3], avšak zdaleka nejčastější je kódování pomocí 256 stupňů šedi, kdy jednomu pixelu odpovídá jeden byte.

I v případě kódování pomocí stupňů šedi existují různé konvence. Někdy vyšší hodnota v hloubkové mapě (více bílé) indikuje výše položené místo [4, 5], jindy místo dále od snímacího zařízení [6, 7, 8, 9]. V této práci je mapa vždy chápána prvním způsobem, tedy výše položené místo má vyšší hodnotu.

Pro popis hloubkových map v této práci používám následující značení: x – vzdálenost (v pixelech) od levého okraje snímku, y – vzdálenost (v pixelech) od horního okraje snímku, δ – hloubková mapa, případně hloubka jednoho bodu (vždy patrné z kontextu), δ_{xy} – hodnota hloubkové mapy v bodě $[x, y]$.

2.2 Detekce osob

Detekcí osob rozumíme postup, během kterého algoritmus zjistí, kde všude se na zadaném snímku nacházejí osoby.

Algoritmus, který provádí detekci, nazýváme detektor.

2.3 Sledování osob

Sledování (nebo také „trasování“) osoby je definováno jako odhadování polohy jednoho či více lidí pomocí kamerového záznamu [10].

2.4 Lokální a globální maximum

Globální maximum je nejvyšší hodnotou v prostoru. V mé práci se tímto prostorem rozumí hloubková mapa.

Lokální maximum je nejvyšší místo ve svém okolí. Toto okolí tvoří všechny pixely do stanovené vzdálenosti (Eukleidovské, Manhattanské či jiné).

Podobně jako globální a lokální maximum je definováno i globální a lokální minimum.

2.5 Morfologický operátor

Algoritmy nazývané jako morfologické operátory slouží k práci s obrázky. Jsou vhodné pro analyzování tvarů objektů [11].

S obrázkem pracují jako s bodovou množinou (dále v textu značena I). Tato diskrétní množina je analyzována pomocí menší bodové množiny B nazývané strukturní element.

Strukturní element může mít libovolný tvar. Každý strukturní element však musí mít počátek v pevně stanoveném bodě [11]. Strukturní element je pak posouván obrázkem a postupně vztažen ke všem jeho bodům (počátek je umístěn do každého bodu).

Morfologické operátory mohou zpracovávat binární, šedotónové či barevné obrázky [11]. Ve své práci se omezují pouze na binární morfologické operátory.

Mezi základní morfologické operátory patří dilatace a eroze [12]. Z nich je pak možné vytvořit mnoho dalších operátorů [12].

Výsledkem dilatace (dilation) je zvětšení objektu na obrázku a potlačení malých děr [11].

Dilatace je provedena jako Minkovského součet¹ $I \oplus B$ [11]. Hodnota každého bodu je tedy spočtena jako sjednocení strukturního elementu (vztaženého k tomuto bodu) a původního obrázku.

Eroze (erosion) naopak objekt zmenšuje. Je definována jako Minkovského rozdíl² $I \ominus B$. Každý bod tedy má kladnou hodnotu, pokud všechny body v okolí mají také kladnou hodnotu. Okolí bodu je definováno strukturním elementem.

2.6 Segmentace obrazu

Segmentace je rozdělení celého snímku na několik souvislých disjunktních oblastí. Sjednocením všech oblastí vznikne původní snímek [13].

¹ $I \oplus B = \bigcup_{b \in B} X_b$
² $I \ominus B = \bigcap_{b \in B} X_b$

Podobné existující algoritmy

Algoritmus popsaný v mé práci má využívat data zachycená zařízením Microsoft Kinect. Toto zařízení je schopno zaznamenávat obrazová data ve formátu RGB a zároveň hloubkové mapy.

Algoritmus tedy může využívat barevné snímky, hloubkové mapy, případně obě informace zároveň, protože každému barevnému snímku odpovídá jedna hloubková mapa.

3.1 Detekce osob s využitím hloubkové mapy

V této kapitole uvádím zdroje, které pro detekci osob, případně i jejich sledování, používají pouze hloubková data.

[6] využívá pro detekci osob algoritmus narůstání oblastí (angl. region growing), který je dále popsán v sekci 5.3.3. Pomocí tohoto algoritmu obrazy segmentuje. Cílem je oddělit oblasti, které zobrazují hlavy osob.

Vstupní hloubková mapa může obsahovat chybějící místa („díry“) a statické objekty (nábytek). Proto jsou před samotnou detekcí díry nalezeny a vyplněny podle svého okolí.

Před spuštěním algoritmu proběhla „kalibrační fáze“, kdy v záběru nebyli žádní lidé. Ze všech snímků během kalibrační fáze je mediánem vytvořena hloubková mapa pozadí. Pixely, které mají hodnotu příliš blízkou pozadí, jsou potlačeny. Tím jsou odstraněny statické objekty.

Oblast může v [6] mít rozsah hodnot maximálně 100 odstínů šedi. Obsahuje-li oblast pixel s hodnotou δ , pak pixely s nižší hodnotou, než je $\delta - 100$ již do oblasti patřit nesmějí. Díky tomu původní oblast obsahuje pouze hlavu. Zbytek těla je pak součástí dalších oblastí.

Oblasti, které obsahují hlavu jsou pak od ostatních (zbytek těla, nákupní košík, ...) rozlišeny pomocí tří příznaků:

- normalizovaný obsah – kolik pixelů oblast zabírá. Na snímcích může dojít k tzv. „efektu rybího oka“, vysoké objekty na snímku zabírají větší

3. PODOBNÉ EXISTUJÍCÍ ALGORITMY

plochu, než stejně velké nižší objekty. Proto je třeba normalizace. Celá hloubková mapa je promítnuta na podlahu (převedená na 3D model, viz sekce 4.8). Tím se obsah stejně velkých objektů sjednotí.

- kulatost – kolik procent pixelů z dané oblasti se nachází nejdále $\sqrt{S/\pi}$ pixelů od jejího těžiště. Proměnná S značí obsah.
- box test – zda je objekt lokálním maximem. Výpočet příznaku probíhá následovně: Okolo každé oblasti R je vytvořena oblast $P_l(R)$. Do oblasti $P_l(R)$ patří všechny pixely které se nacházejí nejdále l pixelů od okraje R , ale nepatří do R (viz obr. 3.1).

Označíme-li maximum celé oblasti $P_l(R)$ jako δ_{max} , pak hodnotou příznaku box test je:

$$\frac{|\{p \in R | \delta(p) > \delta_{max} + c_{height}\}|}{|R|}$$

Proměnná c_{height} a dále používané c značí předem stanovené konstanty. Z každého z těchto příznaků je spočtena „correctness“, tedy číslo náležející do intervalu $(0, 1)$ definované jako

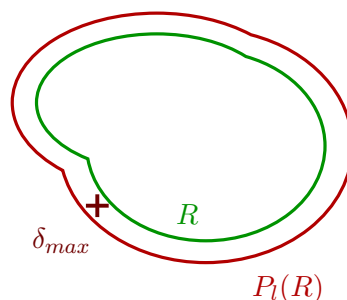
$$\frac{1}{1 + e^{-c(d - c_{height})}}$$

Vynásobením všech „correctness“ navzájem vznikne pravděpodobnost toho, že oblast zobrazuje lidskou hlavu.

Poslední částí algoritmu popsaného v [6] je sledování. To probíhá pomocí maďarského algoritmu (podrobnější popis v sekci 4.6) a Kalmanova filtru (v sekci 4.7). Mezi jednotlivými osobami na předchozím snímku a na současném snímku je spočtena „vzdálenost“ pomocí eukleidovské vzdálenosti a jejich podobnosti. K výpočtu podobnosti jsou použity výše jmenované příznaky. Maďarský algoritmus se snaží přiřadit k sobě osoby tak, aby minimalizoval celkovou vzdálenost.

Může se stát, že osoba nebyla na jednom snímku detekována. Proto [6] do množiny detekovaných potenciálních osob přidává „přízraky“, tedy osoby z předchozího snímku. Jejich poloha je předpovězena pomocí Kalmanova filtru. Vzdálenost přízraku a osoby na předchozím snímku je vynásobena konstantou, takže algoritmus preferuje použití skutečně detekované osoby.

Algoritmus popsaný v [7] nejdříve opraví hloubkovou mapu, ve které chybějí některá místa. Tato chybějící místa (díry) mohou mít dvě příčiny: buď přílišná blízkost použitého zařízení Kinect k hlavám osob (objekty blíže než



Obrázek 3.1: Oblasti použité ve výpočtu příznaku box test

50 cm není Kinect schopen zaznamenat), případně přílišná členitost pozadí. Proto jsou velké díry identifikovány jako hlava a celé oblasti je přiřazena vysoká hodnota, menší díry pak jsou vyplňovány podle svého okolí.

Následně je potlačeno pozadí (pixely, jejichž hodnota je po dobu měření přibližně stejná). Poté najde tzv. „head candidates“, tedy místa, kde by se mohla vyskytovat hlava člověka. K tomu je použita stejně jako v [6] metoda narůstání oblastí.

Všechny „head candidates“ jsou poté ohodnoceny stejnými třemi příznaky, jako používá [6] – tedy normalizovaný obsah, kulatost a box test. Podle nich je poté stanoveno, zda se jedná skutečně o člověka nebo nikoliv. Pro případ, že by došlo v rámci jednoho snímku k chybě, je nakonec použit ještě maďarský algoritmus, který používá data z předchozích snímků.

Rauter [14] detekuje potenciální osoby na místech lokálních maxim pomocí metody potlačení nemaximálních hodnot (viz sekce 4.3). Pro další zpracování je vhodné znát střed hlavy. Ten je nalezen pomocí Hill climbingu, který je podobný algoritmu Mean-shift popsanému v [15].

Potenciální osoby jsou dále klasifikovány pomocí metody SVM (support vector machines), která je označí jako skutečné lidi nebo jako jiné objekty.

Snímek je rozdělen na oblasti o stanovené velikosti. V každé oblasti je spočten součet pixelů a poté histogram rozdílů těchto součtů mezi sousedními oblastmi. Pomocí histogramu provádí SVM klasifikaci.

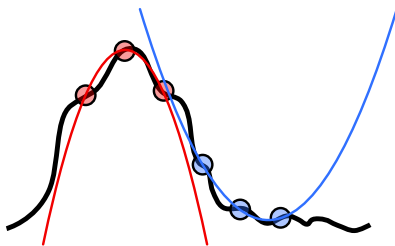
Poslední fází algoritmu je sledování osob, které probíhá za pomoci matice čtverců vzdáleností mezi osobami detekovanými na předchozích snímcích a osobami na současném snímku. Matice slouží jako vstup jednoduchého přiřazovacího algoritmu. Ten se snaží přiřadit každé osobě z předchozích snímků nejbližší osobu na novém snímku. Dojde-li ke kolizi, jsou nejdříve řešeny osoby, které mají nejméně rozumných kandidátů na přiřazení.

Stahlschmidt et al. [4] během preprocessingu odstraní pixely s hodnotou pod stanovenou hranicí, poté rozdělí celý snímek do několika regionů oddělených prázdnými pixely. Každý z regionů je normalizován: hodnoty pixelů jsou přepočítány, aby nejvyšší a nejnižší hodnota každého regionu byla stejná. Krom toho je součástí preprocessingu Gaussovo rozostření.

V takto upraveném snímku je poté možno pomocí algoritmu extended h-maxima transform najít lokální maxima, která reprezentují hlavy osob. Ty jsou pak trasovány pomocí Kalmanova filtru.

Algoritmus zvaný Water Filling je představen v [8]. Vstupem pro Water Filling je hloubková mapa, výstupem pak speciální zobrazovací funkce g , která pomáhá najít lokální maxima. Výhodami g oproti původní hloubkové mapě jsou odolnost vůči šumu a nezávislost na velikosti či výšce objektu. Podrobněji je algoritmus popsán v sekci 4.2

Coskun et al. [16] k Water Filling přidává tracker založený na kombinaci Kalmanova filtru a maďarského algoritmu.



Obrázek 3.2: Paraboly probíhající třemi vybranými body. V případě vlevo se jedná o osobu (a prostřední bod je lokální maximum), v druhém případě ne. Levá parabola je konkávní, pravá konvexní.

Podle [9] jsou detekce a sledování zpravidla příliš náročné na výpočet. Proto představuje jednodušší metodu, navíc použitelnou pro zpracování hloubkových map i barevných snímků.

Problémem je zde však nedostatečná obecnost: metoda vyžaduje, aby se všechny osoby pohybovaly předem stanoveným směrem, případně směrem opačným (např. v uličce).

Podle této metody nejdříve proběhne detekce popředí (objekty o dostatečné výšce, které v průběhu času nezůstávají neměnné). Tímto způsobem jsou detekovány „bloby“, které posléze podstoupí sledování.

Pro sledování je zde použit senzor, který rozděluje snímek na jednotlivé buňky a tyto buňky postupně „aktivuje“, pokud je uvnitř buňky dostatek pixelů patřící do popředí. Pokud jsou postupně aktivovány buňky ve směru předpokládaného pohybu, je „blob“ identifikován jako osoba.

Ostatní zdroje podle [9] vkládají další fázi, „detekci osob“, mezi detekci „blobů“ a sledování. Ta je zde vynechána.

[17] používá místo klasické hloubkové mapy 3D model (viz sekce 4.8). Pomocí bodů z 3D modelu konstruuje paraboly.

Algoritmus postupně prochází po ose x (a poté po ose y) a vždy vybere tři body x_1 , x_2 a x_3 . Těmito body se dá proložit parabola. Pokud se v bodě x_2 nachází hlava osoby, pak musí být tato parabola konkávní.

Parabola je konkávní právě tehdy, když v soustavě rovnic:

$$\delta_1 = w_0 + w_1x_1 + w_2x_1^2$$

$$\delta_2 = w_0 + w_1x_2 + w_2x_2^2$$

$$\delta_3 = w_0 + w_1x_3 + w_2x_3^2$$

je w_2 kladné. Proměnná δ_k v tomto případě značí funkční hodnotu v bodě x_k (hodnotu v hloubkové mapě).

Vzhledem k rozmanitosti tvarů, kterých mohou lidé nabýt (jmenovitě např. člověk na vozíčku nebo člověk nosící klobouk či batoh) tento algoritmus dále

na tvar detekovaného objektu nehledí. Místo toho se soustředí na pohybující se objekty a ty dále sleduje.

Sledování je chápáno jako proces vzniku a zániku (birth-death process) vektorů. Každý objekt je identifikován pomocí jednoho bodu (střed). Body z předchozího a následujícího snímku jsou propojeny a vzniklý vektor je poté analyzován. Pokud se jednotlivé vektory kříží, jsou příliš velké nebo procházejí zdí, jsou zahozeny.

Použití Kalmanova filtru, které je jinak v literatuře časté, naopak [17] považuje za zbytečné, neboť lidé se nechovají jako náhodně se pohybující částice. V tomto případě je ale algoritmus specifický v tom, že sleduje pohyb v jedno-směrné uličce.

Bevilacqua et al. [18] rozdělí snímek na popředí (pohybující se) a pozadí (statické), následně provede přepočítání na 3D model, díky čemuž se zbaví překrývajících se osob.

Poté probíhá iterativní segmentace snímku, které je velice podobná algoritmu k-means. Každý „blob“ má vlastní centroid. Centroidy by měly být od sebe vzdáleny alespoň na šířku hlavy. Každému pixelu je přiřazen centroid, který je mu nejbližší. Bloby, které jsou menší než stanovená konstanta, jsou zahozeny. Nebyl-li nalezen žádný takový blob, je segmentace ukončena.

Výsledek je sledován pomocí Kalmanova filtru.

[19] používá detekci pohybujícího se popředí pomocí Gaussova modelu směsi³ a detekované bloby trasuje pomocí Kalmanova filtru a maďarského algoritmu.

Stejnou detekci popředí používá i [20]. Popředí pak rozdělí pomocí vlastnosti u-disparity. Tato vlastnost popisuje četnost hodnot pixelů v jednotlivých sloupcích hloubkové mapy. Je reprezentovaná pomocí „u-disparity matrix“, tedy matice, jejíž prvek c_{ij} udává, kolikrát se v j -tém sloupci hloubkové mapy vyskytuje pixel s hodnotou i . (Podobně jako u-disparity je definována i vlastnost v-disparity, která však popisuje řádky matice.)

Následně je každá osoba sledována pomocí Kalmanova filtru. Každé osobě na předchozím snímku je přiřazena osoba na tom aktuálním, která je mu nejbližší – ovšem pouze pokud rozdíl v jejich vlastnostech (např. barva) není příliš veliký. V takovém případě je použita některá ze vzdálenějších osob.

Schreiber et al. [21] hledá potenciální osoby pomocí potlačení nemaxim (viz sekce 4.3). Z těchto potenciálních osob vybere skutečné osoby pomocí klasifikátoru SVM. Vstupem pro SVM je histogram rozdílů hloubkových hodnot.

[22] používá pro detekci osob Haarovy příznaky (Haar-like features). Vychází z toho, že každý člověk je na snímku zobrazen jako kruh s vysokou hodnotou (hlava), okolo kterého jsou ze dvou stran ještě ramena, která jsou umístěna o něco níže.

³množina Gaussových pravděpodobnostních rozložení

Z každého snímku je vyroben 3D model a ten následně skenován. Porovnává se prostřední část skenovacího okna R_1 (černá na obr. 3.3) s okrajovými (bílými částmi okna) R_2 . Pokud platí, že

$$\delta_{avg}(R_1) - \delta_{avg}(R_2)/2 > \tau,$$

pak je dané místo označeno jako osoba. Funkce $\delta_{avg}(r_x)$ udává průměrnou výšku ve stanovené oblasti a τ je předem stanovený práh.



Obrázek 3.3: Osoba označená Haarovým příznakem [22]

[23] také používá Haarovy příznaky, avšak trochu jiným způsobem. Počítá s tím, že v okolí osoby je prázdné místo. Kromě toho stejně jako [22] předpokládá, že hlava člověka se nachází uprostřed ramen a má větší výšku než ramena. Dalším rozdílem je, že používá čtyři druhy příznaků, namísto jednoho.

Vyhledávací okénko prochází celý snímek a ohodnocuje jednotlivá místa pomocí klasifikátoru vytvořeného Adaboostem. Vstupem pro tento klasifikátor jsou právě Haarovy příznaky.

Místa klasifikovaná jako osoba jsou pak sledována pomocí Kalmanova filtru.

[24] rozdělí celý snímek na čtvercové bloky o velikosti d_{min} . Tato konstanta značí minimální možný průměr hlavy.

V každém bloku je nalezeno maximum. Následně probíhá kontrola, zda je nalezený bod zároveň maximem ve svém okolí o poloměru d_{min} . Pokud ano, je bod považován za potenciální osobu.

V následující fázi se počítá poměr platných a neplatných pixelů v okolí potenciální osoby. Za platné pixely jsou považovány pixely s hodnotou $d > \tau$. Pokud se počet platných pixelů nachází v zadaném intervalu, na daném místě se nachází osoba.

[5] a [25] sledují osoby na několika kamerách, jejichž obraz se navzájem částečně překrývá.

[5] nejdříve převádí hloubkovou mapu na 3D model. Poté tyto 3D modely spojuje a ve vzniklém modelu provádí detekci. Oproti tomu [25] detekuje a sleduje osoby odděleně a až poté spojuje výsledky.

Detekce u [5] probíhá pomocí odečtení pozadí (pozadí je medián z posledních 20 snímků, pixely příliš podobné pozadí jsou potlačeny). Pixely v popředí jsou rozostřeny pomocí mediánu (každý pixel je spočten jako medián ve svém okolí), aby byl potlačen šum a chybějící pixely. Pak probíhá segmentace snímku popsaná v [26].

Ze snímku jsou odděleny hlavy potenciálních osob, které jsou porovnávány s ručně vytvořeným modelem lidské hlavy. [5] předpokládá, že hlava je

nejvýše položeným místem postavy. Objekty, které neodpovídají modelu jsou zahozeny.

U ostatních potenciálních osob probíhá kontrola, zda mají vedle hlavy viditelná také ramena. To je ověřováno pomocí příznaku HOD (viz dále), který slouží jako vstup pro SVM.

Ke sledování je použit částicový filtr.

[25] pro opravu děr používá stejný algoritmus jako [6]. Poté vytváří upravenou hloubkovou mapu δ_c . V té je hodnota každého pixelu z původní mapy snížena o konstantu c , případně na nulu, pokud by jeho hodnota byla záporná. Díky tomu jsou potlačeny pixely s příliš nízkou hodnotou.

Z mapy δ_c jsou vyseparovány kompaktní oblasti s kladnou hodnotou. Je-li obsah oblasti menší než stanovená hranice, je oblast zahozena. Stejně tak jsou ignorovány nehybné oblasti.

Sledování probíhá za pomoci maďarského algoritmu. Vstupem pro něj je matice vzdáleností mezi detekovanými osobami a osobami na předchozích snímcích. Překročí-li vzdálenost mezi osobami stanovený práh, je nahrazena nekonečnem.

Výsledky sledování všech kamer jsou předány algoritmu, který rozhoduje, zda osoba nalezená na dvou kamerách je tatáž, nebo se jedná o dvě osoby. K tomu se používá vzdálenost v čase t . Aby bylo možné určit vzdálenost mezi osobami, je třeba znát přesné polohy kamer, včetně rotace.

Pokud některá z kamer zachytí snímek dříve nebo později než v čase t , je poloha osoby v tomto čase interpolována z předchozího a následujícího snímku.

[27] a [28] používají pro detekci osob duální záznam z klasických kamer. Kombinací jejich snímků je vytvořena mapa disparity podobná hloubkové mapě zachycené Kinectem či podobným zařízením.

[27] rozděluje mapu disparity na binární obrázky podle výšky. Každý binární obrázek obsahuje pouze pixely s výškou v zadaném intervalu. V těchto obrázcích jsou hledány hlavy osob.

Intervaly jsou nastaveny tak, aby jeden z binárních obrázků zobrazoval z člověka pouze hlavu.

Na binární obrázek je aplikován morfologický operátor otevření⁴. Poté jsou v obrázku nalezeny shluky, které obsahují hlavu člověka.

Nalezeným hlavám jsou přiřazeny příznaky, které slouží ke správnému přiřazení osob během sledování. Těmito příznaky jsou: velikost hlavy, její šířka a délka, průměrná výška, průměrný odstín šedi (v původních duálních snímcích) a souřadnice na ose x a y .

Samotné sledování probíhá za pomoci Kalmanova filtru.

⁴Jedná se o složení eroze a dilatace: $(A \ominus B) \oplus B$. Výsledkem otevření je binární obraz podobný tomu původnímu. Liší se však například v tom, že jsou odstraněny malé shluky a menší „záhyby“.

[28] se snaží zachytit kamerami velkou plochu, k čemuž používá sérii čoček. Obraz je proto obzvláště na okrajích zkreslený. Řešením tohoto problému je převedení mapy disparity na 3D model.

Tento 3D model je následně převeden na „occupancy map“ [29].

Segmentace occupancy map probíhá pomocí Gaussova modelu směsi. Každý pixel má přiřazen jeden model směsi. Tento model se dá popsat jako množina n Gaussových rozložení, kde n je počet osob detekovaných na předchozím snímku.

Každý ze segmentů zobrazuje jednu osobu. Segmenty jsou sledovány s využitím Kalmanova filtru.

3.2 Detekce osob s využitím barevného záznamu

Tato kapitola obsahuje shrnutí zdrojů, které popisují detekci a sledování osob na barevném záznamu.

[30] používá pro detekci osob algoritmus k-means. Vstupem pro něj je snímek s odstraněným pozadím.

K-means vyžaduje předem pevně stanovený parametr k . Ten je určen jako maximální možný počet osob na snímku, tedy počet osob v případě, že je celý snímek zaplněn lidmi s hlavou o poloměru d_{min} . Výstupem k-means je několik shluků, zpravidla více, než je osob na snímku. Shluky, které jsou od sebe blíže než d_{min} , jsou považovány za jednu osobu.

Sledování osob je prováděno hladovým algoritmem.

Barandiaran et al. [31] počítá procházející osoby v uličce. Předpokládá, že osoby v uličce se pohybují pouze dvěma opačnými směry.

Pro počítání používá „counting lines“, tedy úsečky kolmé na směr pohybu. Ve sledované oblasti je několik „counting lines“ vedle sebe.

Osoby jsou detekovány pomocí jejich pohybu. Mezi každým snímekem I^t zachyceným v čase t a předchozím snímekem I^{t-1} je spočtena matice rozdílů D^t . Prvky této matice jsou definovány následovně:

$$D_{xy}^t = \begin{cases} 1 & \text{jestliže } I_{xy}^t - I_{xy}^{t-1} > \tau \\ 0 & \text{v opačném případě} \end{cases}$$

Pohyb osoby je tedy zachycen jen ze dvou následujících snímků. Důvodem je vysoká odolnost vůči změně osvětlení a nízká výpočetní náročnost.

Každý bod ležící na „counting line“ má přiřazenu proměnnou l_k , kde k je jeho poloha na counting line. Tato proměnná popisuje změnu bodu na „counting line“. Proměnná l_k v čase t je značena jako l_k^t a její hodnota je definována jako:

$$l_k^t = l_k^{t-1} + D_{xy}^t.$$

Při průchodu osoby přes „counting line“ se tedy postupně zvyšují hodnoty l_k v místech, kudy osoba prochází. Jestliže se l_k nemění dostatečně dlouho, je opět nastavena na 0.

Pokud je v čase t na „counting line“ souvislý pás s dostatečně velkou hodnotou l_k^t , pak v čase t přes „counting line“ procházel člověk. Tento pás však musí zabírat alespoň polovinu šířky průměrné osoby.

Z naměřených hodnot se počítá směr pohybu jednotlivých osob pomocí optického toku a metody Lucas–Kanade [32].

Mukherjee et al. [33] využívá Cannyho detekci hran. V těchto hranách pak hledá kružnice pomocí detektoru založeném na Houghově transformaci. Zde vychází z toho, že lidská hlava má při pohledu shora přibližně kruhový tvar.

Pro sledování využívá Horn–Schunckovu metodu odhadu optického toku [34].

Poslední fází je odstranění chybných trajektorií (neexistujících, omylem detekovaných osob). Ta je provedena jednak pomocí extrakce popředí (není-li stanovené procento pixelů náležejících osobě vyšší než je stanovená hranice, je osoba považována za chybu měření) a taktéž pomocí detekce překryvů dvou osob (mají-li dvě osoby příliš společných pixelů, jedná se pravděpodobně pouze o jednoho člověka).

Dalal & Triggs [35] představují HOG (histograms of oriented gradients), které používají k detekci osob.

HOG je kolekce histogramů, které popisují snímek. Vstupní snímek je rozdělen do čtvercových buněk. V každé buňce je vytvořen histogram gradientů pixelů. Buňky jsou poté sdruženy do čtvercových bloků. Histogramy v každém bloku jsou normalizovány.

Dalal & Triggs využívají HOG pro klasifikaci obrázků pomocí SVM. Díky tomu tak detekují osoby, které jsou snímány z boku.

Zhu [36] používá pro detekci osob také HOG. Májí-li HOG velkou velikost bloku, nedokáže dostatečně dobře detekovat osobu. Proto Dalal & Triggs používají blok menší. HOG s malým blokem jsou však pomalé. Z toho důvodu Zhu používá kaskádu HOG, kde je snímek nejdříve prohledán pomocí HOG s velkým blokem a poté je velikost bloku postupně snižována.

Ohodnocování HOG probíhá pomocí AdaBoost, který používá SVM jako slabý klasifikátor.

[37] nejdříve detekuje popředí pomocí Gaussova modelu směsi. Malé oblasti v popředí zahazuje, větší klasifikuje pomocí kaskády HOG představené v [36].

Úkolem [38] je identifikovat osoby zachycené shora. Kromě detekce rozlišuje snímky osob mezi sebou, aby našel stejné osoby.

Samotná detekce je oproti ostatním jednodušší v tom, že poloha a směr natočení osoby jsou předem přibližně známy.

K detekci používá odečtení pozadí. Protože je barevný snímek náchylný na změny osvětlení, používá více druhů pozadí zachycených v různých světelných podmínkách. Snímek osoby je porovnán se všemi pozadími, pro každé pozadí

je spočten rozdíl jasu ve všech čtyřech rozích. Vybráno je to pozadí, kde je celkový rozdíl nejmenší.

Součástí popředí je detekovaná osoba, stín a malé oblasti, které vznikly kvůli šumu. Oblasti vzniklé šumem jsou rozpoznatelné na základě jejich velikosti. Stín je pak rozpoznáván pomocí velikosti a jeho polohy (bývá promítnut na jedno místo).

K rozlišení osob slouží tři příznaky: velikost, barva vlasů a účes. Účes je vytvořen pomocí detekce hran na hlavě. Kromě těchto tří [38] uvádí ještě čtvrtý – „vír vlasů“ – tedy „účes“ po rozostření snímku. Z účesu a víru jsou vytvořeny HOG.

Výše popsané příznaky jsou klasifikovány pomocí AdaBoost.

3.3 Hybridní řešení

Někteří autoři k řešení problému sledování osob používají barevný kamerový záznam i hloubkové mapy.

[39] používá hloubkovou mapu pro samotnou detekci osob a barevný záznam pro následnou analýzu.

Snímek je zde rozdělen na popředí a pozadí a rozdělen na segmenty (segment je oblast pixelů patřících do popředí, které jsou od ostatních oblastí odděleny pozadím). V každém segmentu je nalezen nejvyšší bod, který reprezentuje celého člověka v další, sledovací, fázi. Test, zda se v jednom segmentu nenachází více osob, prováděn není.

Všem nalezeným bodům jsou vypočteny souřadnice ve 3D prostoru, které slouží jako vstup sledovacího algoritmu.

Sledování je uskutečněno s pomocí Kalmanova filtru.

Každá osoba má vypočtenou trasu, kterou prošla. Je-li tato trasa příliš krátká, dlouhá nebo celkově složitá, je zahozena.

Protože úkolem [39] je zjištění počtu osob v daném prostoru, snaží se i reidentifikovat osoby, které prostorem prošly a zase se vrátily. K tomu používá i barevné snímky.

Z barevného snímku je odstraněna hlava (pomocí hloubkových dat) a další analýza je prováděna na zbytku těla, kde je možné podle barvy oblečení jednotlivé osoby rozpoznat. Samotná identifikace osob probíhá za pomoci histogramu barev.

I [40] používá pro detekci čistě hloubková data a barevný snímek je využit až v trasovací fázi. Pro detekci osob je však potřeba data opravit, protože v nich mohou chybět některé body.

Z hloubkové mapy je vytvořen binární obrázek (hloubková mapa je rozdělena na body vyšší než τ a ostatní). Na obrázek jsou aplikovány morfologické operace uzavření a posléze dilatace.

Neplatné (nulové) body, které binární mapě mají hodnotu true, se poté obnovují podle nejbližšího platného souseda.

Takto opravený snímek se rozdělí na několik binárních obrázků. Každý z nich obsahuje informaci, zda pixely v hloubkové mapě na dané souřadnici mají hodnotu v rozmezí mezi dvěma stanovenými hranicemi. Pro každý binární obrázek jsou tyto hranice jiné.

Pokud binární obrázek obsahuje shluk pixelů o rozměrech běžné hlavy, pak je dané místo označeno jako potenciální osoba.

Kvůli možným chybám je třeba ověřit, zda se jedná o reálnou osobu či nikoliv. To se provádí tak, se ze snímku vyberou pixely popisující hlavu (90 % výšky osoby), jejichž počet by měl náležet do stanoveného intervalu, a dále pixely popisující „torzo“ (70 % výšky), kterých je n -krát více než pixelů hlavy, přičemž n musí rovněž náležet do stanoveného intervalu.

Během sledování osob se každé osobě na předchozím snímku přiřadí několik osob na tom současném (zpravidla jedna osoba) a naopak každé osobě na aktuálním snímku je přiřazeno několik osob na tom předešlém.

Důvodem k tomuto řešení je možnost, že se několik osob „slilo“ do jednoho „blobu“ a detekční algoritmus je považuje za jednu osobu. Tato situace by však neměla být příliš častá, protože algoritmus se snaží osoby rozdělit, pokud alespoň jejich hlavy identifikuje jako dva objekty.

Každé osobě na jednom snímku je přiřazena na druhém ta, která má minimální „matching score“. Toto „score“ je spočteno jako součet Bhattacharyyovy vzdálenosti⁵ histogramů barev a „depth similarity“ (veličina závislá na podobnosti hodnot v hloubkové mapě).

[42] řeší detekci osob (v tomto případě je ovšem snímání zařízení umístěno z boku) pomocí metody HOD (Histogram of oriented depths). Algoritmus je velice podobný metodě HOG (Histogram of oriented gradients) představené v [35] (popsáno výše). HOG se však zaměřuje na barevné snímky a ne na hloubkovou mapu.

Způsobem, jakým algoritmus funguje, se však od HOG příliš neliší.

HOG nezávisí na samotných hodnotách barev, ale na jejich rozdílu, díky čemuž funguje lépe i na snímcích pořízených za různých světelných podmínek, kde jsou stejné objekty různě barevné. I v hloubkové mapě může být tato robustnost výhodou, protože HOD funguje nezávisle na výšce objektu.

HOD slouží jako vstup pro klasifikační algoritmus, kterým je SVM (stejně jako v [35]).

[42] používá nejen HOD a hloubkovou mapu, ale i odpovídající barevné snímky používající HOG a výstupy z těchto metod pak kombinuje.

⁵ $D_B(A, B) = -\ln \rho(p(A), p(B))$ [41]. D_B je Bhattacharyyova vzdálenost, $p(A)$ pravděpodobnost jevu A a ρ Bhattacharyyův koeficient: $\rho = \int_{\Omega} \left(\frac{dp(A)}{d\nu} \frac{dp(B)}{d\nu} \right)^{\frac{1}{2}}$, kde ν je stupeň volnosti

3.4 Zobecnění

V této kapitole popsané algoritmy můžeme obecně rozdělit do čtyř fází:

1. Preprocessing – zpravidla oprava obrazu: vyplnění děr v hloubkové mapě, odstranění šumu, potlačení pozadí...V některých algoritmech preprocessing zcela chybí, problémy obrazu mohou být řešeny použitím robustnějšího algoritmu v detekční fázi.
2. Detekce potenciálních osob – nalezení míst, kde by se mohli vyskytovat lidé. Určitá rozumná míra false positive nevádí, důležité je, aby byli v této fázi nalezeni všichni lidé.
3. Klasifikace potenciálních osob – místa nalezena v předchozím kroku jsou zkoumána klasifikátorem, který podle vzhledu objektu na snímku zjistí, zda se jedná o člověka, nebo ne.
4. Sledování – Nově detekované osoby jsou přiřazeny k osobám na předchozích snímcích. Díky tomu se dá zjistit, jak se daná osoba pohybovala. Sledování rozněž pomáhá potlačit občasné nepřesnosti předchozích fází.

Vybrané metody

V této kapitole se zabývám algoritmy, které sice neřeší problém sledování osob jako takový, ale je možné je využít k řešení jeho podproblémů. Některé z těchto algoritmů se vyskytují již v předchozí kapitole, kde však nebyly dopodrobna probrány.

4.1 Narůstání oblastí

Metoda narůstání oblastí (region growing) slouží k detekci lokálních maxim a segmentaci obrázku.

Lokální maxima hledá tak, že postupně hledá globální maximum snímku a pak postupně maže všechny sousedící body od tohoto místa níže.

Metoda vychází z [7], používá ji i [6]. Pracuje následujícím způsobem:

Algoritmus 1: Narůstání oblastí

vstup : obrázek I

výstup: množina M_{detect} obsahující nalezené body

```
1  $M_{detect} \leftarrow \emptyset$ 
2 while  $I$  obsahuje nenulovou hodnotu do
3    $G \leftarrow \text{global\_max}(I)$ 
4    $M \leftarrow \{G\}$ 
5   do
6      $M_{neigh} \leftarrow \{A | \exists B : |AB| \leq 1; I[A] < I[B]; B \in M\}$ 
7      $M \leftarrow M \cup M_{neigh}$ 
8     for  $B \in M$  do
9        $I[B] = 0$ 
10    end
11  while  $M_{neigh} \neq \emptyset$ 
12   $M_{detect} \leftarrow M_{detect} \cup \{G\}$ 
13 end
```

Narůstání oblastí může být použito také k segmentaci obrázku [6]. K tomu je nutné provést úpravu zde popsaného algoritmu. Upravený algoritmus vrací množinu „blobů“ – tedy množinu množin bodů. Úprava spočívá ve změně řádku 12, kde je nutné do množiny M_{detect} přidávat celou množinu M místo bodu G .

Segmentační algoritmus, který používá [6] vrací pouze bloby, které mají hodnoty v intervalu $(I[G] - 100, I[G])$. Toho lze docílit přidáním podmínky $I[A] > I[G] - range$ definující prvky množiny na řádku 6.

4.2 Water filling

Algoritmus Water filling slouží k detekci (potenciálních) osob, a to za pomoci nalezení lokálních maxim. Dovede potlačit šum a nízké objekty.

Algoritmus je představen v [8] a dále použit např. v [16].

Hlavní idea algoritmu je inspirovaná vlastnostmi vody při dopadu kapky. Ta po dopadu na šikmý povrch nejprve teče dolů, a následně vytvoří spolu s ostatními kapkami rovnou hladinu. Nejvíce vody ulpí v nejnižších oblastech, dají se tak tedy najít lokální minima (resp. maxima při použití inverzních hodnot).

Během detekce lokálních maxim je vytvořena speciální mapa g , která symbolizuje „vodu“, která dopadá na původní hloubkovou mapu.

Tvorba mapy g se dá popsat následovně:

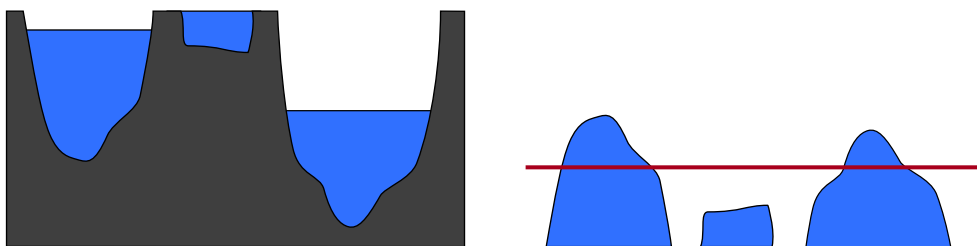
Algoritmus 2: Water Filling

vstup : hloubková mapa δ o velikosti $w \times h$, konst. c_{drop} , počet iterací n

výstup: matice g

```
1  $g \leftarrow$  nulová matice o velikosti  $w \times h$ 
2 for  $i \in 1 \dots n$  do
3   |  $A, A \in \delta$ , určen náhodně či jinak
4   |  $dopad\_kapky(A)$ 
5 end
6 Funkce  $dopad\_kapky(A)$ 
7   | do
8   |   |  $A_{last} \leftarrow A$ 
9   |   | for  $B \in okolí(A)$  do
10  |   |   | if  $\delta_B - g_B > \delta_A - g_A$  then
11  |   |   |   |  $A \leftarrow B$ 
12  |   |   | end
13  |   | end
14  | while  $A_{last} \neq A$ 
15  |  $g_A \leftarrow g_A + c_{drop}$ 
```

Ve výsledné mapě g jsou hledány oblasti, jejichž hodnota je vyšší než práh τ (viz obr. 4.1).



(a) Šedé pole značí funkci δ , tedy snímek ze záznamu. V tomto případě jsou data obrázku převrácena (čím vyšší je objekt, tím má pixel nižší hodnotu), aby mohla být ilustrována inspirace nádobami plněnými vodou. Modré pole, tedy „voda“ je funkce g .

(b) Samotná funkce g (modrá), vodorovná čára znázorňuje stanovenou hranici, která odděluje potenciální osoby od velmi nízkých objektů a šumu

Obrázek 4.1: Znázornění metody Water Filling na jednorozměrném snímku

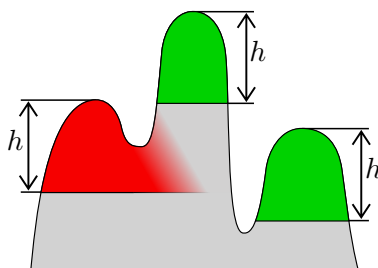
4.3 Potlačení nemaximálních hodnot

Morfologický operátor potlačení nemaximálních hodnot (angl. non-maximum suppression) ponechává v obrázku pouze ty pixely, které nemají v okolí žádný bod s hodnotou vyšší, než mají ony.

Pixely se nacházejí v okolí, pokud mají od středového pixelu vzdálenost d nebo nižší. Metrika vzdálenosti bývá manhattanská⁶ nebo eukleidovská⁷.

4.4 H-maxima

H-maxima je morfologický operátor, který potlačuje všechny „kupole“, které nejsou dostatečně vysoké [43]. Jeho funkci zobrazuje obr. 4.2.



Obrázek 4.2: Ukázka funkčnosti operátoru h-maxima na jednorozměrném snímku

⁶Součet vzdáleností na ose x a y . Prohledávání okolí s takovouto metrikou je implementačně jednodušší, protože se dá použít jednoduchý for-cyklus nebo dynamické programování.

⁷Klasická vzdálenost mezi dvěma body tak, jak ji intuitivně chápe člověk. Ve dvourozměrném prostoru se počítá pomocí Pythagorovy věty.

Algoritmus **extended h-maxima** vrací lokální maxima z výstupu operátoru h-maxima [43]. Činnost algoritmu lze popsat následovně:

Algoritmus 3: Extended h-maxima

vstup : obrázek I , konstanta h

výstup: obrázek J

```
1 for  $A \in I$  do
2   if is_h_max( $A$ ) then
3      $J_A \leftarrow I_A$ 
4   else
5      $J_A \leftarrow 0$ 
6   end
7 end
8 Funkce is_h_max( $A$ )
9    $max = I_A$ 
10   $M \leftarrow \{A\}$ 
11  do
12     $N \leftarrow \{B | B \notin M; \exists C \in M : B \text{ je soused } C\}$ 
13    for  $B \in N$  do
14      if  $I_B > max$  then
15        return false
16      end
17      if  $I_B \leq max - h$  then
18         $N \leftarrow N \cap \{B\}$ 
19      end
20    end
21  while  $N \neq \emptyset$ 
22  return true
```

4.5 Problém přiřazení

Problém přiřazení (assignment problem) spočívá v přiřazení každého prvku z množiny A nějakému prvku z množiny B . Každý prvek z A musí být přiřazen jinému prvku B . Přiřazení každé dvojice prvků je ohodnoceno cenovým koeficientem. Úkolem problému přiřazení je minimalizovat (nebo maximalizovat) celkovou cenu [44]. Cenu spočteme jako součet cenových koeficientů všech vybraných přiřazení.

Obě množiny by měly být stejně velké. Je-li jedna množina menší, přidávají se do ní fiktivní prvky s nulovým cenovým koeficientem [44].

Problém přiřazení je možné popsat pomocí matice, kde sloupce popisují prvky množiny A a řádky prvky množiny B . Každý prvek c_{ij} udává cenový koeficient přiřazení j -tého prvku z množiny A i -tému prvku z množiny B .

Problém přiřazení se tak dá popsat jako minimalizace následujícího výrazu [44, 45]:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

Proměnná x_{ij} nabývá hodnoty 1, pokud je i -tý prvek přiřazen j -tému prvku, nebo 0 v opačném případě.

4.6 Maďarský algoritmus

Maďarský algoritmus [46] řeší problém přiřazení (viz předchozí sekce). Dá se tedy využít pro sledování osob.

Algoritmus pracuje následovně [45]:

1. Redukce matice
2. Výběr nezávislých nul
3. Kontrola správnosti výběru
4. Sekundární redukce matice
5. Zpět na bod 2, pokud není nalezeno stejně nul, jako je řádků (a sloupců) matice
6. Řešením je přiřazení těch prvků, které mezi sebou mají v matici nezávislou nulu

4.6.1 Redukce matice

Během redukce je v každém řádku matice nalezen nejmenší prvek. Ten je odečten od všech prvků v řádku [45]. Minimálně jeden prvek řádku by tak měl zůstat nulový.

Tentýž postup je aplikován na sloupce.

4.6.2 Výběr nezávislých nul

V dalším kroku probíhá vyhledávání prvků o nulové hodnotě. Nezávislou nulou se rozumí nulový prvek, pro který platí, že $x_{ij} = 1$ [44].

Heuristiku pro výběr nul popisuje [44] následovně (upraveno):

1. Pro všechna $c_{ij} = 0$ spočítáme počet k_{ij} všech dalších nul v i -tém řádku a j -tém sloupci.
2. Vyhledáme volnou nulu s maximální hodnotou k_{ij} a označíme jí jako nezávislou

3. Vyškrtneme řádek i sloupec, ve kterém se nula nachází. Pokud se v matici nachází alespoň jedno nevyškrtnuté políčko s nulovou hodnotou, pokračujeme bodem 1. V opačném případě jsme našli počáteční soustavu nezávislých nul.

4.6.3 Kontrola správnosti výběru

Podle Köning-Egerványho věty platí, že

Maximální počet nul, který je možné vybrat z redukované matice sazeb, je roven minimálnímu počtu čar, kterými lze pokrýt všechny nuly v matici.

(citace podle [45])

4.6.4 Sekundární redukce matice

Sekundární redukce pracuje s minimální pokrytím nul. Všechny prvky matice jsou rozděleny na „přeškrtnuté“ a „nepřeškrtnuté“. Přeškrtnuté jsou ty, kterými prochází čára obsažená v minimálním pokrytí všech nul. Dvakrát přeškrtnuté pak ty, kterými prochází dvě takové čáry.

Nejdříve je nalezeno minimum ze všech nepřeškrtnutých prvků. Toto minimum je odečteno od všech nepřeškrtnutých prvků a přičteno ke všem dvakrát přeškrtnutým.

4.7 Kalmanův filtr

Kalmanův filtr je algoritmus používaný k odhadu dat měřených senzorem, který obsahuje šum [47]. Měření dat musí probíhat v diskrétním čase [48].

Maybeck [49] popisuje Kalmanův filtr jako „optimální rekurzivní algoritmus pro zpracování dat“.

Kalmanův filtr se může se nacházet ve dvou stavech, které se neustále střídají.

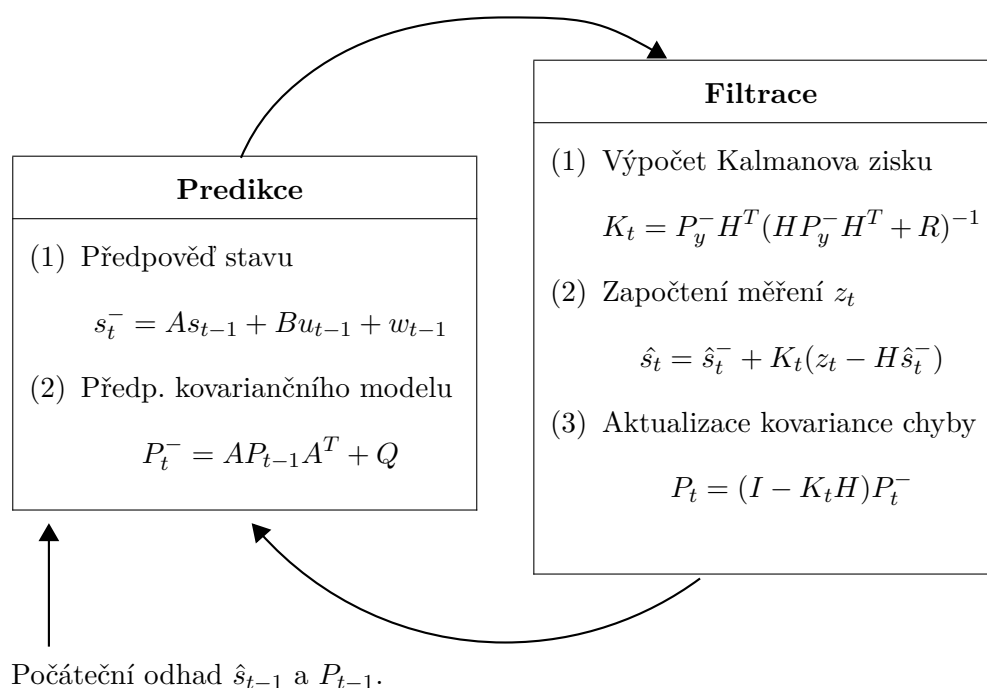
V prvním stavu, který se nazývá „predikce“ [50] (prediction), předpovídá stav veličiny v příštím měření.

Druhý stav je „filtrace“ [50] (update), kdy algoritmus využívá nejnovější měření, aby odhadl současný stav veličiny.

Při každém měření proběhne odhad současného stavu (filtrace) a následně predikce dalšího stavu.

Odhadovaná veličina \vec{s} má podobu vektoru. Jednotlivé složky vektoru na sobě mohou být závislé⁸, například poloha osoby a její rychlost. Této vlastnosti umí Kalmanův filtr využít.

⁸Znalost jedné veličiny zlepšuje odhad té druhé



Obrázek 4.3: Princip fungování Kalmanova filtru. Podrobnější vysvětlení je obsaženo v textu.

Kalmanův filtr pro odhady používá lineární stochastický model [50]. Jeho funkce je popsána v následujícím textu.

4.7.1 Predikce

Během predikce je stav veličiny s v čase t odhadnut jako:

$$s_t^- = As_{t-1} + Bu_{t-1} + w_{t-1}.$$

Proměnné použité v této rovnici jsou následující:

- s_t^- – predikce veličiny v čase t ,
- A – matice, která popisuje vztah veličiny s v čase $t-1$ a stav této veličiny v čase t ,
- u – známé vnější změny, například poloha plynového pedálu u filtru odhadujícího rychlost auta,
- B – matice, která popisuje, jaký vliv mají známé vnější změny u_t na s_t ,
- w – „process noise“ – šum (náhodná veličina s normálním rozložením).

Dále je predikována kovarianční matice P . Jednotlivé prvky P_{ij} kovarianční matice udávají kovarianci mezi i -tou a j -tou složkou veličiny \vec{s} .

$$P_t^- = A_t P_{t-1} A_t^T + Q$$

Proměnná Q značí kovarianci náhodné veličiny „process noise“.

4.7.2 Filtrace

V čase t dostane filtr výsledek měření z_t . Ten je zpravidla nepřesný a je možné jej vyjádřit pomocí matice H , stavu s a šumu v jako:

$$z_t = H s_t + v_t$$

Proměnné použité v této rovnici jsou následující:

- z – výstup senzoru,
- H – matice popisující vliv s_t na výstup senzoru,
- s – skutečný stav veličiny,
- v – „measurement noise“ – šum vzniklý nepřesností měření (náhodná veličina s normálním rozložením).

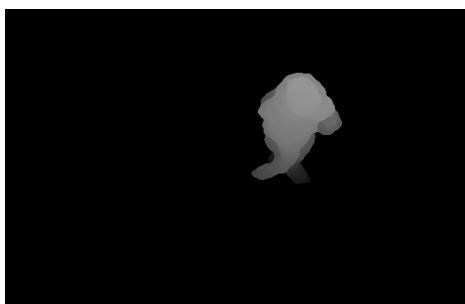
Stav veličiny s po provedení měření je Gaussovo rozložení. Toto rozložení se dá popsat pomocí střední hodnoty \hat{s} a matice kovariance P_k .

K výpočtu střední hodnoty je třeba znát Kalmanův zisk K . Jedná se o matici o velikosti $n \times m$, kde m a n jsou počet složek vektorů \vec{s} a \vec{z} . Kalmanův zisk je stanoven tak, aby byla minimalizována matice P_k , tedy aposteriorní kovariance chyby. (Aposteriorní chyba je v tomto případě $s_t - \hat{s}_t$.) Způsob minimalizace chyby a výpočtu K se může lišit [47], nejčastěji zmiňovaný je však výpočet podle následující rovnice [47]:

$$K_t = \frac{P_y^- H^T}{H P_y^- H^T + R}$$

Pomocí K je pak možné spočítat střední hodnotu

$$\hat{s}_t = \hat{s}_t^- + K_t(z_t - H_t \hat{s}_t^-).$$



(a) Osoba uprostřed snímku je snímána shora

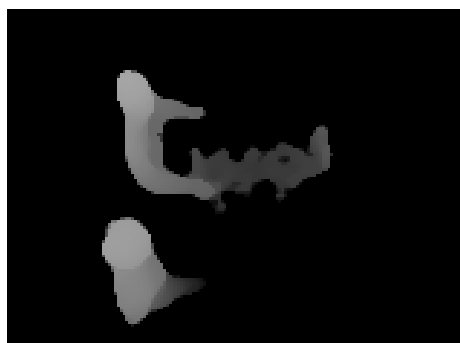


(b) Osoba na okraji snímku je snímána částečně z boku

Obrázek 4.4: Disproporce obrazu osoby v závislosti na jejím umístění na snímku



(a) 3D model



(b) Původní snímek

Obrázek 4.5: Převod snímku na 3D model

4.8 Převod hloubkové mapy na 3D model

Snímá-li jedno zařízení velkou plochu, může dojít k tomu, že na okrajích snímků jsou osoby snímány částečně z boku (viz Obrázek 4.4). To poté znesnadňuje jejich detekci, protože se osoby mohou překrývat, není možné se spolehnout na kruhový tvar hlavy a podobně.

Řešení tohoto problému je projekce všech osob do půdorysu, kterou používá řada zdrojů, např. [17, 18, 28, 39, 20] a další. Vzniká tak 3D model postavy, který se nemění v závislosti na umístění osoby na snímku. Promítnutí do půdorysu je provedeno pro každý naměřený pixel zvlášť, [18] uvádí následující vzorec:

$$x_P = (x - x_c) \cdot k_x \cdot \frac{\delta' - f}{d'}$$

$$y_P = (y - y_c) \cdot k_y \cdot \frac{\delta' - f}{d'}$$

kde x_P je výsledná pozice daného pixelu na ose x v půdorysu, x je pozice pixelu na ose x v původním snímku, x_c určený referenční bod (střed), k_x je konstanta sloužící k regulaci velikosti výsledného snímku, d' je naměřená hodnota a f ohnisková vzdálenost snímacího zařízení a δ' je vzdálenost od snímacího zařízení, tedy $\delta' = maximum - \delta$, kde δ je původní hodnota hloubkové mapy.

Značení pro pozici na ose y je obdobné.

[20] rozlišuje ohniskové vzdálenosti na jednotlivých osách, krom toho samotný výpočet probíhá trochu jinak:

$$x_P = (x - x_c) \cdot \frac{\delta'}{f_x}$$

Výška v 3D modelu je v obou případech rovna původní hodnotě δ .

Někteří autoři, např. [28], k tvorbě tohoto modelu využívají stereo barevný záznam, postup je podobný.

4.9 Watershed

Transformace Watershed slouží k segmentaci snímku. K segmentaci potřebuje několik bodů, následně každému z nich přiřadí oblast.

Průběh transformace je možné ilustrovat pomocí krajiny, která je během povodně postupně plněna vodou [51]. Na začátku jsou pod vodou pouze výchozí body, ideálně lokální minima. Voda pak postupně zaplavuje výše položená místa (tedy pixely s vyšší hodnotou).

Každý z bodů má definované „povodí“, tedy zaplavenou část krajiny, která bod obklopuje. V případě, že se dvě povodí setkají, je mezi nimi postavena „hráz“. Nakonec voda zaplaví celý snímek.

Výsledkem transformace je několik oblastí oddělených hrázemi.

Návrh a implementace

V této kapitole popisují návrh algoritmu detekce a sledování, použité metody a zároveň také metody, které byly testovány, ale nepoužity.

Nejdříve je popsán celkový návrh algoritmu. Ten je rozdělen do sedmi fází. Podrobný popis metod testovaných v těchto fázích se nachází dále v sekcích 5.2–5.8. Popsány jsou zde jak metody využití v mém algoritmu, tak i metody testované a nepoužité.

Popis každé fáze obsahuje informaci o tom, které z metod byly využity. Pro snadnější orientaci je však sekce 5.9 – Shrnutí zmiňuje v jednoduchém seznamu. Na závěr této kapitoly je zařazen popis implementace.

5.1 Obecný návrh

Jak již je řečeno v sekci 3.4, dostupné algoritmy se dají rozdělit na 4 fáze: preprocessing, detekce potenciálních osob, ověření potenciálních osob a sledování.

Ve své práci se budu tohoto schématu držet, udělal jsem v něm však několik změn. Mezi tyto změny patří rozdělení „detekce potenciálních osob“ na samotný detekční algoritmus a na úpravu výsledků detekce (viz sekce 5.4). Další změnou je rozdělení ověření pravosti potenciální osoby do tří fází – „segmentace“, „deskripce“ a „klasifikace“ (viz sekce 5.5, 5.6 a 5.7). Důvodem k těmto změnám je vyšší přehlednost celé práce.

Schéma mého návrhu je načrtnuto v obrázku 5.1 a obsahuje následující fáze:

- **Preprocessing** – popsán v sekci 5.2
- **Detekce** má za úkol najít místa, kde by se mohly nacházet osoby. Tyto potenciální osoby jsou pak analyzovány dále. Vstupem jsou obrazová data, výstupem souřadnice míst, kde by se mohli nacházet lidé. Detailnější popis se nachází v sekci 5.3.

- **Úprava výsledku detekce** slučuje příliš blízké potenciální osoby a mění jejich polohu tak, aby byla jednodušeji zpracovatelná následujícími algoritmy.

Snahou této fáze je, aby každá osoba byla identifikována pouze pomocí jednoho bodu. Další informace jsou v sekci 5.4.

- **Segmentace** rozděluje vstupní snímek mezi jednotlivé (potenciální) osoby.

Vstupem jsou obrazová data a seznam potenciálních osob. Výstupem je pak pro každou osobu jeden obrázek. Detailnější popis se nachází v sekci 5.5.

- **Deskripce** znamená vytvoření příznaků – jednoduchých hodnot, které popisují osobu. Mezi příznaky patří výška osoby, velikost její hlavy, průměr osoby atd., viz sekce 5.6.

Vstupem pro deskripci jsou snímky jednotlivých osob, tedy výstup segmentace. Výstupem jsou čísla popisující konkrétní osoby, případně vektor.

- **Klasifikace** používá statistický model, který rozhoduje, nakolik je pravděpodobné, že potenciální osoba je skutečně osobou a nikoli jiným objektem.

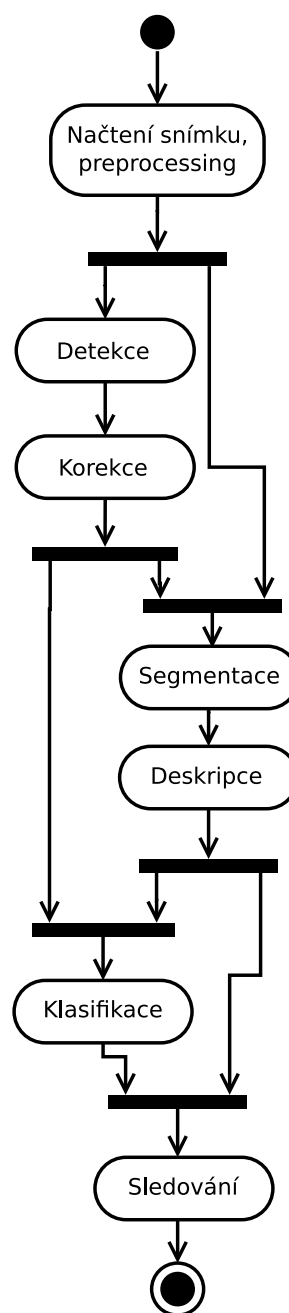
Vstupem je tedy seznam potenciálních osob a jejich příznaků, výstupem je seznam potenciálních osob a jim přiřazená pravděpodobnost.

Podrobnější popis se nachází v sekci 5.7.

- **Sledování** přiřazuje osobám nalezeným na předchozích snímcích osoby na současném snímku. Jako jediná část si pamatuje historii a využívá informace z předchozích snímků.

Využívá výstup klasifikace a příznaky, které umožní rozpoznat jednotlivé osoby mezi sebou.

Výstupem je seznam souřadnic a k nim přiřazených ID. Tyto ID jsou unikátní identifikátory každé osoby. Další podrobnosti je možné nalézt v sekci 5.8.



Obrázek 5.1: Celkový návrh algoritmu detekce a sledování osob

5.2 Preprocessing

Preprocessing spočívá v úpravě vstupního snímku tak, aby byla zvýšena přesnost následujících algoritmů. V této fázi by měly být opraveny všechny chyby, které negativně ovlivňují úspěšnost detekce, tedy šum a chybějící místa.

5.2.1 Eliminace šumu

Některé detekční metody spoléhají na to, že každý člověk je reprezentován jedním lokálním maximem v hloubkové mapě. Tímto maximem je zpravidla temeno hlavy.

S snímkem jedné osoby se však může v určitých případech nacházet více lokálních maxim.

Důvodem tohoto jevu je šum na snímcích – tedy pixely s hodnotou vyšší nebo nižší, než je vzdálenost od Kinectu ve skutečnosti. Šum bývá způsoben nedokonalostmi snímacího zařízení.

Pomocí rozostření je možné šum potlačit. V následujícím textu jsou uvedeny dvě testované rozostřovací techniky – Gaussovo rozostření a rozostření pomocí mediánu.

V Gaussově rozostření je rozostřený snímek b spočten jako konvoluce původní hloubkové mapy d a Gaussovy funkce G . Vzorec pro výpočet každého bodu $[x, y]$ je tedy následující:

$$b_{xy} = \sum_{i=0}^{\text{šířka}} \sum_{j=0}^{\text{výška}} d_{xy} G(i-x, j-y),$$

kde

$$G(a, b) = \frac{1}{2\pi\sigma^2} e^{-\frac{a^2+b^2}{2\sigma^2}}.$$

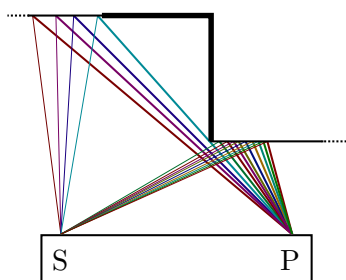
Směrodatná odchylka σ je stanovena před začátkem výpočtu.

Gaussovo rozostření je schopné potlačit menší nerovnosti. Při volbě větší hodnoty σ je možné dosáhnout toho, že jediným lokálním maximem je temeno hlavy.

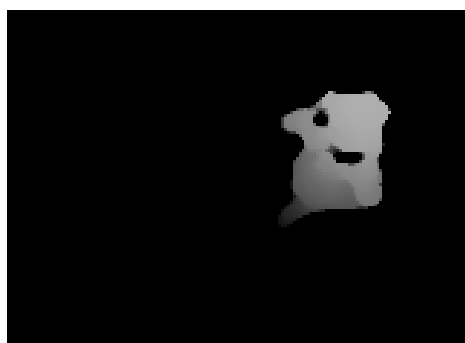
Dalším testovaným rozostřením bylo rozostření pomocí mediánu. Zde je hodnota každého pixelu spočtena jako medián všech pixelů v jeho okolí. Okolím je myšlen čtverec se středem v daném pixelu.

Rozostření pomocí mediánu zcela eliminuje drobný šum. Gaussovo rozostření jej naopak v některých případech může pouze výrazně snížit. Příkladem je situace, kdy snímek obsahuje velkou plochu se stejnou hodnotou, uprostřed které je pixel s hodnotou vyšší.

Třetím testovaným rozostřovacím algoritmem byl algoritmus, který pixelu přiřadí nejvyšší hodnotu v okolí.



Obrázek 5.2: Vznik děr. Zařízení Kinect (dole) snímá objekt v horní části obrázku. Projektor (vpravo) promítá infračervené paprsky na jednotlivá místa snímaného objektu, ty jsou pak snímány senzorem. Na některá místa není možné paprsek promítnout, proto zůstanou senzorem nezachycena.



(a) Díry na místě ramen a krku



(b) Obzvláště u osob na okraji snímku může dojít k oddělení hlavy a těla

Obrázek 5.3: Díry v hloubkové mapě

5.2.2 Oprava děr

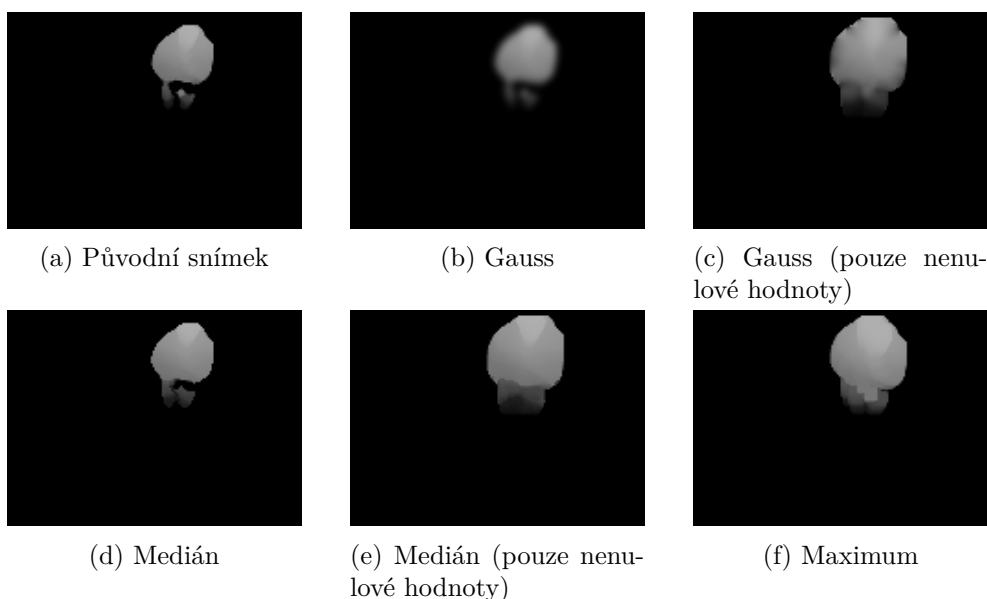
Mezi problematické chyby patří i „díry“ v hloubkové mapě. Brazey [7] díry rozděljuje na dva druhy.

První druh děr jsou ty způsobené přílišnou blízkostí Kinectu a osoby. Takto vzniklé díry bývají velké, zpravidla zabírají podstatnou část hlavy. Z toho důvodu je třeba jejich odstranění věnovat velkou pozornost. V mém případě byl však Kinect umístěn dostatečně vysoko, takže k podobným chybám nedochází.

Druhým typem jsou menší díry. Tyto díry vznikají patrně přílišnou členitostí objektu – infračervený projektor není schopen promítnout značku na malou plochu, která je zastíněna vysokým objektem. Vznik takovéto díry je znázorněn na obr. 5.2. Nejčastěji se jedná o chybějící okolí hlavy, jak ukazuje obr. 5.3.

Řešením může být použití morfologického operátoru dilatace.

V případě mé práce je operátor dilatace spojen s rozostřením pomocí me-



Obrázek 5.4: Různé techniky rozostření

Různá rozostření (s poloměrem 9, v případě Gaussova rozostření $\sigma^2 = 2$)

diánu. Medián je počítán pouze z těch pixelů v okolí, které mají nenulovou hodnotu.

Podobným způsobem bylo upraven Gaussovo rozostření. Testoval jsem tedy pět rozostřovacích technik – Gaussovo rozostření, Gaussovo rozostření ignorující nulové pixely, rozostření pomocí mediánu, rozostření pomocí mediánu ignorující nuly a rozostření pomocí maxima.

V případě „rozostření“ pomocí maximální hodnoty nezáleží na tom, jestli ignorujeme nulové hodnoty nebo nikoliv – výsledek bude stejný.

5.2.3 Převod na 3D model

Původní hloubková mapa obsahuje zkraslení, které je možné potlačit vytvořením trojrozměrného modelu. Tvorbu tohoto modelu v ostatních zdrojích popisuje sekce 4.8.

Výsledný 3D model je reprezentován hloubkovou mapou (ukázka je na obr. 4.5). Je možné, že v 3D modelu je přítomno více bodů se souřadnicemi $[x_P, y_P]$, které se liší pouze svou výškou. V takovém případě je v bodě $[x_P, y_P]$ uložena nejvyšší ze všech hodnot.

Příkladem takovéto kolize je ruka člověka visící kolmo k zemi (jako na obr. 4.4b), která může být na snímku zachycená z boku celá.

Převodem na 3D model se tak ztrácejí některé informace o osobě. Proto v následujících sekcích používám oba vstupy. Není-li řečeno jinak, je zpracovávána původní hloubková mapa.

Jednou ze zvláštností mého problému je fakt, že ulička není snímána zcela kolmo. Všechny Kinecty jsou umístěny nad jedním z regálů. Aby byly schopné snímat celou uličku až k protějším regálům, byly o několik stupňů natočeny.

Z tohoto důvodu je nutné přemístit referenční bod na ose y . V normálním případě se všechny souřadnice na ose x i y porovnávaly se středem snímku. Nyní se referenční bod nachází téměř na jeho dolním okraji.

5.3 Detekce objektů

Během detekční fáze jsou na snímku nalezena všechna místa, kde by se mohly nacházet osoby.

Pokud osoba není detektorem objevena, je informace o ní navždy ztracena. Naopak pokud detektor vrátí několik míst, kde osoby nejsou, v dalších částech algoritmu je možné je odstranit. Více potenciálních osob ale zvyšuje výpočetní náročnost algoritmu a snižuje možnost použít v dalších fázích složitější metody.

Metoda detekce by měla být dostatečně robustní, aby detekovala osoby různé velikosti (dospělí i děti) v různém oblečení (klobouk, batoh, sukně) v různých pozicích (stojící, shýbající se) a shluky osob (např. dvojice držící se za ruku, osoby překrývající se na snímku).

V následujícím textu jsou popsány jednotlivé metody, které je možné k detekci použít. Ze všech testovaných metod jsem vybral potlačení nemaxim pro její robustnost, jednoduchost, nízké nároky na výpočet a dobré výsledky během testování. Tato metoda měla ze všech největší AUC (s výjimkou extended h-maxima, důvod nepoužití je v sekci 5.3.6).

5.3.1 Hill climbing

Hill climbing je algoritmus používaný pro hledání lokálních maxim.

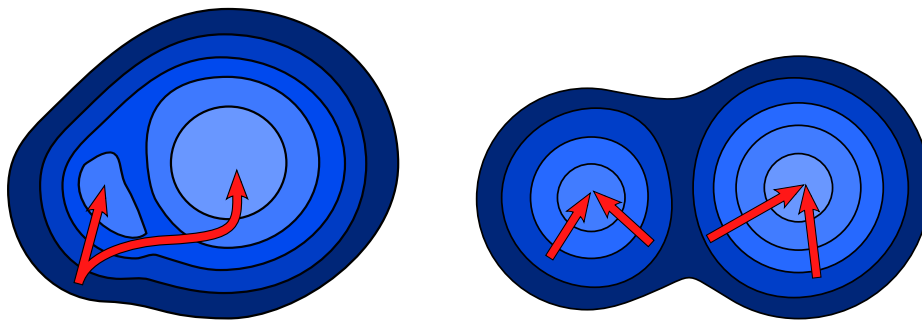
Ve své práci využívám hill climbing dvakrát. Na hill climbingu je založen detekční algoritmus popsáný v této sekci. Dále je použit pro úpravu výsledků vybraných detekčních algoritmů (viz str. 5.4).

Úprava výsledků používá jednoduchý hill climbing. Naopak hill climbing v detekčním algoritmu je výrazně upraven.

Jednoduchý hill climbing vybere v prostoru (detekční mapě) jeden bod $[x, y]$. V okolí tohoto bodu vybere jeden bod (ten nejvyšší) a pokračuje jím. Tento postup je opakován, dokud v okolí existuje nějaký vyšší bod, než $[x, y]$.

Prohledávání v mém algoritmu však probíhalo rozdílně. V každé iteraci není vybírán jeden (nejvyšší) bod, ale množina bodů, a to takto:

1. Je dán výchozí bod $[x, y]$.
2. Najdi všechny body na stejné „úrovni“ jako je zadaný bod. „Úroveň“ kompaktní množina pixelů, které mají s bodem $[x, y]$ stejnou hodnotu.



(a) Uvznutí v lokálním maximu o malé ploše (např. rameno) (b) Uvznutí v lokálním maximu o velké ploše (např. druhá, menší osoba)

Obrázek 5.5: Ukázka průběhu různých implementací hill climbingu. Červeně je znázorněný průběh jednoduchého hill climbingu, zeleně jeho rekurzivní implementace použitá v mé práci.

3. Výstupem této části algoritmu je seznam bodů, které sousedí s některým z bodů z „úrovně“ a zároveň mají vyšší hodnotu než tento jejich souseď.

Upravený hill climbing je rekurzivně spouštěn z každého bodu z této množiny. Výsledkem algoritmu je nejvyšší bod ze všech nalezených.

Důvodem k tomuto řešení je možnost, že algoritmus by jinak mohl uvíznout ve velice malém „hrbolku“, který by mohlo vytvořit například rameno. Jeden člověk by tak byl definován dvěma body (viz obr. 5.5a), případně by mohlo být detekováno pouze rameno.

Pokud se na snímku „slíjí“ dvě osoby do jednoho „blobu“, je uvíznutí v lokálním maximu naopak žádoucí (viz obr. 5.5b). K tomu dojde, pokud je prohledávací algoritmus spouštěn z bodů dostatečně blízko sebe (např. vzdálené od sebe nejvýše na minimální předpokládanou šířku lidské hlavy).

Výsledný detekční algoritmus postupně prohledává na každém d_{min} -tém řádku každý d_{min} -tý bod. Proměnná d_{min} značí minimální velikost hlavy.

Algoritmus má dva parametry – minimální velikost hlavy d_{min} a práh τ . Všechny potenciální osoby nižší než τ jsou ignorovány.

5.3.2 Potlačení nemaxim

Algoritmus potlačení nemaxim (non-maxima suppression) hledá na snímku lokální maxima. Na snímek je aplikován operátor potlačení nemaximálních hodnot (viz sekce 4.3). V bodech, které na snímku zůstanou, se nacházejí potenciální osoby.

Metoda vychází z toho, že každý člověk má v hloubkové mapě zpravidla jedno lokální maximum (stojí-li, je tímto maximem hlava). Od tohoto maxima se tvar člověka postupně svažuje.

Ve své práci jsem testoval dva způsoby implementace tohoto algoritmu.

Prvním způsobem je „naivní“ implementace:

Algoritmus 4: Naivní implementace potlačení nemaxim

vstup : hloubková mapa δ , poloměr skenovacího okénka r

výstup: množina M_{detect} obsahující nalezené body

```
1  $M_{detect} \leftarrow \emptyset$ 
2 for  $y \leftarrow 1$  to  $height$  do
3   for  $x \leftarrow 1$  to  $width$  do
4      $isMaximum \leftarrow true$ 
5     for  $i \leftarrow \max(1, y - r)$  to  $\min(height, y + r)$  do
6       for  $j \leftarrow \max(1, x - r)$  to  $\min(width, x + r)$  do
7         if  $\delta_{i,j} > \delta_{x,y}$  then
8            $isMaximum \leftarrow false$ 
9         end
10      end
11    end
12    if  $isMaximum$  then
13       $M \leftarrow M \cup \{(x, y)\}$ 
14    end
15  end
16 end
```

Druhým způsobem implementace je popsán v [24]. Snímek je rozdělen na čtvercové bloky s délkou strany $2r + 1$, následně je v každém čtverci nalezeno maximum a poté je ověřeno, zda je nalezený bod $[x, y]$ nejvyšší hodnotou ve čtvercové oblasti s délkou strany $2r + 1$ a se středem v bodě $[x, y]$.

Obě implementace vrací velmi podobné výsledky. Chování se liší pouze, pokud je lokálním maximem ve čtvercové oblasti více bodů se stejnou hodnotou. „Rozdělující“ algoritmus vybere jeden z nich, a podrobí jej zkoušce, zda je maximem ve svém okolí. Naivní zkouší všechny body. Pokud z nich je maximem ve svém okolí jen jeden bod, najde jej pouze naivní implementace.

Následující schéma ukazuje takový případ. Čísla reprezentují hodnoty v hloubkové mapě, svislé a vodorovné čáry naznačují hranice čtverců, v nichž jsou v „rozdělující“ implementaci hledána maxima. Čtverec znázorňuje oblast, kde „rozdělující“ algoritmus ověřuje, zda je bod lokálním maximem.

„Rozdělující“ algoritmus zkouší bod bB, protože žádný větší bod v čtverci není a našel jej jako první. Po kontrole jej zahodí, protože bod aA je větší. Naivní algoritmus však detekuje potenciální osobu na souřadnicích cC.

		a	b	c	d	e	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
A	⋯	4	2	2	1	1	⋯
B	⋯	3	2	2	1	0	⋯
C	⋯	2	2	2	0	0	⋯
D	⋯	2	1	1	0	0	⋯
	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Takto detekované body však pravděpodobně nejsou temenem hlavy. Proto implementace podle [24] není jen rychlejší, ale také má lepší úspěšnost detekce. Z toho důvodu pro detekci používám právě tento algoritmus.

Oba algoritmy mají dva parametry – velikost skenovacího okénka r a práh τ . Všechny hodnoty nižší než τ jsou ignorovány.

5.3.3 Narůstání oblastí

Metoda narůstání oblastí hledá lokální maxima. Podrobnější popis se nachází v sekci 4.1.

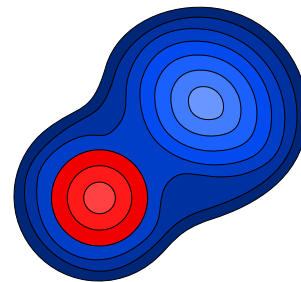
Na rozdíl od potlačení nemaxim je tato metoda nechybuje u velmi zvolna se svažujících křivek osob. Pokud se na snímku nachází čtverec stejných hodnot o velikosti $m \times m$, kde m je velikost skenovacího okénka, pak jej potlačení nemaxim identifikuje jako potenciální osobu. Nezáleží přitom na tom, jestli v okolí této plochy jsou pixely s vyšší hodnotou. Narůstání oblastí v takovém případě funguje správně a oblast zařadí do výsledku, pouze pokud je lokálním maximem.

V sekci 4.1 se zmiňuji o možnosti omezit hodnoty oblasti do zadaného intervalu. Takovéto řešení pro detekci osob používá [6]. Problémem tohoto řešení je fakt, že jedna osoba je tak detekována vícekrát – jednou hlava a několikrát zbylé části osoby. V [6] je provedena detekce hlavy pomocí metod strojového učení a ostatní části zahozeny.

V mé práci se však zabývám i snímky, kdy je člověk shrbený nebo v jiné netradiční pozici. Proto mnou testované narůstání oblastí nemá podobnou podmínku.

Kvůli tomu se ale takovýto algoritmus nehodí pro segmentaci a je možné jím osoby pouze detekovat. Při detekci dvou osob, které na snímku splývají, je totiž téměř celý snímek přiřazen vyšší osobě (viz obr. 5.6).

Algoritmus má jediný parametr, kterým je práh τ . Má-li nalezené globální maximum hodnotu menší nebo rovnu τ , je detekce osob ukončena.



Obrázek 5.6: Narůstání oblastí segmentující dvě splývající osoby

5.3.4 Water filling

Metoda Water filling na základě vstupního snímku vytvoří hloubkovou mapu, ve které hledá lokální maxima.

Metoda je představena v [8]. Podrobnější popis je možné najít v sekci 4.2.

Problém, který řeší [8] či jiní autoři využívající Water fillig je velice podobný problému rozebíranému v mé práci. Proto nebylo nutné v algoritmu dělat téměř žádné úpravy, jediným rozdílem tak mohou být odlišné konstanty.

Algoritmus má čtyři parametry:

1. počet „kapek“, které dopadnou na jedno místo
2. velikost okna pro hledání vyššího pixelu v okolí
3. hodnota, kterou musí pixel přesáhnout, aby na něj dopadaly „kapky“
4. hodnota, kterou musí bod ve výsledné funkci g přesáhnout, aby byl považován za potenciální osobu

5.3.5 K-means

Segmentační algoritmus K-means využívá [30] pro detekci osob v barevném záznamu. Snímek rozdělí na k oblastí, kde k je stanoveno jako minimální velikost hlavy d_{min} . Centroidy, které jsou navzájem bližší než d_{min} , jsou sloučeny. Každý centroid pak reprezentuje jednu osobu.

Ostatní zde jmenované detekční metody využívají spíše fakt, že člověk bývá ve svém okolí nejvyšším objektem. K-means na výšku žádným způsobem nehledí a snaží se spíše hledat větší souvislé plochy.

V barevném snímku, ve kterém [30] osoby detekuje, je možné hlavu člověka oddělit pomocí barvy. Právě tato skutečnost stojí za dobrou použitelností k-means pro detekci osob. V hloubkové mapě tato skutečnost neplatí a algoritmus je nutné upravit.

Úprava probíhá následovně: v hloubkové mapy jsou odstraněny všechny body s hodnotou nižší nebo rovnou předem stanovenému prahu τ . Zbylé body pak slouží jako vstup pro k-means. Takovýto algoritmus však v sobě skrývá mnoho problémů.

Může se stát, že shrbenou postavu, která zabírá na snímku více místa, identifikuje K-means dvakrát. Změnou parametrů je sice možné K-means přizpůsobit, aby v tomto případě fungoval správně, v takovém případě ale hrozí riziko, že nenajde stojící postavu. Proto jsem krom klasického k-means testoval i weighted k-means.

Weighted k-means je obdobou k-means, obsahuje však jednu modifikaci. Každý bod ve weighted k-means má váhu w . Umístění centroidů v každé iteraci algoritmu není vypočteno jako průměr poloh všech bodů, které jsou k centroidu přiřazeny, ale jako jejich vážený průměr.

Váha jednoho bodu w_{xy} je v mé práci stanovena jako výška δ_{xy} , ke které je přičtena konstanta w_{const} . Výsledek algoritmu tak bude více konvergovat k vyšším bodům. Body s hodnotou nižší než τ mají váhu nulovou. Díky w_{const} je možné regulovat, jak moc velký význam hraje výška v přepočtu umístění centroidů.

Obě metody fungovaly s účinností srovnatelnou s většinou ostatních metod. Ukázalo se, že úprava k-means nepřinesla žádoucí zlepšení detekce.

Detekční algoritmus založený na k-means má tři parametry: práh τ , minimální velikost hlavy d_{min} a maximální možný počet iterací k-means.

Algoritmus založený na weighted k-means má parametry čtyři: tři výše uvedené a w_{const} .

5.3.6 Extended h-maxima

Algoritmus extended h-maxima hledá lokální maxima. Algoritmus je popsán v sekci 4.4.

Detekční algoritmus založený na h-maxima nejdříve hledá lokální maxima, pak určité z nich vyřazuje. Máme-li bod A s hodnotou δ_A , který je lokálním maximem, nesmí od něj existovat cesta k dalšímu lokálnímu maximu B bez toho, aby tato cesta procházela nějakým jiným bodem o hodnotě menší (nebo rovné) $\delta_A - h$. Body, které tuto podmínku splňují, jsou prohlášeny za potenciální osoby.

Tímto algoritmem je inspirován příznak h , který značí maximální možnou konstantu h , se kterou by bod byl detekován pomocí extended h-maxima (viz sekce 5.6).

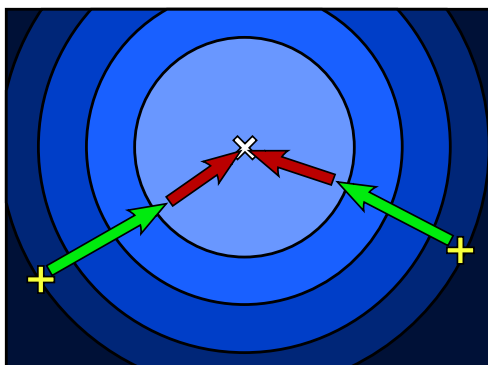
Extended h-maxima se dá nahradit pomocí algoritmu potlačení nemaxim (naivní implementace s velikostí skenovacího okénka 1) a následným odstraněním všech osob s příznakem h menším než stanovená konstanta. Obě metody vrací zcela stejný výsledek.

Příznak h se navíc dá zkombinovat s ostatními příznaky. Proto jsem algoritmus pro detekci nepoužil, i když detekoval potenciální osoby velice dobře.

5.4 Úprava výsledku detekce

Detekční algoritmy mohou vracet více bodů, které se nacházejí velice blízko sebe a označují tutéž osobu. Aby tyto body nebyly dále považovány za několik různých osob, je nutné je sloučit. K tomu jsem použil následující dva algoritmy.

- **Odstranění nízkých potenciálních osob** – Je velice nepravděpodobné, že by objekt vysoký několik centimetrů byl osobou. Proto jsou nízké objekty zahazovány. Zahazení nízkých objektů je v mém návrhu součástí již samotných detekčních algoritmů. U algoritmů narůstání oblastí zrychlí průběh, protože není procházet nízko položené oblasti. Algo-



Obrázek 5.7: Úprava výsledků detekce. Dva detekované objekty (křížky ve tvaru +) jsou posunuty do lokálního maxima (nejsvětlejší kruh) pomocí hill climbingu. Následně jsou oba umístěny do středu „úrovně“, díky čemuž jsou sloučeny do jednoho bodu (křížek ve tvaru ×).

ritmus založený na k-means dokonce potřebuje hranici τ , která převede hloubkovou mapu na binární obrázek.

- **Odstranění potenciálních osob s malým příznakem „h“** – Příznak „h“ je popsán dále v sekci Příznaky (5.6) a používán pro klasifikaci. Klasifikační algoritmus jej dokáže využít lépe, než prostá eliminace všech objektů, které mají „h“ menší než stanovená hranice. Zpracovává však segmentované snímky a segmentační algoritmus vyžaduje, aby žádná osoba nebyla detekována vícekrát.

Dojde-li k detekci jedné osoby vícekrát, má nejvýše detekovaný bod zpravidla velkou hodnotu příznaku „h“, zatímco ostatní body mají tuto hodnotu velmi malou.

Odstraněním bodů s nízkou hodnotou „h“ jsou tedy potlačeny duplicitní detekce, zatímco osoby a jiné detekované objekty zůstanou.

- **Hill climbing** – Jedná se o klasický horolezecký algoritmus. Vstupem pro hill climbing je rozostřený snímek, protože hill climbing je náchylný na šum. Všechny body se tak posunou do jednoho místa, případně alespoň velmi blízko sebe. To se však stává pouze v případě, který je řešen dále.
- **Umístění do středu „úrovně“** – Úrovní je myšlen kompaktní shluk pixelů, které mají stejnou hodnotu. Důvodem k této operaci je fakt, že hill climbing nemusí vždy posunout pixely do jediného místa, ale může je zanechat blízko sebe, pokud je lokální maximum větší než jen jeden pixel.

Různé úpravy výsledků detekce je dobré kombinovat s různými detekčními algoritmy. Například algoritmy hledající lokální maxima je zbytečné kombinovat s hill climbingem, který jejich výsledky nijak neupraví.

Na zvolený detektor potlačení nemaxim jsem aplikoval odstranění nízkých potenciálních osob, umístění do středů úrovní a odstranění bodů s nízkou hodnotou „h“.

5.5 Segmentace

Segmentační algoritmus má za úkol rozdělit vstupní snímek na několik částí, z nichž každá zobrazuje jednu osobu.

Tato sekce popisuje testované segmentační algoritmy. Vybraným algoritmem pro rozdělení snímek mezi osoby byla primitivní segmentace pro svou jednoduchost. Protože primitivní segmentace nezvládá rozdělit překrývající se osoby, je doplněna segmentačním algoritmem watershed.

Kromě toho používám segmentaci „selekce hlavy“. Díky tomu je možné vytvořit příznaky popisující nejen celou osobu, ale i samotnou hlavu.

5.5.1 Primitivní segmentace

Primitivní segmentace od sebe oddělí osoby stojící daleko od sebe. Složitější případy, kdy se osoby překrývají, primitivní segmentace neřeší. Takovéto snímky jsou segmentovány pomocí watershed.

Průběh algoritmu je následující:

1. Ze vstupního snímku je vyroben binární obrázek, kde pixely s hodnotou větší než $\tau = 50$ mají hodnotu `true`, zatímco ostatní `false`.
2. Na binární snímek je aplikován morfologický operátor dilatace. Důvodem k tomu je, aby osoby s oddělenou hlavou (jako na obr. 5.3b) byly na snímku celé.
3. Binární snímek je rozdělen na oblasti dále zvané jako „bloby“. Blob je kompaktní oblast pixelů s hodnotou `true`,
4. Každému blobu je přiřazen obrázek δ_{seg} . Tento obrázek je podmnožina původního snímku, která obsahuje pouze pixely přítomné v blobu.
5. Každé osobě je přiřazen „blob“, který se nachází na stejném místě.
6. Pokud je blob přiřazen pouze jedné osobě, tak je této osobě přiřazen obrázek δ_{seg} . Pokud je přiřazen více než jedné osobě, je na obrázek δ_{seg} aplikován algoritmus watershed.

Kdyby na snímek byl aplikován pouze watershed, mělo by to dvě nevýhody. První je rychlost výpočtu. Druhou je fakt, že samotný watershed přidělí k

jedné z osob i další nedetekované objekty, které se na snímku nacházejí (např. nákupní vozík). Primitivní segmentace je nepřirazuje k žádné osobě, pokud k ní nejsou příliš blízko.

5.5.2 Selekcce hlavy

Pokud není nutné, aby rozdělené snímky obsahovaly celou postavu, dá se použít jednoduchý algoritmus na vyříznutí hlavy. Jeho výsledkem je kompaktní shluk pixelů, které mají hodnotu alespoň $d > d_{max}\tau$ (případně $d > d_{max} - \tau$), přičemž d_{max} je hodnota nejvyššího bodu postavy a τ pevně stanovená konstanta⁹.

Všechny ostatní pixely jsou zahozeny.

5.5.3 Rozlití osoby

Tento algoritmus rekurzivně hledá v sousedství pixelu místa nižší nebo se stejnou hodnotou, než je aktuální pixel. Všechna místa, která projde přiřadí dané osobě.

V případě, že dojde ke konfliktu (jeden pixel náleží více osobám), je nutné pixely přiřazovat podle vzdálenosti.

Vstupem pro „rozlití osoby“ nutně musí být dobře rozostřený snímek, jinak jej výrazně ovlivní šum. Body, které jsou kvůli šumu lokálními maximy, nejsou přiřazeny žádné osobě. I přesto však tento způsob segmentace ignoroval malé „hrbolky“, které ve snímku zůstaly i po rozostření. Proto byl místo něj použit lépe fungující watershed.

5.5.4 Watershed

Funkce algoritmu je popsána v sekci 4.9. Počátečními body pro Watershed jsou detekované potenciální osoby. Algoritmus je nutné uzpůsobit tak, aby postupoval od vyšších pixelů k nižším.

5.6 Příznaky

Příznaky jsou číselné či vektorové hodnoty popisující (potenciální) osobu. Ve své práci je používám jako vstup pro klasifikaci, zároveň slouží pro správné přiřazování osob během sledování.

Testoval jsem následující příznaky:

1. HOD – Histogram of oriented depths [42],
2. plocha – počet pixelů s hodnotou větší než τ ,

⁹[40] považuje za hlavu všechny pixely vyšší než $0.9 \cdot d_{max}$

3. h – Tento příznak je inspirovaný algoritmem extended h-maxima (viz sekce 4.4). Hodnota příznaku „ h “ je stanovena jako nejvyšší možný parametr h , který je možné předat algoritmu extended h-maxima, aby danou osobu našel.

Extended h-maxima používá [4] pro detekci potenciálních osob. Ve své práci však takovýto detekční algoritmus hlouběji netestují.

Všechny body detekované tímto algoritmem jsou lokální maxima. Je tedy nepravděpodobné, že by našel výrazně více potenciálních osob, než jiné algoritmy založené na detekci lokálních maxim (např. potlačení nemaxim, water filling apod.). Mohlo by se stát, že extended h-maxima najde body ležící blízko u sebe, které jsou odděleny velkým „příkopem“. V takovém případě potlačení nemaxim najde pouze jeden, nejvyšší, z nich.

Výhodou extended h-maxima je naopak schopnost rozlišit ty detekované objekty, které lidmi nejsou. Tato schopnost se však dá lépe využít převedením detektoru na příznak.

4. průměr – maximální vzdálenost od prvního pixelu s hodnotou větší než τ k poslednímu takovému pixelu. Průměr byl počítán ve čtyřech směrech – horizontálním, vertikálním a dvěma směry napříč. Ze všech „průměrů“ byl spočten jejich průměr a rozptýl.
5. kulatost – poměr plochy a kruhu, který má stejný poloměr jako detekovaný objekt
6. „G-kulatost“ – kulatost popsaná v [6] (viz str. 10). Pro rozlišení ode mnou definované kulatosti obsahuje prefix „G“ podle jmen autorů článku.
7. výška osoby nebo nejvyššího bodu v shluku
8. Výstup různých metod detekce – vektor příznaků. Každý snímek byl vyhodnocen všemi detekčními algoritmy popsány v sekci 5.3. Následně proběhlo spárování osob nalezených použitým detekčním algoritmem (potlačení nemaxim) a ostatními algoritmy. Spárování probíhá podle algoritmu 5.

Algoritmus 5: Propojení osob detekovaných různými algoritmy

```

1  $A \leftarrow$  potenciální osoby detekované potlačením nemaxim
2  $B \leftarrow$  potenciální osoby detekované druhým algoritmem
3 while  $A \neq \emptyset$  and  $B \neq \emptyset$  do
4    $a, b \leftarrow a \in A, b \in B$ , s nejmenší vzájemnou vzdáleností
5   if  $|a, b| \geq 20$  then
6     break
7   end
8   propoj( $a, b$ )
9   odeber  $a$  z  $A$ 
10  odeber  $b$  z  $B$ 
11 end

```

Pokud daný detekční algoritmus osobu našel, má příznak hodnotu 1, v opačném případě 0.

Důvodem k použití tohoto příznaku je fakt, že detekční algoritmy chybují v různých situacích. Například problémem pro k-means je přílišná velikost osoby, problémem pro algoritmy založené na hledání lokálních maxim pak přílišná členitost povrchu osoby. Informace o tom, které algoritmy osobu našly tak poskytují informaci o vzhledu osoby na snímku.

9. poloha na ose x a y – užitečná v kombinaci s ostatními příznaky. Například plocha osoby na okraji snímku bývá jiná než plocha osoby uprostřed.

5.7 Klasifikace objektů

Klasifikátor má za úkol rozdělit detekované objekty na osoby a jiné objekty.

Vstupem pro něj jsou souřadnice detekovaného objektu a obrázek, který jej zobrazuje (resp. jeho popis pomocí příznaků).

Výstupem klasifikátoru je hodnota osoba/omyl přiřazená každému objektu. Objekty označené jako omyl nejsou v této fázi vyřazeny, protože je možné, že objekt byl takto klasifikován omylem. Definitivní rozhodnutí tak učiní až sledovací fáze, která využívá informaci z předchozích snímků (viz sekce 5.8.7).

Klasifikace probíhá pomocí metod strojového učení s učitelem (podrobnější popis v sekci 5.7.2). Ty jsou natrénovány 3218 snímků obsahujícími 1–4 osoby. Snímky byly vybrány ze záznamu z jedné kamery v prodejně v průběhu jednoho dne.

5.7.1 Způsob klasifikace

Zvažoval jsem dva způsoby klasifikace, které se liší vstupním snímkem.

První z nich nejdříve sloučí stejné osoby a segmentuje obrázek. Poté segmentovaný obrázek popíše pomocí příznaků a následně ohodnotí pomocí binárního klasifikátoru. Tento způsob jsem vybral a implementoval.

Druhým způsobem klasifikace je použití složitějšího klasifikátoru, který umí zpracovat nesegmentovaný snímek obsahující více osob. Tento obrázek rozděluje do tříd podle toho, kolik osob na snímku je.

Složitější klasifikátor má mnoho výhod: neklade velké nároky na detekční algoritmus a nepotřebuje slučování osob a segmentaci. Vstupem pro takový klasifikátor mohou být jednotlivé „bloby“ (oblasti segmentované „primitivní segmentací“, viz sekce 5.5.1).

Přesto jsem tento způsob zavrhl pro jeho nedostatečnou robustnost.

Tento způsob klasifikace by mohl dosahovat velké úspěšnosti na datech obsahující jednu či dvě osoby. Klasifikátor může být dobře natrénován pro klasifikaci dvou stojících osob, osoby a dítěte, sehnuté osoby a kočárku apod.

V takovém případě je výsledný algoritmus úspěšnější, protože není ovlivněn možnými chybami segmentace.

U velkých shluků (např. pět osob) však počet různých tvarů „blobu“ stoupá, protože každá osoba se může nacházet v jiné poloze.

Pro vyšší počty osob je navíc problematické najít dostatek trénovacích dat. Jak je již zmíněno výše, náhodně zvolená trénovací data obsahují maximálně 4 osoby (a to jen tři takové snímky). Najít dostatek snímků v různých polohách např. pro 20 osob (protože i taková situace může nastat), je téměř nemožné. Takové snímky by tedy bylo nutné uměle vytvořit.

Binární klasifikátor oproti tomu může správně fungovat i v počtu osob, na které nebyl natrénován.

Vylepšením binárního klasifikátoru je pak klasifikace potenciální osoby do více tříd. Tento způsob využívám ve sledovací fázi, kde je použit velmi jednoduchý (manuálně vytvořený) rozhodovací strom. Ve své práci tak používám dva klasifikátory. Důvody rozdělení jsou popsány spolu s vysvětlením klasifikátoru v sekci 5.8.7.

Možným vylepšením binárního klasifikátoru, které jsem nezkoušel, je klasifikace do více tříd podle toho, jak je pravděpodobné, že je detekovaný objekt člověkem. Další možností je místo klasifikace použít regresi. Pravděpodobnost bylo možné využít ve sledovací fázi. (Filtr popsáný v sekci 5.8.7 by této vlastnosti mohl využít. Problémem by však bylo, že děti a netradičně vypadající osoby by měly dlouhodobě nízkou „pravděpodobnost existence“, která je popsána ve zmíněné sekci.)

5.7.2 Metody učení

Oba druhy klasifikace jsem používal s následujícími metodami učení: rozhodovací strom, náhodný les, support vector machines (SVM), boosting a k -nn (k experimentálně stanoveno jako 5). K -nn jsem testoval hlavně kvůli porovnání výsledků s ostatními metodami. Pro praktické využití v mém algoritmu se nehodí kvůli výpočetní náročnosti.

Implementace všech metod byla převzata z knihovny OpenCV¹⁰.

Všechny metody jsou použitelné pro klasifikaci, každá dokáže lépe zpracovávat jiné příznaky.

Nejúspěšnější metodou se ukázal být boosting, konkrétně diskretní AdaBoost.

Tabulky výsledků jsou uvedeny v kapitole Testování v sekci 6.6.

5.8 Sledování

Během fáze sledování je nutno najít stejné osoby na předchozím a aktuálním snímku. Díky tomu je možné sledovat chování jednotlivých osob.

¹⁰<http://opencv.org>

Sledování má také za úkol odstranit chyby vzniklé v předchozích fázích, k čemuž využívá informace z předchozích snímků.

Tracker (objekt zajišťující sledování) sleduje a uchovává v paměti všechny objekty, které detektor zachytil. Avšak pouze ty, které byly klasifikovány jako osoby, se objeví na výstupu algoritmu.

5.8.1 Přiřazení osob

Sledování jsem pojal jako řešení problému přiřazení (assignment problem), které řeším pomocí maďarského algoritmu [52] (viz sekce 4.5 a 4.6).

Výhodou tohoto pojetí je odolnost vůči chybám v předchozích částech algoritmu a flexibilita (viz dále).

Vstupem pro problém přiřazení je čtvercová tabulka (matice). Hodnotami tabulky jsou vzdálenosti mezi osobami detekovanými na novém snímku a mezi osobami uloženými v trackeru, vzdálenost osoby k okraji snímku a „error distance“ – číslo indikující, jak moc je pravděpodobné, že osoba byla detekována omylem. Buňky se tedy rozdělují na tři typy. Jejich popis a důvod jejich zavedení jsou popsány dále v textu.

Jednoduchý tracker se dá pouze vytvořit za pomoci vzdáleností mezi osobami samotnými – tedy jen jedné části mnou používané tabulky. Vzdálenost používám čtvercovou eukleidovskou (čtvercová pouze z výpočetních důvodů). Kromě toho se ke vzdálenosti připočítává i rozdíl mezi tvarem osob (více v sekci 5.8.4).

Tabulka využívající pouze vzdálenosti může vypadat například takto:

Tabulka 5.1: Tabulka přiřazení osob v trackeru. Nahoře jsou osoby detekované na předchozím snímku, vlevo osoby detekované na aktuálním snímku.

akt.\předch.	Alice	Bob
Alice	5 cm	50 cm
Bob	60 cm	18 cm

Takto definovaná matice dobře řeší problém sledování několika osob, dokud do snímaného prostoru nevejde nová osoba nebo někdo neodejde. Poté není možné zkonstruovat čtvercovou matici.

Bylo by možné přidat do matice několik nulových vektorů, aby byla opět čtvercová. Ta osoba, která je přiřazena takto vloženému sloupci resp. řádku, prostor pravděpodobně opustila nebo do něj vstoupila.

Alternativně je místo nulového vektoru možné přiřadit vektor vzdáleností osob k nejbližšímu okraji. Toto řešení má tu výhodu, že jako vstupující/vystupující spíše označí tu osobu, která je k východu blíže.

V případě, že prostor jedna osoba opustila a další do něj v ten samý okamžik vstoupila by takto zkonstruovaná metoda opět selhala. Proto jsem se rozhodl do matice vložit m řádků a n sloupců (m je počet sloupců matice a

Tabulka 5.2: Tabulka přiřazení v případě, že jedna osoba prostor opustila

akt.\předch.	Alice	Bob
Alice	5 cm	50 cm
vzdálenost k okraji	60 cm	18 cm

n počet jejích řádků). Opět platí, že je-li osoba přiřazena vloženému řádku či sloupci, znamená to, že do snímaného prostoru vstoupila či jej opustila. Vzniklá matice je opět čtvercová.

Takovýto algoritmus funguje správně i v případě, prostor zároveň opustily všechny osoby a stejný počet do něj vstoupil.

Tabulka 5.3: Tabulka přiřazení v případě, že jedna osoba prostor opustila a druhá do něj vstoupila

akt.\předch.	Alice	Bob	nová osoba	nová osoba
Alice	5 cm	50 cm	70 cm	70 cm
Cyril	80 cm	100 cm	5 cm	5 cm
konec	70 cm	10 cm	0	0
konec	70 cm	10 cm	0	0

Pomocí assignment problému je možné řešit i situaci, kdy je osoba v klasifikační fázi v několika snímcích špatně označena, není nalezena, nebo je naopak chybně detekován jiný objekt.

Řešením těchto problémů je zavedení proměnné, kterou v tomto textu pojmenovávám jako „error distance“. Do matice přiřazení je přidáno m řádků a sloupců, které obsahují právě „error distance“. (Číslo m udává počet detekovaných osob na předchozím či aktuálním snímku, podle toho, co je větší. Je totiž nutné, aby matice zůstala čtvercová.)

Přiřadí-li maďarský algoritmus osobu k takto přidanému řádku nebo sloupci, znamená to, že osoba nemá odpovídající osobu na předchozím/aktuálním snímku.

V případě nově detekované osoby to znamená, že byla pravděpodobně detekována omylem. Je-li tomuto vektoru přiřazena sledovaná osoba, znamená to, že detektor (pravděpodobně omylem) osobu nedetekoval.

Hodnota „error distance“ se počítá pomocí výstupu klasifikátoru a historie objektu, detailní popis je v sekci 5.8.5.

Výsledná matice přiřazení má tedy tři druhy sloupců a řádků, jak ukazuje tabulka 5.4.

Výstup maďarského algoritmu je přiřazení, které může nabývat následujících hodnot:

- osoba/objekt \rightarrow osoba/objekt – Daná osoba z předchozího snímku je přiřazenou osobou na následujícím snímku. Takto detekovaní lidé jsou

výstupem celého algoritmu.

- osoba/objekt → výstup – Daná osoba vystoupila ze sledovaného prostoru.
- vstup → osoba-objekt – Daná osoba vstoupila do sledovaného prostoru.
- osoba/objekt → ne-osoba – Objekt z předchozího snímku není člověkem nebo jej detektor nezachytil
- ne-osoba → osoba/objekt – Objekt z aktuálního snímku byl detekován omylem
- ostatní přiřazení – žádný význam

Tabulka 5.4: Význam buněk v matici přiřazení

	Osoby/jiné	Vstup do prostoru	Chyba
Osoby (a jiné objekty)	vzdálenost mezi sebou + rozdíl mezi tvarem	vzdálenost k okraji	error distance
Opuštění prostoru	vzdálenost k okraji	0	∞
Chyba	error distance	∞	0

5.8.2 Použití 3D modelu

Použitím 3D modelu (obecný popis v sekci 4.8, popis implementace v 5.2.3) zlepšuje účinnost algoritmu, protože při rovnoměrném pohybu nedochází k velkým výkyvům vzdáleností mezi osobami a rychlosti osob.

Jestliže k převodu nedojde, hlava osoby se bude kvůli „efektu rybího oka“ pohybovat u okrajů snímku rychleji než uprostřed.

Nevýhodou může být problematické zjišťování vzdálenosti od okraje, protože okraj takto vzniklého modelu bude lichoběžníkový, zatímco původní snímek měl čtvercový tvar. Proto k zjišťování vzdálenosti od okraje používám souřadnice v původním snímku.

5.8.3 Kalmanův filtr

Správné funkčnosti trasovacího algoritmu pomáhá Kalmanův filtr, který eliminuje šum a zároveň do algoritmu přidává informaci o pohybu. Využitím této informace je možné lépe sledovat osoby jdoucí těsně vedle sebe, nebo se těsně míjejí.

Kalmanův filtr zpracovává čtyři skalární veličiny – polohu osoby na ose x , polohu na ose y a rychlost¹¹ na ose x a y .

Ve své práci používám dva Kalmanova filtry. První z nich slouží k odstranění šumu souřadnic osoby v 3D prostoru. Tyto souřadnice jsou zásadní pro vzájemné přiřazování osob. Druhý vypočítává souřadnice na snímku. Poloha osoby na snímku slouží jako výstup celého algoritmu, proto je nutné, aby se na výsledné poloze osoby neprojevovaly krátkodobé odchylky kamery a detektoru.

5.8.4 Využití příznaků

K přiřazení jednotlivých osob používá můj algoritmus nejen informaci o poloze. Jednotlivé osoby na snímku je možné rozlišit i pomocí jejich fyzických vlastností.

Konkrétně jsem použil výšku a velikost hlavy, které se v průběhu času příliš nemění.

5.8.5 Pravděpodobnost existence

Každé sledované osobě je možné přiřadit pravděpodobnost existence p . Podobnou hodnotu má přiřazena nově detekovaná osoba. Výpočet pravděpodobnosti obou druhů osob je popsán dále.

Pravděpodobnost existence má více využití. Rozhoduje, zda sledovaný objekt je skutečně osobou, nebo nikoliv. Je-li pravděpodobnost existence menší než konstanta τ_{ignore} , osoba se neobjevuje na výstupu. To znamená, že při dotazu na nově detekované osoby tracker předstírá, že osoba neexistuje. Pokud její pravděpodobnost existence překročí τ_{ignore} , opět se k ní vrátí.

Po vynásobení p konstantou c_{prob} vznikne „error distance“, kterou používá maďarský algoritmus pro správné přiřazování osob. Vynásobení konstantou je nutné k tomu, aby pravděpodobnost mohla být porovnávána se vzdálenostmi.

Konstanty τ_{ignore} a c_{prob} jsem experimentálně stanovil jako 0,55 a 500.

Pravděpodobnost existence se počítá jako $1 - p_{err}$, kde p_{err} je pravděpodobnost chyby. Pravděpodobnost chyby je spočtena ze čtyř jejích složek, které jsou popsány dále. Je-li p_i jedna ze složek a $c_i \in \langle 0; 1 \rangle$ váha přiřazená této složce, pak

$$p_{err} = \prod_i p_i c_i + (1 - c_i).$$

Všechny složky jsou čísla v intervalu $\langle 0; 1 \rangle$.

¹¹ změna vzdálenosti osob mezi dvěma sousedními snímky

„Sledovaná osoba“ uložená v trackeru má následující složky p_{err} :

- **doba trvání** – p_{dur} – osoba, která se v prostoru vyskytuje dlouho má vyšší pravděpodobnost existence než ta, která byla zachycena na jednom nebo dvou snímcích. Vyskytuje-li se osoba na snímku pouze krátce, může se jednat o chybu měření.

Doba trvání je tvořena stromem. Hodnota p_{dur} je:

- 1, pokud se osoba objevila pouze na aktuálním
- 0,6, pokud se osoba objevila na druhém snímku
- 0,3, pokud se osoba objevila na 3 snímcích
- 0,1, pokud se osoba objevila na 4 snímcích
- 0,05, pokud se osoba objevila na 5 snímcích
- 0,025, pokud se osoba objevila na 6 snímcích
- 0, pokud se v prostoru nachází déle

- **nepřítomnost** – p_{ex} – osoba, kterou často detektor nezachytí, je spíše chyba měření než osoba naměřena často. Je spočtena jako poměr počtu snímků, na kterých byla osoba zachycena, vůči počtu všech snímků od prvního výskytu osoby.

- **nepřítomnost na posledních snímcích** – p_{lex} – Osoba, která na posledních snímcích nebyla detekována, má nižší pravděpodobnost existence než osoby, které detekovány byly. Jedná se především o osoby, které z prostoru již odešly, případně osoby, které se z prostoru „ztratily“ (například dítě schované za vyšší osobou).

Kdyby tato složka nebyla součástí pravděpodobnosti existence (a tím pádem i „error distance“ používané v přiřazovacím algoritmu), pak by tracker přiřazoval nově detekovanou osobu té osobě, která je nové detekci nejbližší. Tak by se v prostoru mohla znovu objevit osoba, která z něj již zmizela, avšak kvůli dlouhé době přítomnosti v prostoru nemá nízkou složku „přítomnost“. Skutečný obraz osoby nacházející se jen o něco dále by pak z prostoru zmizel.

Je-li n počet snímků, pak je p_{lex} definována jako:

$$p_{lex} = 1 - \frac{2}{1 + e^{-n}}$$

- **klasifikace a stav** – p_c – využívá data z klasifikátoru. Aby bylo možné data z klasifikátoru lépe využívat v čase, je používán klasifikační filtr (viz sekce 5.8.7). Tento filtr vrací pravděpodobnost, že sledovaný objekt je skutečně osobou. Tato pravděpodobnost je čtvrtou složkou pravděpodobnosti existence.

Váhy pro jednotlivé složky jsem pomocí testů stanovil jako:

- doba trvání – 0,6
- nepřítomnost – 0,7
- nepřítomnost na posledních snímcích – 0,5

Pokud je osoba zastíněna jinou, pak se zvyšuje na 0,6. Osoba je zastíněna, když se na snímku nachází osoba s výrazně vyšší výškou – o 30 odstínů šedi – v dostatečně malé vzdálenosti – 30 px. Stínící osoba musí mít větší polohu na ose y – tedy být blíže Kinectu. V takovém případě je žádoucí, aby maďarský algoritmus přiřadil novou detekci spíše oné velké osobě, než malé, která nejspíše není vidět.

- klasifikace – 0,9

Nově detekovaná osoba pak má pravděpodobnost chyby slouženou ze svou složek:

- **výsledek klasifikace** – výstup klasifikátoru (viz sekce 5.7), který je převeden na číselnou hodnotu 1 (osoba) nebo 0 (jiný objekt)
- **zaldněnost okolí** – hodnota spočtená pomocí vzdálenosti k nejbližší osobě. Čím blíže jsou si detekované osoby, tím spíše je jedna z nich detekována chybně. Snížením pravděpodobnosti existence se sníží pravděpodobnost existence obou, následně je jedna z nich přiřazena sledované osobě a druhá pro nízkou p zahozena.

Motivací k zavedení p_{err} byla penalizace dvakrát detekované jedné osoby. Experimenty ukázaly, že zaldněnost okolí má pouze malý vliv na výsledek algoritmu.

Váha pro výsledek klasifikace je 0,9. Zaldněnost okolí má váhu 0,6.

5.8.6 Pravděpodobnost odchodu

Pravděpodobnost odchodu udává, jak moc je pravděpodobné, že osoba se na snímku již nikdy neobjeví.

Překročí-li stanovenou mez, přestane ji tracker uchovávat v paměti a není možné jí přiřadit nově detekované osobě.

Počítá se z doby nepřítomnosti na snímku a vzdálenosti k okraji. Je-li osoba zastíněna jinou, pravděpodobnost odchodu se snižuje.

5.8.7 Klasifikace a stav

Finální rozhodnutí, zda potenciální osobu ponechat nebo zahodit, neprobíhá v klasifikační fázi (sekce 5.7), ale až během sledování. Je totiž možné, že klasifikátor občas udělá chybu.

Tracker obsahuje svůj vlastní jednoduchý klasifikátor, který rozděluje sledované objekty na osoby a ostatní objekty. K tomuto rozhodování využívá především výstup z klasifikační fáze.

Výstupem tohoto klasifikátoru je hodnota 0/1, která se využívá k výpočtu pravděpodobnosti chyby p_{err} (složka klasifikace a stav). Je-li objekt klasifikován na většině snímků jako osoba, pak je výstupem 1. V opačném případě vrací klasifikátor nulu.

Kromě toho také rozděluje osoby do dvou stavů:

1. stojící či jsoucí – výchozí stav,
2. shrbená – osoba, která se pro něco sklání,
3. na okraji – osoba vcházející do prostoru nebo odcházející.

Shrbená postava může být díky své výšce a zabrané ploše zaměněna například za nákupní vozík. K jejich rozpoznání využívám informaci o pohybu. Shrbená osoba zůstává na místě, zatímco vozík se často pohybuje.

Všechny objekty klasifikované jako osoby s výškou nižší než 85 (z 256 odstínů šedi) jsou vyhodnoceny pomocí manuálně vytvořeného rozhodovacího stromu:

- Osoba je označena jako shrbená, pokud se nehýbe, na většině předchozích snímků byla klasifikována jako osoba, zabírá plochu alespoň 600 čtverečních pixelů a má dostatečně velký příznak rozptyl průměru. Nehýbat se v tomto případě znamená, že se osoba za posledních 5 snímků posunula maximálně o 100 px (čtvercová eukleidovská vzdálenost).
- Osoba je označena jako dítě (stojící osoba), pokud jeho hlava zabírá plochu mezi 30 px² a 50 px². Velikost celého dítěte musí být mezi 100 a 500 px².
- Zbylé nízké potenciální osoby jsou označeny jako chyba detekce, i když je klasifikátor označil jako osoby.

Všechny uvedené parametry byly vytvořeny experimentálně. Parametry pro dítě jsou vytvořeny tak, aby jimi prošlo co nejméně objektů – a to i za cenu toho, že bude vyřazeno skutečné dítě. Malé děti totiž zpravidla nenakupují, takže není nutné jejich pohyb analyzovat a jejich vynechání není tak závažný problém, jako chybná detekce jiného objektu.

Osoba se nachází na okraji, pokud je její poloha na ose x menší než 45 px nebo větší než 125 px (ze 160). Tyto osoby často nedokáže Kinect zaznamenat celé, takže na snímku například není vidět horní polovina postavy.

U osob na okraji jsem změnil výpočet vzdálenosti – je vydělena dvěma a nezapočítávají se do ní příznaky. Kvůli nedokonalosti záznamu na okraji snímku je totiž velmi pravděpodobné, že příznaky i poloha osoby se budou v budoucnu měnit – a to třeba i velmi výrazně.

Podobnou úpravu jsem zkoušel na stavu stojící/shrbená. Velkým problémem shrbených osob totiž je přesun lokálního maxima z hlavy na záda. To znamená, že v určitém okamžiku se poloha osoby změní. Příznaky jako je výška či velikost jsou u shrbených osob také velice nestálé. Tato úprava však nepřinesla očekávaný výsledek. Algoritmus kvůli tomu může shýbající se osobu prohlásit za nově příchozí.

5.9 Shrnutí

Tato sekce popisuje seznam všech metod použitých v mnou navrhovaném algoritmu detekce a sledování osob. Podrobnější popis jednotlivých metod je v příslušné sekci výše.

1. Preprocessing
 - a) Eliminace šumu
 - b) Oprava děr
2. Detekce
 - a) Potlačení nemaxim
3. Korekce
 - a) Potlačení potenciálních osob s malou výškou
 - b) Sloučení do středu „blobu“
 - c) Potlačení potenciálních osob s malou hodnotou „h“
4. Segmentace – vytvoření dvou segmentovaných snímků ke každé osobě
 - a) Primitivní segmentace, není-li to možné, pak watershed
 - b) Selektce hlavy
5. Vytvoření příznaků
 - a) Plocha
 - b) Výška
 - c) HOD
6. Klasifikace
 - a) Diskrétní AdaBoost
7. Sledování
 - a) Maďarský algoritmus – přiřazení osob k sobě
 - b) Kalmanův filtr – odhad polohy osoby

5.10 Implementace

Výše popsané algoritmy jsem implementoval v jazyce Java za použití knihovny OpenCV. V Javě jsou napsány i ostatní součásti systému, jehož je můj algoritmus součástí.

Knihovna OpenCV¹² je jedna z nejrozšířenějších knihoven, které se specializují na počítačové vidění. Převzaty z ní byly všechny metody strojového učení (zmíněny v sekci 5.7.2), implementace Kalmanova filtru a Gaussovo rozostření (pouze metoda neignorující nuly, ostatní mnou použité způsoby rozostření OpenCV neobsahuje).

Implementaci Maďarského algoritmu jsem převzal z repozitáře Kevina Sterna na Github.com [53], kde byla zveřejněna pod licencí MIT¹³.

Ostatní součásti algoritmu jsem naprogramoval přímo v jazyce Java.

¹²<http://opencv.org>

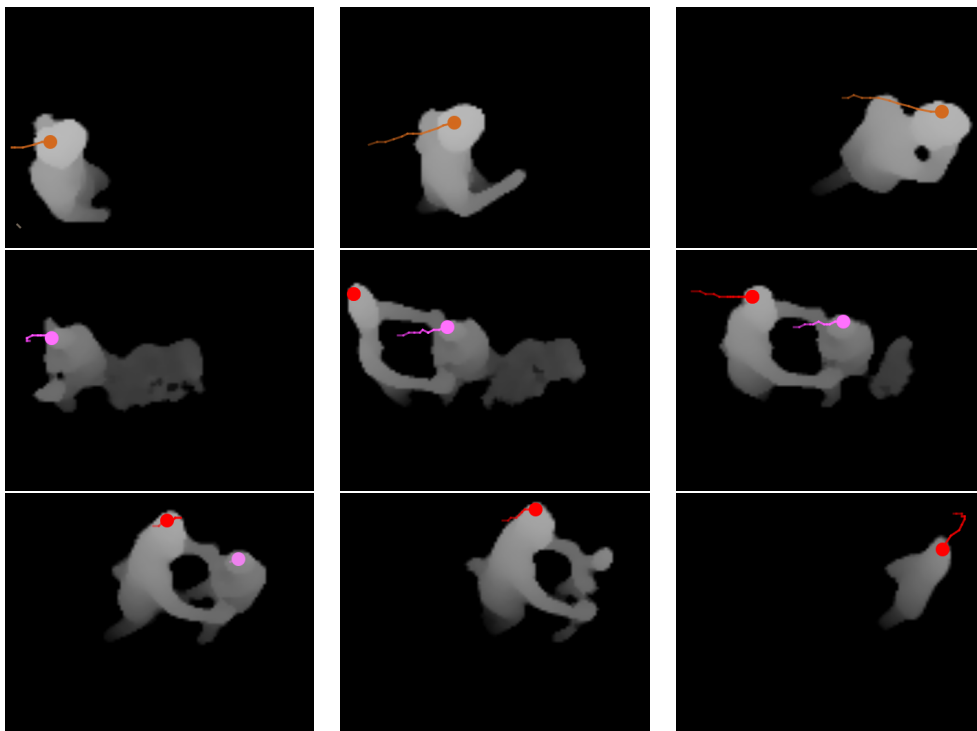
¹³<https://opensource.org/licenses/MIT>

Testování

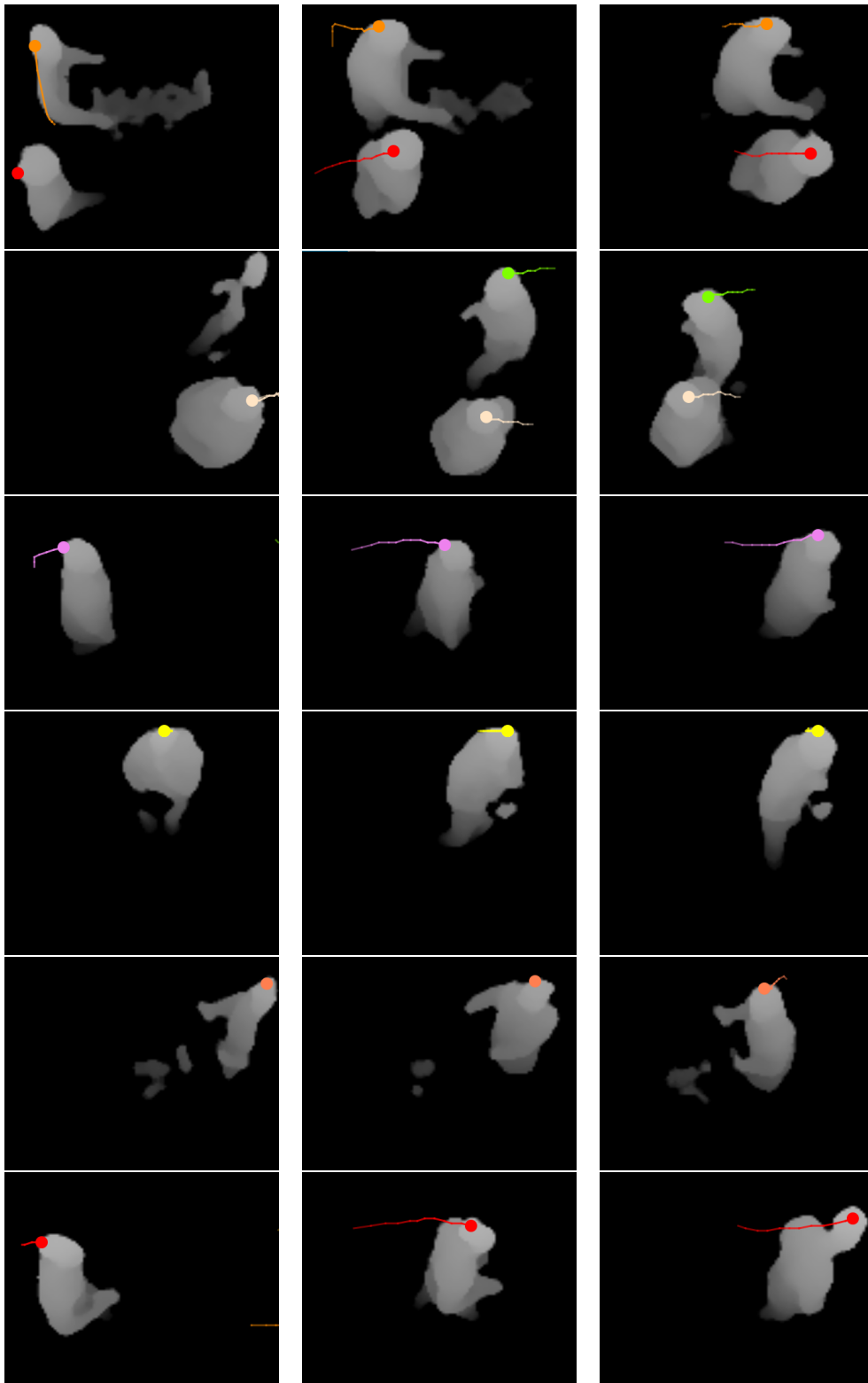
Během testování jsem měřil schopnost algoritmu stanovit počet osob na snímku. Algoritmus byl spuštěn nad množinou snímků a následně porovnán s referenčním řešením. Toto referenční řešení je množina snímků, na které byl počet osob zjištěn člověkem.

6.1 Ukázka výsledků

Následující obrázky ukazují výsledky algoritmu.



6. TESTOVÁNÍ



6.2 Metrika

Tato sekce popisuje způsob, jakým byly testované metody porovnávány. Během testování jsem měřil schopnost algoritmu stanovit počet osob na snímku.

Způsob testování sledovací fáze byl odlišný od ostatního testování. Sledování totiž má za úkol nejen najít na snímku osoby, ale zároveň je i správně přiřadit k sobě. O metrice použité k hodnocení sledování pojednává sekce 6.7.

Všechny ostatní metody jsem však testoval podle postupu, který je popsán v této sekci.

Testovaný algoritmus byl spuštěn nad množinou snímků a následně porovnán s referenčním řešením. Toto referenční řešení je množina snímků, na které byl počet osob zjištěn manuálně.

Každý snímek můžeme označit tímto příznakem

Positive (P) – na snímku se nachází osoba

Negative (N) – na snímku se žádná osoba nenachází

V tabulkách níže jsou uvedeny naměřené hodnoty

True positive (TP) – počet snímků, na kterých byla správně identifikována osoba

True negative (TN) – počet snímků, o kterých algoritmus správně určil, že na nich nikdo není

False positive (FP) – počet snímků, na kterých algoritmus našel více osob, než na snímku skutečně bylo

False negative (FN) – počet snímků, na nichž algoritmus některé osoby nedetekoval

Accuracy – přesnost algoritmu (udává, kolik procent snímků bylo ohodnoceno správně)

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision – schopnost algoritmu zachytit pouze skutečné osoby

$$precision = \frac{TP}{TP + FP}$$

Recall – schopnost algoritmu zachytit co nejvíce osob

$$recall = \frac{TP}{TP + FN}$$

Dataset o 2 827 snímcích obsahoval 2 616 osob, které byly zachyceny na 2 482 snímcích (345 snímků bylo prázdných).

6.3 Detekce potenciálních osob

Detekce potenciálních osob má na zadaném snímku najít místa, kde by se mohly nacházet osoby.

Výsledek každé metody závisí na jejich parameterech. Změnou parametrů je možné zvýšit počet TP na úkor TN a naopak.

U detekce potenciálních osob je důležitá co nejvyšší hodnota TP a co nejnižší FN, zatímco nízká hodnota TN a rozumně vysoká hodnota TP příliš nevadí. Falešně pozitivní výsledky mohou odstranit ostatní fáze.

Výsledky detekce popisuje tabulka 6.1.

Tabulka 6.1: Úspěšnost detekce osob

Úspěšnost detekčních algoritmů. Parametry algoritmů byly nastaveny tak, aby accuracy byla co nejvyšší.

Metoda	TP	TN	FP	FN	Accuracy	AUC
Extended h-maxima	1 849	180	87	63	93,12 %	0,952
„Rozdělující“ potl. nem.	1 839	181	100	59	92,84 %	0,954
Potlačení nemaxim	1 832	180	114	53	92,34 %	0,931
Water Filling	1 678	215	182	104	86,87 %	0,864
K-means	1 619	230	52	278	84,86 %	0,881
Weighted k-means	1 609	231	54	285	84,44 %	0,888
Hill climbing	1 493	220	114	352	78,61 %	0,798
Narůst. oblastí	1 486	221	118	354	78,34 %	0,782

6.4 Preprocessing

Preprocessing má za úkol zvýšit úspěšnost detekčního algoritmu. Proto je pro otestování preprocessingu nutné testovat samotný detekční algoritmus s použitím preprocessingu a bez něj.

Jako detekční algoritmus bylo vybráno potlačení nemaximálních hodnot (viz sekce 5.3.2) s poloměrem skenovacího okna 9 px a s minimální hodnotou 70. Tyto hodnoty jsem vybral proto, že s nimi má výsledek detekce velmi nízkou hodnotu FN a přijatelný počet osob detekovaných omylem.

Na tento detekční algoritmus byla aplikována úprava „umístění do středu úrovně“. Jak ukazuje tabulka 6.4 v následující sekci, tato úprava je pro potlačení nemaxim prakticky nezbytná.

Všechny rozostřovací techniky v tabulkách 6.2 a 6.3 mají velikost okénka 9. Směrodatnou odchylku (v případě Gaussova rozostření) jsem stanovil jako 2. Tyto parametry umožnily snížit počet falešně detekovaných osob, aniž by došlo k výraznému nárůstu FN.

Podle tabulky 6.2 má nejlepší precision i recall Gaussovo rozostření. Dobrého výsledků dosahuje také rozostření pomocí maxima. Rozostření pomocí mediánu dosahuje naopak velmi špatných hodnot.

Tato situace se však zásadně změní, jsou-li z výsledků smazány všechny potenciální osoby s hodnotou „h“ menší než 20.

Tabulka 6.2: Úspěšnost preprocessingu

Metoda	TP	TN	FP	FN	Acc.	Prec.	Rec.
Původní snímek	1 432	105	637	5	70,54 %	69,21 %	99,65 %
Gaussovo rozostření	1 545	120	512	2	76,41 %	75,11 %	99,87 %
Nenulový Gauss	1 165	93	908	13	57,73 %	56,20 %	98,90 %
Rozostření mediánem	115	35	2 019	10	6,88 %	5,39 %	92,00 %
Nenulový medián	1 318	102	756	3	65,17 %	63,55 %	99,77 %
Maximum	1 507	112	555	5	74,30 %	73,08 %	99,67 %

Tabulka 6.3: Úspěšnost preprocessingu po smazání výsledků s malým „h“

Metoda	TP	TN	FP	FN	Acc.	Prec.	Rec.
Původní snímek	1 749	129	295	6	86,19 %	85,57 %	99,66 %
Gaussovo rozostření	1 771	148	254	6	88,07 %	87,46 %	99,66 %
Nenulový Gauss	1 782	154	172	71	88,85 %	91,20 %	96,17 %
Rozostření mediánem	1 761	164	157	97	88,34 %	91,81 %	94,78 %
Nenulový medián	1 741	162	145	131	87,33 %	92,31 %	93,00 %
Maximum	1 764	135	274	6	87,15 %	86,56 %	99,66 %

Po zmíněné úpravě je rozostření pomocí mediánu (v obou variantách) naopak jedno z nejlepších, i když je to dáno částečně snížením počtu detekovaných osob (parametry detekčního algoritmu byly schválně vybrány tak, aby počet FN byl co nejmenší a účinnost rozostření se mohla projevit).

Všechny druhy preprocessingu mají srovnatelnou accuracy, navíc ne o moc větší než samotná detekce na původních snímcích. Změna parametrů rozostření nemá na tuto skutečnost velký vliv. S rostoucí velikostí okna roste precision, ale klesá recall.

6.5 Úprava výsledků detekce

Tabulka 6.4 ukazuje, jak se mění výsledky jednotlivých detekčních algoritmů po použití postprocessingu. První řádek vždy ukazuje výsledky testování daného algoritmu se zvolenými parametry, kdy nebyl použit žádný postprocessing.

Parametry jsem vybíral tak, aby měly nízkou hodnotu FN. Postprocessing totiž dokáže eliminovat vícekrát detekovanou jednu osobu a tím snížit FP. S nedetekovanými osobami však nemůže udělat nic.

Na algoritmy potlačení nemaxim a hill climbing nebyla aplikována úprava „hill climbing“. Tyto metody totiž samy o sobě hledají lokální maxima (detektor hill climbing dokonce téměř stejným způsobem). Jejich výsledky se tak s touto úpravou vůbec nezměnily.

6. TESTOVÁNÍ

Algoritmus slití úrovní byl prováděn vždy v kombinaci s hill climbingem, pokud nebyl přínos hill climbingu nulový. Nejdříve byl aplikován hill climbing, poté slití úrovní. Důvodem je fakt, že tento algoritmus může správně fungovat pouze v případě, že jsou všechny body shluknuty vedle sebe (do lokálních maxim). Toho může docílit buď hill climbing nebo samotný detekční algoritmus zaměřený na lokální maxima.

Tabulka 6.4: Úspěšnost úpravy výsledků detekce

Detekce	Úprava	TP	TN	FP	FN	Acc.	Prec.	Rec.
Potlačení nemaxim	—	710	68	1 397	4	35,70 %	33,70 %	99,44 %
	Slití úrovní	1 514	122	538	5	75,08 %	73,78 %	99,67 %
	H	1 774	146	253	6	88,11 %	87,52 %	99,66 %
	Kombinace	1 776	146	251	6	88,21 %	87,62 %	99,66 %
K-means	—	1 129	154	871	25	58,88 %	56,45 %	97,83 %
	H	1 837	164	134	44	91,83 %	93,20 %	97,66 %
	Hill climb.	1 551	155	447	26	78,29 %	77,63 %	98,35 %
	Slití úrovní	1 720	155	276	28	86,05 %	86,17 %	98,40 %
	Kombinace	1 828	161	159	31	91,28 %	92,00 %	98,33 %
Vážený k-means	—	1 194	152	813	20	61,77 %	59,49 %	98,35 %
	H	1 820	164	153	42	91,05 %	92,25 %	97,74 %
	Hill climb.	1 547	156	455	21	78,16 %	77,27 %	98,66 %
	Slití úrovní	1 725	156	275	23	86,32 %	86,25 %	98,68 %
	Kombinace	1 814	160	178	27	90,59 %	91,06 %	98,53 %
Water filling	—	1 663	145	347	24	82,97 %	82,74 %	98,58 %
	H	1 683	146	318	32	83,94 %	84,11 %	98,13 %
	Hill climb.	1 667	145	341	26	83,16 %	83,02 %	98,46 %
	Slití úrovní	1 671	145	337	26	83,34 %	83,22 %	98,47 %
	Kombinace	1 691	146	313	29	84,30 %	84,38 %	98,31 %
Hill climbing	—	999	94	1 085	1	50,16 %	47,93 %	99,90 %
	H	1 685	133	343	18	83,43 %	83,09 %	98,94 %
	Slití úrovní	1 000	95	1 082	2	50,25 %	48,03 %	99,80 %
	Kombinace	1 688	133	340	18	83,57 %	83,23 %	98,94 %
Narůstání oblastí	—	975	130	1 019	55	50,71 %	48,90 %	94,66 %
	H	1 632	170	316	61	82,70 %	83,78 %	96,40 %
	Hill climb.	978	130	1 016	55	50,85 %	49,05 %	94,68 %
	Slití úrovní	994	130	999	56	51,58 %	49,87 %	94,67 %
	Kombinace	1 640	170	308	61	83,07 %	84,19 %	96,41 %

6.6 Klasifikace

Další fází algoritmu je klasifikace, neboli ověření, zda detekovaný „blob“ je skutečně osoba, nebo ne.

Během klasifikace je nutné odstranit co nejvíce false positive potenciálních osob, naopak true positive by měly zůstat zachovány. Na rozdíl od minulé fáze je nutné mít co nejvyšší obě hodnoty – TP i TN.

Tabulky 6.5–6.10 uvádějí výsledky testovaných klasifikačních metod, které byly spuštěny nad výstupy z předchozí fáze. Výsledek klasifikace tedy závisí do velké míry na detekci potenciálních osob.

Jako vstup validační fáze byl použit výstup potlačení nemaxim, jehož parametry jsem nastavil tak, aby měl nízkou hodnotu FN. Jeho výsledky jsou uvedeny v horním řádku tabulek.

V následujících tabulkách jsou uvedeny výsledky jednotlivých detekčních metod. Nejdříve jsou uvedeny výsledky jednotlivých příznaků sloužících jako vstup jednotlivým metodám. V dalších tabulkách se nacházejí výsledky klasifikačních metod při použití vybraných kombinací příznaků.

6.6.1 Jednotlivé příznaky

Následující tabulky obsahují výsledky testovaných metod strojového učení, které jsou naučeny rozpoznávat osoby pomocí jednoho příznaku.

Jak je možné na výsledcích vidět, mnoho příznaků výsledek dokonce zhorší. V kombinaci s ostatními příznaky však dokáží vytvořit dobrý klasifikační model.

První sloupec v tabulce vlevo udává způsob segmentace.

Zajímavým jevem je různé chování příznaku výška pro různé druhy segmentace. Výška osoby by měla být na způsobu segmentace nezávislá. Fakt, že se hodnoty (a i výsledky) liší, je způsoben implementací.

Při vyhodnocování výšky algoritmus hledá globální maximum segmentovaného snímku. Kdyby totiž vrátil pouze hodnotu hloubkové mapy na místě, kde se osoba nachází, nemuselo by jít o skutečnou výšku člověka. Ne všechny detekční algoritmy umísťují osoby do lokálních maxim.

Příznak výška pro celý snímek tudíž nedává smysl a v tabulce se ocitl omylem. Příznak totiž obsahuje výšku nejvyšší osoby na snímku.

Vzhledem k tomu, že v kombinaci s SVM dosáhl překvapivě dobré úspěšnosti, rozhodl jsem se ho v tabulce ponechat pro ilustraci.

Položky hill climbing, k-means, ...jsou jednotlivými druhy příznaku „výstup různých metod detekce“.

Značky vedle příznaku průměr popisují, jakým směrem byl průměr počítán.

6. TESTOVÁNÍ

Tabulka 6.5: Výsledky klasifikace – rozhodovací strom

Metoda		TP	TN	FP	FN	Acc.	Prec.	Rec.
Původní hodnota		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Poloha x		1 644	164	150	221	82,97 %	91,64 %	88,15 %
Poloha y		1 139	171	95	774	60,12 %	92,30 %	59,54 %
Orig.	HOD	1 459	197	76	447	76,00 %	95,05 %	76,55 %
	Výška	1 743	208	65	163	89,54 %	96,40 %	91,45 %
	H	1 728	199	58	194	88,44 %	96,75 %	89,91 %
Watershed	G-kulatost	1 771	147	190	71	88,02 %	90,31 %	96,15 %
	H	1 741	199	66	173	89,03 %	96,35 %	90,96 %
	Kulatost	1 768	180	85	146	89,40 %	95,41 %	92,37 %
	Plocha	1 726	178	101	174	87,38 %	94,47 %	90,84 %
	Průměr –	1 719	162	104	194	86,32 %	94,30 %	89,86 %
	Průměr	1 736	180	117	146	87,93 %	93,69 %	92,24 %
	Průměr /	1 700	162	112	205	85,45 %	93,82 %	89,24 %
	Průměr \	1 738	159	128	154	87,06 %	93,14 %	91,86 %
	Průměr průměrů	1 739	163	128	149	87,29 %	93,14 %	92,11 %
	Rozptyl průměrů	1 494	182	81	422	76,92 %	94,86 %	77,97 %
	Výška	1 751	208	56	164	89,90 %	96,90 %	91,44 %
Hlava	G-kulatost	1 777	148	183	71	88,34 %	90,66 %	96,16 %
	H	1 752	200	60	167	89,58 %	96,69 %	91,30 %
	Kulatost	1 707	185	87	200	86,83 %	95,15 %	89,51 %
	Plocha	1 678	181	104	216	85,31 %	94,16 %	88,60 %
	Průměr –	1 708	181	100	190	86,69 %	94,47 %	89,99 %
	Průměr	1 778	172	108	121	89,49 %	94,27 %	93,63 %
	Průměr /	1 725	176	99	179	87,24 %	94,57 %	90,60 %
	Průměr \	1 735	185	89	170	88,11 %	95,12 %	91,08 %
	Průměr průměrů	1 699	184	87	209	86,42 %	95,13 %	89,05 %
	Rozptyl průměrů	1 724	174	113	168	87,10 %	93,85 %	91,12 %
Výška	1 746	234	23	176	90,87 %	98,70 %	90,84 %	
Hill climbing		1 509	238	15	417	80,17 %	99,02 %	78,35 %
K-means		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Weighted k-means		1 105	233	12	829	61,40 %	98,93 %	57,14 %
Narůst. oblastí		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Water filling		1 597	224	36	322	83,57 %	97,80 %	83,22 %

Tabulka 6.6: Výsledky klasifikace – náhodný les

Metoda		TP	TN	FP	FN	Acc.	Prec.	Rec.
Původní hodnota		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Poloha x		1 644	164	150	221	82,97 %	91,64 %	88,15 %
Poloha y		1 048	186	75	870	56,63 %	93,32 %	54,64 %
Orig.	HOD	1 452	202	64	461	75,91 %	95,78 %	75,90 %
	Výška	1 749	207	66	157	89,77 %	96,36 %	91,76 %
	H	1 721	198	67	193	88,07 %	96,25 %	89,92 %
Watershed	G-kulatost	1 777	148	183	71	88,34 %	90,66 %	96,16 %
	H	1 748	205	53	173	89,63 %	97,06 %	90,99 %
	Kulatost	1 701	185	87	206	86,55 %	95,13 %	89,20 %
	Plocha	1 691	179	107	202	85,82 %	94,05 %	89,33 %
	Průměr –	1 708	181	100	190	86,69 %	94,47 %	89,99 %
	Průměr	1 782	170	110	117	89,58 %	94,19 %	93,84 %
	Průměr /	1 725	176	99	179	87,24 %	94,57 %	90,60 %
	Průměr \	1 735	185	89	170	88,11 %	95,12 %	91,08 %
	Průměr průměrů	1 696	184	87	212	86,28 %	95,12 %	88,89 %
	Rozptyl průměrů	1 725	174	113	167	87,15 %	93,85 %	91,17 %
	Výška	1 760	211	48	160	90,45 %	97,35 %	91,67 %
	Hlava	G-kulatost	1 738	202	63	176	89,03 %	96,50 %
H		1 756	176	105	142	88,66 %	94,36 %	92,52 %
Kulatost		1 771	147	190	71	88,02 %	90,31 %	96,15 %
Plocha		1 726	160	122	171	86,55 %	93,40 %	90,99 %
Průměr –		1 711	180	110	178	86,78 %	93,96 %	90,58 %
Průměr		1 701	162	112	204	85,50 %	93,82 %	89,29 %
Průměr /		1 738	159	128	154	87,06 %	93,14 %	91,86 %
Průměr \		1 740	163	128	148	87,33 %	93,15 %	92,16 %
Průměr průměrů		1 518	184	76	401	78,11 %	95,23 %	79,10 %
Rozptyl průměrů		1 768	181	84	146	89,44 %	95,46 %	92,37 %
Výška		1 757	210	54	158	90,27 %	97,02 %	91,75 %
Hill climbing		1 597	224	36	322	83,57 %	97,80 %	83,22 %
K-means		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Weighted k-means		1 509	238	15	417	80,17 %	99,02 %	78,35 %
Narůst. oblastí		1 105	233	12	829	61,40 %	98,93 %	57,14 %
Water filling		1 764	147	197	71	87,70 %	89,95 %	96,13 %

6. TESTOVÁNÍ

Tabulka 6.7: Výsledky validace – boosting

Metoda		TP	TN	FP	FN	Acc.	Prec.	Rec.
Původní hodnota		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Poloha x		1 598	164	147	270	80,86 %	91,58 %	85,55 %
Poloha y		1 495	148	165	371	75,40 %	90,06 %	80,12 %
Orig.	HOD	1 764	147	197	71	87,70 %	89,95 %	96,13 %
	Výška	1 598	164	147	270	80,86 %	91,58 %	85,55 %
	H	1 495	148	165	371	75,40 %	90,06 %	80,12 %
Watershed	G-kulatost	1 764	147	197	71	87,70 %	89,95 %	96,13 %
	H	1 730	234	24	191	90,13 %	98,63 %	90,06 %
	Kulatost	1 768	180	85	146	89,40 %	95,41 %	92,37 %
	Plocha	1 778	164	120	117	89,12 %	93,68 %	93,83 %
	Průměr –	1 744	160	134	141	87,38 %	92,86 %	92,52 %
	Průměr	1 771	164	136	108	88,80 %	92,87 %	94,25 %
	Průměr /	1 701	162	125	191	85,50 %	93,15 %	89,90 %
	Průměr \	1 752	159	133	135	87,70 %	92,94 %	92,85 %
	Průměr průměrů	1 766	161	132	120	88,44 %	93,05 %	93,64 %
	Rozptyl průměrů	1 620	175	133	251	82,38 %	92,41 %	86,58 %
	Výška	1 743	234	27	175	90,73 %	98,47 %	90,88 %
	Hlava	G-kulatost	1 777	148	183	71	88,34 %	90,66 %
H		1 735	234	23	187	90,36 %	98,69 %	90,27 %
Kulatost		1 707	185	87	200	86,83 %	95,15 %	89,51 %
Plocha		1 716	179	108	176	86,97 %	94,08 %	90,70 %
Průměr –		1 709	181	100	189	86,74 %	94,47 %	90,04 %
Průměr		1 778	172	108	121	89,49 %	94,27 %	93,63 %
Průměr /		1 725	176	99	179	87,24 %	94,57 %	90,60 %
Průměr \		1 735	185	89	170	88,11 %	95,12 %	91,08 %
Průměr průměrů		1 739	178	96	166	87,98 %	94,77 %	91,29 %
Rozptyl průměrů		1 726	173	115	165	87,15 %	93,75 %	91,27 %
Výška	1 746	234	23	176	90,87 %	98,70 %	90,84 %	
Hill climbing		1 509	238	15	417	80,17 %	99,02 %	78,35 %
K-means		1 105	233	12	829	61,40 %	98,93 %	57,14 %
Weighted k-means		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Narůst. oblastí		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Water filling		1 597	224	36	322	83,57 %	97,80 %	83,22 %

Tabulka 6.8: Výsledky validace – SVM

Metoda		TP	TN	FP	FN	Acc.	Prec.	Rec.
Původní hodnota		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Poloha x		1 220	185	87	687	64,48 %	93,34 %	63,97 %
Poloha y		919	194	80	986	51,08 %	91,99 %	48,24 %
Orig.	HOD	1 473	170	118	418	75,40 %	92,58 %	77,90 %
	Výška	1 736	234	38	171	90,41 %	97,86 %	91,03 %
	H	1 712	234	22	211	89,31 %	98,73 %	89,03 %
Watershed	G-kulatost	1 771	147	190	71	88,02 %	90,31 %	96,15 %
	H	1 735	234	25	185	90,36 %	98,58 %	90,36 %
	Kulatost	1 479	228	13	459	78,34 %	99,13 %	76,32 %
	Plocha	1 690	148	190	151	84,35 %	89,89 %	91,80 %
	Průměr –	1 045	226	41	867	58,33 %	96,22 %	54,65 %
	Průměr	1 448	164	120	447	73,98 %	92,35 %	76,41 %
	Průměr /	1 200	151	148	680	62,00 %	89,02 %	63,83 %
	Průměr \	1 559	153	140	327	78,57 %	91,76 %	82,66 %
	Průměr průměrů	1 299	151	150	579	66,54 %	89,65 %	69,17 %
	Rozptyl průměrů	1 752	147	183	97	87,15 %	90,54 %	94,75 %
	Výška	1 744	234	29	172	90,78 %	98,36 %	91,02 %
Hlava	G-kulatost	1 764	147	197	71	87,70 %	89,95 %	96,13 %
	H	1 730	234	26	189	90,13 %	98,52 %	90,15 %
	Kulatost	1 657	151	182	189	82,97 %	90,10 %	89,76 %
	Plocha	1 764	147	195	73	87,70 %	90,05 %	96,03 %
	Průměr –	1 755	148	192	84	87,33 %	90,14 %	95,43 %
	Průměr	1 741	150	184	104	86,78 %	90,44 %	94,36 %
	Průměr /	1 751	149	189	90	87,20 %	90,26 %	95,11 %
	Průměr \	1 753	148	190	88	87,24 %	90,22 %	95,22 %
	Průměr průměrů	1 742	150	187	100	86,83 %	90,31 %	94,57 %
	Rozptyl průměrů	1 764	147	196	72	87,70 %	90,00 %	96,08 %
Výška	1 746	234	25	174	90,87 %	98,59 %	90,94 %	
Hill climbing		1 509	238	15	417	80,17 %	99,02 %	78,35 %
K-means		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Weighted k-means		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Narůst. oblastí		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Water filling		1 597	224	36	322	83,57 %	97,80 %	83,22 %

6. TESTOVÁNÍ

Tabulka 6.9: Výsledky validace – 5-nn

Metoda		TP	TN	FP	FN	Acc.	Prec.	Rec.
Původní hodnota		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Poloha x		1 174	175	113	717	61,91 %	91,22 %	62,08 %
Poloha y		1 010	173	95	901	54,29 %	91,40 %	52,85 %
Orig.	HOD	1 716	198	81	184	87,84 %	95,49 %	90,32 %
	Výška	1 246	206	51	676	66,64 %	96,07 %	64,83 %
	H	1 704	207	55	213	87,70 %	96,87 %	88,89 %
Watershed	G-kulatost	1 764	147	197	71	87,70 %	89,95 %	96,13 %
	H	1 713	206	57	203	88,07 %	96,78 %	89,41 %
	Kulatost	1 627	190	62	300	83,39 %	96,33 %	84,43 %
	Plocha	1 668	183	94	234	84,95 %	94,67 %	87,70 %
	Průměr –	1 186	193	55	745	63,29 %	95,57 %	61,42 %
	Průměr	1 052	168	120	839	55,99 %	89,76 %	55,63 %
	Průměr /	997	182	78	922	54,11 %	92,74 %	51,95 %
	Průměr \	976	179	91	933	53,01 %	91,47 %	51,13 %
	Průměr průměrů	1 155	167	122	735	60,67 %	90,45 %	61,11 %
	Rozptyl průměrů	1 301	185	86	607	68,20 %	93,80 %	68,19 %
	Výška	1 242	206	51	680	66,45 %	96,06 %	64,62 %
Hlava	G-kulatost	1 764	147	197	71	87,70 %	89,95 %	96,13 %
	H	1 727	207	54	191	88,76 %	96,97 %	90,04 %
	Kulatost	1 465	195	69	450	76,18 %	95,50 %	76,50 %
	Plocha	1 493	187	86	413	77,10 %	94,55 %	78,33 %
	Průměr –	1 356	213	45	565	72,01 %	96,79 %	70,59 %
	Průměr	1 005	193	65	916	54,98 %	93,93 %	52,32 %
	Průměr /	1 532	203	61	383	79,62 %	96,17 %	80,00 %
	Průměr \	1 453	203	55	468	76,00 %	96,35 %	75,64 %
	Průměr průměrů	1 413	205	48	513	74,25 %	96,71 %	73,36 %
	Rozptyl průměrů	1 261	187	79	652	66,45 %	94,10 %	65,92 %
Výška	1 247	207	45	680	66,73 %	96,52 %	64,71 %	
Hill climbing		1 764	147	197	71	87,70 %	89,95 %	96,13 %
K-means		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Weighted k-means		1 509	238	15	417	80,17 %	99,02 %	78,35 %
Narůst. oblastí		1 764	147	197	71	87,70 %	89,95 %	96,13 %
Water filling		1 764	147	197	71	87,70 %	89,95 %	96,13 %

6.6.2 Vybrané kombinace

V tabulce 6.10 se nacházejí vybrané kombinace příznaků. Tabulka obsahuje nejúspěšnější kombinaci, jednu z největších (kombinuje všechny příznaky, které byly samy o sobě úspěšné) a další kombinace vybrané náhodně, aby ilustrovaly schopnost klasifikace jednotlivých metod.

Symbole (P), (H) a (O) v tabulce značí způsob segmentace – P znamená celá postava, H pouze hlava a O původní snímek.

Do tabulky se nevešly tyto řádky: (1) – h (P, O) plocha (P, H), kulatost (P, H), průměr – průměrná hodnota i rozptyl (P, H), výška (P, H), G-kulatost (P, H), x, y; (2) – Plocha (P), h (P, O), kulatost (P), průměr.

Kombinoval jsem především ty příznaky, které dosahovaly uspokojivých výsledků v předchozích tabulkách. Konkrétně to jsou: kulatost (celého těla), g-kulatost (celého těla i hlavy), plocha (watershed i hlava), h (všechny segmentace) a výška hlavy.

Nejlepší klasifikátor dosáhl accuracy 91,92 %, tedy o 4,22 procentního bodu více, než původní data. SVM s omylem zařazeným příznakem „nejvyšší místo na snímku“ zvýšily accuracy o 2,71 procentního bodu. Klasifikační část je tedy místem, kde je možné algoritmus ještě zlepšit.

Na druhou stranu výsledky sledování (tabulka 6.12) na zcela jiných datech ukazují, že tato klasifikace skutečně nezanedbatelně zlepšuje přesnost algoritmu.

6.7 Sledování

Sledovací část vyžaduje pro testování jinou metriku než předchozí části. Je to dáno tím, že nás zajímá nejen počet detekovaných osob, ale i to, jak dobře dokáže přiřazovat jednotlivé snímky mezi sebou.

Zkoušel jsem tři dále popsané metriky. Protože každá z metod je náchylná na určité jevy (viz dále), rozhodl jsem se použít všechny tři.

6.7.1 Počet osob v časovém intervalu

První metrikou je počet sledovaných osob. Jedná se o unikátní ID osob, nikoli o počet osob detekovaných na snímcích. Testování probíhá na delším časovém intervalu než ostatní testy. Záznam obsahuje 11 593 snímků, kterými prošlo 125 unikátních osob.

Testovací metoda se zaměřovala na počet osob, které do prostoru vstoupily a které z něj vystoupily. Manuálně jsem vytvořil referenční data, která obsahují informaci, kolik lidí do prostoru vstoupilo z levé strany, kolik z pravé a kolik jej z levé či pravé strany opustilo.

Tato data byla porovnána s výsledkem sledování. Každá osoba v testovacích datech má přiřazenu stranu, ze které přišla. Pokud má na prvním snímku, na kterém se objevuje, souřadnici na ose y menší, než je polovina šířky obrázku, přišla zleva. V opačném případě přišla zprava. Podobně je osobám přiřazena strana odchodu (podle posledního výskytu).

Algoritmus pak porovnává počet osob podle místa příchodu a odchodu.

Tato metoda ignoruje počet osob na jednotlivých snímcích a zaměřuje se na osoby samotné. Z toho vyplývá výhoda – algoritmus, který najde osobu

Tabulka 6.10: Vybrané výsledky klasifikace

Metoda	Příznaky	TP	TN	FP	FN	Accuracy	Precision	Recall
Rozhodovací strom	(1)	1 731	227	18	203	89,86 %	98,97 %	89,50 %
	Výška (P), h (H)	1 753	199	68	159	89,58 %	96,27 %	91,68 %
	Výška (H), h(P), kulatost(P)	1 751	193	66	169	89,22 %	96,37 %	91,20 %
	(2) – (H), výška (H), x, y	1 725	218	25	211	89,17 %	98,57 %	89,10 %
Náhodný lest	Výška (P), plocha (P), HOD, x, y	1 611	222	28	318	84,12 %	98,29 %	83,51 %
	(1)	1 764	208	42	165	90,50 %	97,67 %	91,45 %
	Výška (P) a h (H)	1 754	205	59	161	89,90 %	96,75 %	91,59 %
	Výška (H), h(P), kulatost(P)	1 751	199	50	179	89,49 %	97,22 %	90,73 %
Boosting	(2)	1 781	211	36	151	91,42 %	98,02 %	92,18 %
	Výška (P), plocha (P), HOD, x, y	1 705	160	149	165	85,59 %	91,96 %	91,18 %
	(1)	1 772	191	78	138	90,09 %	95,78 %	92,77 %
	Výška (P) a h (H)	1 746	234	23	176	90,87 %	98,70 %	90,84 %
SVM	Výška (H), h(P), kulatost(P)	1 730	216	30	203	89,31 %	98,30 %	89,50 %
	(2)	1 783	212	46	138	91,56 %	97,48 %	92,82 %
	Výška (P), plocha (P), HOD, x, y	1 787	216	32	144	91,92 %	98,24 %	92,54 %
	(1)	1 579	216	48	336	82,38 %	97,05 %	82,45 %
5-nn	Výška (P) a h (H)	1 736	234	21	188	90,41 %	98,80 %	90,23 %
	Výška (H), h(P), kulatost(P)	1 775	204	57	143	90,82 %	96,89 %	92,54 %
	(2)	1 791	162	123	103	89,63 %	93,57 %	94,56 %
	Výška (P), plocha (P), HOD, x, y	1 764	147	197	71	87,70 %	89,95 %	96,13 %
5-nn	(1)	1 747	200	56	176	89,35 %	96,89 %	90,85 %
	Výška (P) a h (H)	1 729	206	60	184	88,80 %	96,65 %	90,38 %
	Výška (H), h(P), kulatost(P)	1 715	205	48	211	88,11 %	97,28 %	89,04 %
	(2)	1 737	207	47	188	89,22 %	97,37 %	90,23 %
	Výška (P), plocha (P), HOD, x, y	1 720	220	39	200	89,03 %	97,78 %	89,58 %

o několik snímků dříve (nebo později) než referenční řešení je stejně úspěšný jako ten, který ji najde ve stejný čas.

Další výhodou této metody je snadnost přípravy velkého počtu referenčních dat.

Nevýhodou je možnost, že testovaný algoritmus ignoruje skutečnou osobu a místo ní detekuje (třeba i v jiném časovém intervalu) jiný objekt. Ačkoli takový algoritmus udělal dvě chyby, ve výsledku má stoprocentní úspěšnost.

V následující tabulce uvádím dva výsledky mého algoritmu. Prvním z nich je počet osob, které algoritmus zaznamenal. V druhém jsou smazány ty osoby, které byly zjevně detekovány omylem. Má-li algoritmu sloužit pro počítání osob, není problém tyto osoby vyřadit i v reálném nasazení.

Jedná se o osoby, které v prostoru byly přítomny 10 snímků nebo méně a které na ose x neurazily vzdálenost větší než 25 pixelů (obrázek je široký 160 px). Konstanty 10 a 25 jsem stanovil před začátkem měření tak, aby vyřadily skutečně jen chybné vzorky.

Tabulka 6.11: Počet osob podle strany vstupu a odchodu

	Příchod zleva	Příchod zprava	Odchod vlevo	Odchod vpravo	Celkem
Skutečnost	79	26	20	85	105
Algoritmus	100 +27 %	50 +92 %	42 +110 %	108 +27 %	150 +43 %
Algoritmus s vyřazením	90 +14 %	33 +26 %	33 +65 %	90 +6 %	123 +17 %

6.7.2 Počet osob na snímcích

Další metrikou bylo porovnání počtu osob na každém snímku, stejně jako v testování předchozích částí. Lišila se však data – nebyl zde kladen důraz na různorodost snímků, ale naopak bylo potřeba, aby šlo o kontinuální záznam.

Data obsahují 5564 snímků, na nichž byla 5476krát detekována osoba (celkem 39 unikátních osob).

Metrika počtu osob na jednotlivých snímcích na rozdíl od té předchozí penalizuje situaci, kdy algoritmus najde stejný počet osob jako v referenčním řešení, ale na jiných snímcích.

Problémem je, že vůbec nezpracovává informaci o ID osob, pouze jejich počet.

Výsledky sledování jsou uvedeny v tabulce 6.12.

6.7.3 Vazby

Třetí metoda testuje správnost přiřazení jednotlivých „vazeb“. Mezi dvěma detekcemi osob na sousedních snímcích je vazba, pokud je sledovací algoritmus

Tabulka 6.12: Úspěšnost sledování osob na jednotlivých snímcích

Metoda	TP	TN	FP	FN	Acc.	Prec.	Rec.
Detekce a úprava	5 018	25	407	114	90,64 %	92,50 %	97,78 %
Klasifikace	5 049	183	133	199	94,03 %	97,43 %	96,21 %
Sledování	5 116	229	78	141	96,06 %	98,50 %	97,31 %

přiřadil k sobě.

Pojmy pozitivní a negativní jsou zde definovány následovně:

Positive (P) – byla nalezena vazba mezi osobou na předchozím snímku a osobou na současném snímku

Negative (N) – vazba mezi osobami nebyla nalezena

Může se stát, že se osoba objeví pouze na jednom snímku a ihned odejde. Aby bylo takové případy možné detekovat, je vstup osoby do prostoru považován za vazbu mezi „neexistencí“ a aktuální polohou.

Mezi jednotlivými řešeními můžeme tedy najít následující čtyři jevy:

True positive (TP) – Vazba mezi osobami se nachází jak v referenčním řešení, tak ve výsledku testovaného řešení.

True negative (TN) – Snímek je v obou řešeních prázdný. Takto definované „true negative“ neodpovídá ostatním bodům a může tak vzniknout několik problémů: například neplatí, že algoritmus tipující existenci/neexistenci vazby náhodně bude mít úspěšnost vyšší než nebo alespoň přibližně rovnu rovnu 50 % – tato úspěšnost bude nejspíše mnohem nižší. Teoreticky by mělo být TN definováno jako stav, kdy neexistuje vazba mezi osobou na předchozím snímku a osobou na tou současném. Počet jednotlivých osob na snímcích však není znám a může se v jednotlivých řešeních lišit. Díky tomu by bylo možné zvýšit počet TN přidáním ve skutečnosti neexistující osoby (se spoustou správně označených neexistujících vazeb). Proto je TN definováno tímto způsobem.

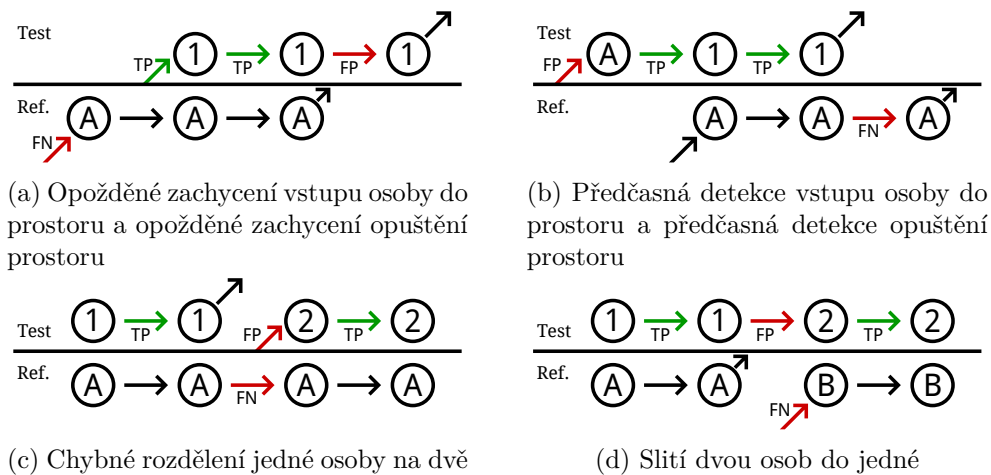
False positive (FP) – Mezi osobami byla nalezena vazba, která v referenčním řešení není.

False negative (FN) – V testovaném řešení chybí vazba nacházející se v referenčním řešení

6.7.3.1 Přiřazovací algoritmus

Výstupem sledovací části jsou osoby popsané pomocí polohy a identifikačního čísla. Shoda identifikačního čísla znamená, že mezi osobami je vazba (jedná se o tutéž osobu). Identifikační číslo jedné osoby v referenčním řešení a v testovacím setu samozřejmě nemusí být shodné. Proto je nutné vytvořit algoritmus, který bude jednotlivá ID mezi sebou přiřazovat. Popisu takového algoritmu se věnuje tato sekce.

Mnou navržený algoritmus postupně prochází všechny snímky a každé osobě v obou řešeních přiřadí ID nějaké osoby v druhém řešení. Přiřazení je



Obrázek 6.2: Ukázka ohodnocení vybraných výsledků sledování. Horní části obrázků zobrazují výsledky sledovacího algoritmu, dolní část je referenční řešení. Šipky znázorňují přechody mezi osobami, šipka zespoda pak vstup osoby do prostoru a šipka nahoru opuštění prostoru.

prosté¹⁴ a symetrické¹⁵. V případě, že se přiřazení nepodaří (v druhém řešení není dostatek volných osob), považuje to algoritmus za chybu.

Přiřazení a ohodnocování na jednotlivých snímcích probíhá podle následujících pravidel:

1. Přiřazení již známých osob. Pokud má nějaká osoba z testovaného řešení přiřazeno ID osoby, která se nachází na stejném snímku v referenčním řešení, pak tuto osobu považujeme za správně přiřazenou (TP).
2. Přiřazení nových osob. Za nové osoby považujeme ty, které ještě nemají přiřazeno žádné ID. Tyto osoby jsou navzájem spárovány (ID jedné je přiřazeno druhé). Je-li takových osob více, spárují se ty, které mají mezi sebou nejmenší vzdálenost.

I v tomto případě se jedná o TP.

3. Spárování zbytku. Zbývají-li na snímku ještě nějaké osoby, pokusí se je algoritmus přiřadit navzájem. Nachází-li se v jednom z řešení nikdy nevyužitá osoba, je upřednostněna ta. Důvodem k tomu je možnost, že sledovací algoritmus chybně označil, že z prostoru jedna osoba odešla a další vstoupila (i když ve skutečnosti se jedná o tutéž osobu) nebo naopak (viz obr. 6.2c a 6.2d). Každá změna ID znamená zvýšení počtu FN, resp. FP, podle toho, kde se měněná osoba nachází. FN je změna ID referenční osoby, FP změna ID osoby z testovacího řešení. Není-li

¹⁴Každé osobě je přiřazeno maximálně jedno ID.

¹⁵Je-li osobě *A* přiřazen identifikátor *B*, pak i osobě *B* musí být přiřazen identifikátor *A*.

6. TESTOVÁNÍ

jasné, jak osoby spárovat, párují se vždy ty osoby, které mají mezi sebou nejnižší vzdálenost.

4. Všechny zbývající osoby jsou považovány za false positive nebo false negative, podle toho, zda se nacházejí v testovaném řešení nebo v tom referenčním.

6.7.4 Výsledky měření

Metrikou vazeb jsem měřil nejen samotný sledovací algoritmus, ale i jeho části. V následující tabulce jsou uvedeny výsledky sledovacího algoritmu za normální situace a dále výsledky s odstraněnými vybranými částmi. Díky tomu se dá odhadnout, jakým způsobem se jednotlivé části podílejí na výsledku. Toto číslo samozřejmě není zcela přesné, protože veškeré konstanty algoritmu jsem nastavoval tak, aby vracely co nejlepší výsledek s úplným algoritmem.

Je-li odstraněn Kalmanův filtr, pak je poloha sledované osoby stanovena tam, kde se nacházela při poslední detekci.

Po odstranění sledovacího klasifikátoru se třída, do které osoba/objekt náleží, počítá jako poslední výstup klasifikační fáze (bez ohledu na předešlé klasifikace).

Tabulka 6.13: Úspěšnost sledování osob na jednotlivých snímcích

Metoda	TP	TN	FP	FN	Acc.	Prec.	Rec.
Sledování	5 114	229	303	150	92,18 %	94,41 %	97,15 %
Odstraněn KF 3D souřadnic	5 108	226	307	156	92,01 %	94,33 %	97,04 %
Odstraněn KF souřadnic v obrázku	3 767	168	276	1 600	67,72 %	93,17 %	70,18 %
Odstraněn sledovací klasifikátor	5 067	198	337	199	90,76 %	93,76 %	96,22 %

Závěr

Cílem mé práce bylo nastudovat metody využívané pro detekci a sledování osob v kamerovém záznamu a poté vytvořit a otestovat algoritmus, který tuto úlohu řeší.

Nejdříve jsem se seznámil s literaturou, která se tímto tématem zabývá. Podobné existující algoritmy jsou popsány na začátku mé práce. Tyto algoritmy se zpravidla dají rozdělit do čtyř částí. Při návrhu vlastního algoritmu jsem toto rozdělení přejal a pro jednotlivé části algoritmu našel vhodné metody. Ty jsou popsány v další kapitole mé práce.

Pomocí těchto metod jsem navrhl a implementoval algoritmus řešící zadanou úlohu, který je popsán v předposlední kapitole mé práce.

Poslední kapitolou je pak testování. Algoritmus jsem spustil nad snímky, které byly zaznamenány během provozu obchodního domu. Jeho výsledek jsem porovnal s manuálně vytvořeným řešením.

Pomocí testování jsem vybral z dostupných metod ty, které problém řešily nejlépe. Testování také ukázalo, že algoritmus je schopen sledovat osoby v kamerovém záznamu.

Literatura

- [1] Umer, M.: Introduction to Microsoft Kinect. *Microsoft Pakistan DX Community Blog* [online]. 26. 1. 2013 [cit. 24. 2. 2017]. Dostupné z: <https://blogs.msdn.microsoft.com/pakistan/2013/01/26/part-1-introduction-to-microsoft-kinect>
- [2] Smíšek, J.; Jancosek, M.; Pajdla, T.: 3D with Kinect. Technická zpráva, FEE, Czech Technical University in Prague, 2011.
- [3] Fu, H.; Ma, H.; Xiao, H.: Scene-adaptive accurate and fast vertical crowd counting via joint using depth and color information. *Multimedia Tools and Applications*, ročník 73, č. 1, 2013: s. 273–289, doi: 10.1007/s11042-013-1608-4. Dostupné z: <http://dx.doi.org/10.1007/s11042-013-1608-4>
- [4] Stahlschmidt, C.; Gavriilidis, A.; Velten, J.; aj.: *Multimedia Communications, Services and Security: 6th International Conference, MCSS 2013, Krakow, Poland, June 6-7, 2013. Proceedings*, kapitola People Detection and Tracking from a Top-View Position Using a Time-of-Flight Camera. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ISBN 978-3-642-38559-9, s. 213–223, doi:10.1007/978-3-642-38559-9_19. Dostupné z: http://dx.doi.org/10.1007/978-3-642-38559-9_19
- [5] Tseng, T. E.; Liu, A. S.; Hsiao, P. H.; aj.: Real-time people detection and tracking for indoor surveillance using multiple top-view depth cameras. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, s. 4077–4082, doi: 10.1109/IROS.2014.6943136.
- [6] Galčík, F.; Gargalík, R.: Real-Time Depth Map Based People Counting. In *Advanced Concepts for Intelligent Vision Systems, Lecture Notes in Computer Science*, ročník 8192, editace J. Blanc-Talon; A. Kasinski; W. Philips; D. Popescu; P. Scheunders, Springer International Pu-

- blishing, 2013, ISBN 978-3-319-02894-1, s. 330–341, doi:10.1007/978-3-319-02895-8_30. Dostupné z: http://dx.doi.org/10.1007/978-3-319-02895-8_30
- [7] Brazey, D.; Gout, C.: An algorithm for automatic people detection from depth map sequences. In *Visual Information Processing (EUVIP), 2014 5th European Workshop on*, Dec 2014, s. 1–6, doi:10.1109/EUVIP.2014.7018368.
- [8] Zhang, X.; Yan, J.; Feng, S.; aj.: Water Filling: Unsupervised People Counting via Vertical Kinect Sensor. In *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, Sept 2012, s. 215–220, doi:10.1109/AVSS.2012.82. Dostupné z: <http://dx.doi.org/10.1109/AVSS.2012.82>
- [9] Pizzo, L. D.; Foggia, P.; Greco, A.; aj.: A versatile and effective method for counting people on either RGB or depth overhead cameras. In *Multimedia Expo Workshops (ICMEW), 2015 IEEE International Conference on*, June 2015, s. 1–6, doi:10.1109/ICMEW.2015.7169795.
- [10] Maggio, E.; Cavallaro, A.: *What is video tracking?*, kapitola 1. John Wiley & Sons, Ltd, 2010, ISBN 9780470974377, str. 1, doi:10.1002/9780470974377. Dostupné z: <http://dx.doi.org/10.1002/9780470974377>
- [11] Hlaváč, V.: *Matematická morfologie* [online]. České vysoké učení technické, Fakulta elektrotechnická [cit. 12. 4. 2017]. Dostupné z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/71-3MatMorpholBinCz.pdf>
- [12] Owens, R.: *Mathematical Morphology* [online]. University of Edinburgh, 1997 [cit. 12. 4. 2017]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT3/node3.html
- [13] Ftáčnik, M.; Škovierová, J.: Segmentácia. *Základy počítačovej grafiky a spracovania obrazu* [online] [cit. 24. 2. 2017]. Dostupné z: <http://sccg.sk/~ftacnik/ZPGSO-6-2016-S0.pptx>
- [14] Rauter, M.: Reliable Human Detection and Tracking in Top-View Depth Images. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, June 2013, s. 529–534, doi:10.1109/CVPRW.2013.84.
- [15] Comaniciu, D.; Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 24, č. 5, May 2002: s. 603–619, ISSN 0162-8828, doi:10.1109/34.1000236.

-
- [16] Coskun, A.; Kara, A.; Parlaktuna, M.; aj.: People counting system by using kinect sensor. In *Innovations in Intelligent SysTems and Applications (INISTA), 2015 International Symposium on*, Sept 2015, s. 1–7, doi:10.1109/INISTA.2015.7276740.
- [17] Tanner, R.; Studer, M.; Zanolli, A.; aj.: People Detection and Tracking with TOF Sensor. In *Advanced Video and Signal Based Surveillance, 2008. AVSS 08. IEEE Fifth International Conference on*, Sept 2008, s. 356–361, doi:10.1109/AVSS.2008.18.
- [18] Bevilacqua, A.; Stefano, L. D.; Azzari, P.: People Tracking Using a Time-of-Flight Depth Sensor. In *2006 IEEE International Conference on Video and Signal Based Surveillance*, Nov 2006, s. 89–89, doi:10.1109/AVSS.2006.92.
- [19] Álvarez Álvarez, P.; García-Ordás, M.; Fidalgo-Fernández, E.; aj.: Tracking and counting people from a zenithal view using RGB-D images and Kalman filter. In *Actas de las XXXVI Jornadas de Automática*, September 2015, s. 153–158.
- [20] Wateosot, C.; Suvonvorn, N.: Top-view Based People Counting Using Mixture of Depth and Color Information. *The Second Asian Conference on Information Systems*, 2013.
- [21] Schreiber, D.; Kriechbaum, A.; Rauter, M.: A multisensor surveillance system for Automated Border Control (eGate). In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, Aug 2013, s. 432–437, doi:10.1109/AVSS.2013.6636678.
- [22] Ikemura, S.; Fujiyoshi, H.: Human detection by Haar-like filtering using depth information. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, ISSN 1051-4651, s. 813–816.
- [23] Zhu, L.; Wong, K.-H.: Human Tracking and Counting Using the KINECT Range Sensor Based on Adaboost and Kalman Filter. In *Advances in Visual Computing: 9th International Symposium, ISVC 2013, Rethymnon, Crete, Greece, July 29-31, 2013. Proceedings, Part II*, 2013, s. 582–591, doi:10.1007/978-3-642-41939-3_57. Dostupné z: http://dx.doi.org/10.1007/978-3-642-41939-3_57
- [24] Malawski, F.: Top-view people counting in public transportation using Kinect. *Challenges of Modern Technology*, ročník 5, 2014: s. 17–20.
- [25] Vera, P.; Monjaraz, S.; Salas, J.: Counting pedestrians with a zenithal arrangement of depth cameras. *Machine Vision and Applications*, ročník 27, č. 2, 2015: s. 303–315, ISSN 1432-1769, doi:10.1007/s00138-015-0739-1. Dostupné z: <http://dx.doi.org/10.1007/s00138-015-0739-1>

- [26] Felzenszwalb, P. F.; Huttenlocher, D. P.: Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, ročník 59, č. 2, 2004: s. 167–181, ISSN 1573-1405, doi:10.1023/B:VISI.0000022288.19776.77. Dostupné z: <http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77>
- [27] Yahiaoui, T.; Meurie, C.; Khoudour, L.; aj.: A People Counting System Based on Dense and Close Stereovision. In *3rd International Conference on Image and Signal Processing, ICISP'2008, Lecture Notes in Computer Science*, ročník 5099, editace A. E. et al., Cherbourg-Octeville: Springer, 2008, s. 59–66, doi:10.1007/978-3-540-69905-7_7. Dostupné z: <https://hal.archives-ouvertes.fr/hal-00521106>
- [28] Beymer, D.: Person counting using stereo. In *Human Motion, 2000. Proceedings. Workshop on*, 2000, s. 127–133, doi:10.1109/HUMO.2000.897382.
- [29] Bagnell, D.: Occupancy Maps. *Statistical Techniques in Robotics* [online]. Carnegie Mellon University, 2011 [cit. 30. 4. 2017]. Dostupné z: http://www.cs.cmu.edu/~16831-f14/notes/F14/16831_lecture06_agiri_dmcconac_kumarsha_nbhakta.pdf
- [30] Antić, B.; Letić, D.; Čulibrk, D.; aj.: K-means Based Segmentation for Real-time Zenithal People Counting. In *Proceedings of the 16th IEEE International Conference on Image Processing, ICIP09, Piscataway, NJ, USA: IEEE Press, 2009, ISBN 978-1-4244-5653-6*, s. 2537–2540. Dostupné z: <http://dl.acm.org/citation.cfm?id=1819298.1819443>
- [31] Barandiaran, J.; Murguia, B.; Boto, F.: Real-Time People Counting Using Multiple Lines. *Image Analysis for Multimedia Interactive Services, International Workshop on*, ročník 0, 2008: s. 159–162, doi:10.1109/WIAMIS.2008.27.
- [32] Lucas, B. D.; Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'81, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981*, s. 674–679.
- [33] Mukherjee, S.; Saha, B.; Jamal, I.; aj.: Anovel [sic!] framework for automatic passenger counting. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept 2011, ISSN 1522-4880, s. 2969–2972, doi:10.1109/ICIP.2011.6116284.
- [34] Horn, B. K.; Schunck, B. G.: Determining Optical Flow. *Technická zpráva*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1980.

-
- [35] Dalal, N.; Triggs, B.: Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*, ročník 1, June 2005, ISSN 1063-6919, s. 886–893 vol. 1, doi:10.1109/CVPR.2005.177.
- [36] Zhu, Q.; Yeh, M.-C.; Cheng, K.-T.; aj.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06*, Washington, DC, USA: IEEE Computer Society, 2006, ISBN 0-7695-2597-0, s. 1491–1498, doi:10.1109/CVPR.2006.119. Dostupné z: <http://dx.doi.org/10.1109/CVPR.2006.119>
- [37] Xu, H.; Lv, P.; Meng, L.: A people counting system based on head-shoulder detection and tracking in surveillance video. In *2010 International Conference On Computer Design and Applications*, ročník 1, June 2010, s. V1–394–V1–398, doi:10.1109/ICCDA.2010.5540833.
- [38] Nakatani, R.; Kouno, D.; Shimada, K.; aj.: A Person Identification Method Using a Top-View Head Image from an Overhead Camera. *JACIII*, ročník 16, 2012: s. 696–703.
- [39] Hernandez, D.; Castrillon, M.; Lorenzo, J.: People counting with re-identification using depth cameras. In *Imaging for Crime Detection and Prevention 2011 (ICDP 2011), 4th International Conference on*, Nov 2011, s. 1–6, doi:10.1049/ic.2011.0113.
- [40] Dan, B. K.; Kim, Y. S.; Suryanto; aj.: Robust people counting system based on sensor fusion. *IEEE Transactions on Consumer Electronics*, ročník 58, č. 3, August 2012: s. 1013–1021, ISSN 0098-3063, doi:10.1109/TCE.2012.6311350.
- [41] Springer Verlag GmbH, European Mathematical Society: Bhattacharyya distance. Kluwer Academic Publishers, 2002. Dostupné z: http://www.encyclopediaofmath.org/index.php?title=Bhattacharyya_distance&oldid=15124
- [42] Spinello, L.; Arras, K. O.: People detection in RGB-D data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, ISSN 2153-0858, s. 3838–3843, doi:10.1109/IROS.2011.6095074.
- [43] Clouard, R.: Tutorial: Mathematical Morphology. *Pantheon Project* [online]. University of Caen, 2012 [cit. 7. 2. 2017]. Dostupné z: <https://clouard.users.greyc.fr/Pantheon/experiments/morphology/index-en.html>

- [44] Petrovičová, A.: *Využitie maďarskej metódy pri riešení priradovacieho problému*. Bakalárska práca, Vysoká škola ekonomická v Praze, 2014.
- [45] Kosková, I.: *Distribuční úlohy I*. Česká zemědělská univerzita v Praze, 2004.
- [46] Kuhn, H. W.; Yaw, B.: The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, 1955: s. 83–97.
- [47] Welch, G.; Bishop, G.: An Introduction to the Kalman Filter. Technická zpráva, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.
- [48] Bazzani, L.; Bloisi, D.; Murino, V.: A Comparison of Multi Hypothesis Kalman Filter and Particle Filter for Multi-target Tracking. In *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, 2009, s. 47–54.
- [49] Maybeck, P. S.: *Stochastic models, estimation, and control*. Mathematics in Science and Engineering, New York: Academic Press, 1979, ISBN 0-12-480701-1. Dostupné z: https://www.cs.unc.edu/~welch/kalman/media/pdf/maybeck_ch1.pdf
- [50] Havlena, V.: *Odhadování, filtrace a detekce* [online]. České vysoké učení technické, Fakulta elektrotechnická [cit. 19. 4. 2017]. Dostupné z: https://moodle.dce.fel.cvut.cz/pluginfile.php/6198/mod_page/content/22/0FD_slides_1up.pdf
- [51] Roerdink, J. B.; Meijster, A.: The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundam. Inf.*, ročník 41, č. 1,2, Duben 2000: s. 187–228, ISSN 0169-2968. Dostupné z: <http://dl.acm.org/citation.cfm?id=2372488.2372495>
- [52] Kuhn, H. W.: The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, ročník 2, č. 1-2, 1955: s. 83–97, ISSN 1931-9193, doi:10.1002/nav.3800020109. Dostupné z: <http://dx.doi.org/10.1002/nav.3800020109>
- [53] Stern, K.: *HungarianAlgorithm* [software] [cit. 2. 12. 2016]. Dostupné z: https://github.com/KevinStern/software-and-algorithms/blob/master/src/main/java/blogspot/software_and_algorithms/stern_library/optimization/HungarianAlgorithm.java

Použité značení

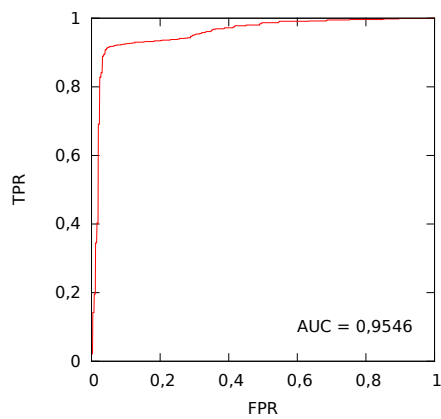
Následující seznam obsahuje vybrané používané proměnné a jejich význam.

- x Horizontální poloha bodu v hloubkové mapě nebo v barevném snímku, může značit i horizontální osu
- y Vertikální poloha bodu v hloubkové mapě nebo v barevném snímku, případně vertikální osa
- x_P Poloha bodu v 3D modelu na ose x
- y_P Poloha bodu v 3D modelu na ose y
- d Hloubková mapa
- d_{min} Minimální možný průměr hlavy
- h Výška
- g Upravená hloubková mapa pomocí algoritmu water filling
- τ Práh, předem stanovená konstanta
- f Ohnisková vzdálenost
- σ Směrodatná odchylka
- σ^2 Rozptyl
- k Počet centroidů v algoritmu k-means
- $\delta_{avg}(R)$ Průměrná výška v oblasti R
- S Obsah oblasti
- R Oblast, množina pixelů
- i, j, k, m, n Indexy či jiné proměnné, význam se může lišit

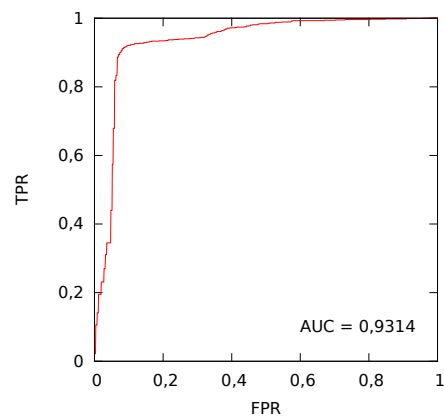
Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src	
├ impl	Java knihovna obsahující implementaci algoritmů
├ thesis	zdrojová forma práce ve formátu \LaTeX
├ └ images	obrázky použité v práci, formát PDF
├ └ └ ROC	ROC křivky testovaných metod
text	text práce
├ thesis.pdf	text práce ve formátu PDF
rec	ukázkový vstupní záznam
├ vga	barevné snímky
├ depth	hloubkové mapy
video	ukázky funkce algoritmu v různých situacích ve formátu OGV
└ literature	použitá literatura

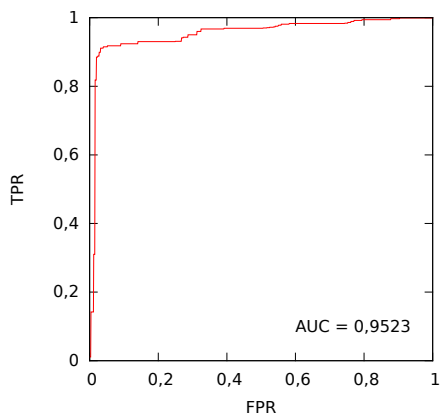
ROC křivky detekčních metod



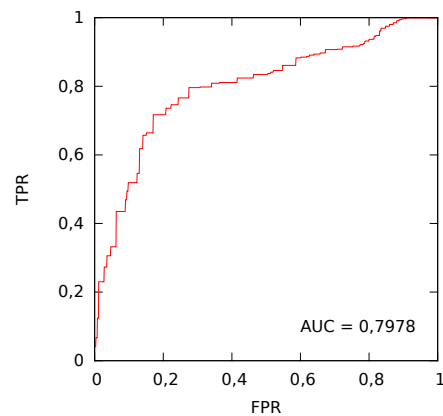
(a) Potlačení nemaxim



(b) Potl. nemaxim (naivní)

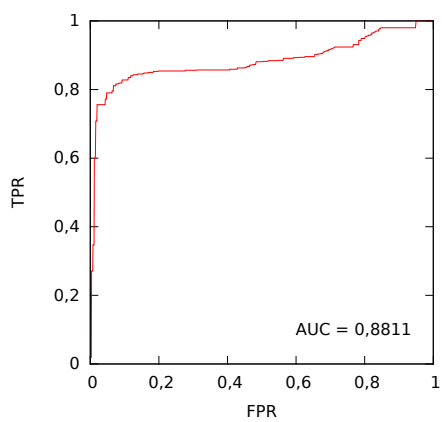


(c) Extended h-maxima

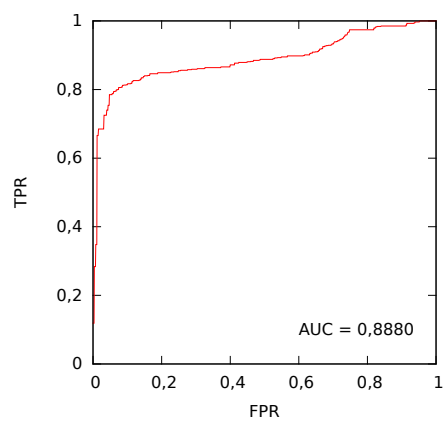


(d) Hill Climbing

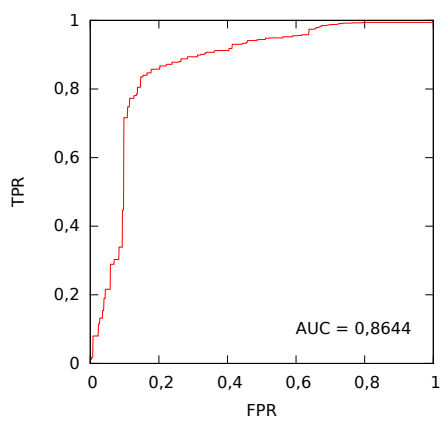
C. ROC KŘIVKY DETEKČNÍCH METOD



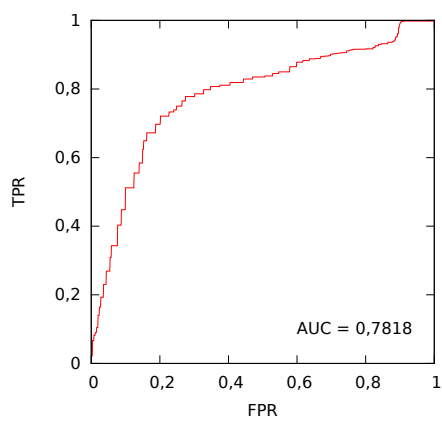
(e) K-means



(f) Weighted k-means



(g) Water filling



(h) Narůstání oblastí