

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE POHYBLIVÉHO OBJEKTU VE VIDEU

BAKALÁŘSKÁ PRÁCE

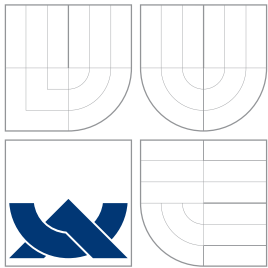
BACHELOR'S THESIS

AUTOR PRÁCE

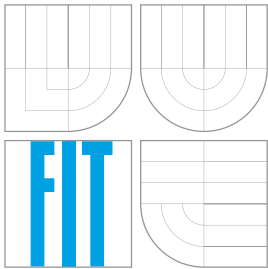
AUTHOR

RADIM BUCHTELA

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE POHYBLIVÉHO OBJEKTU VE VIDEU

MOVING OBJECT DETECTION IN VIDEO

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADIM BUCHTELA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ADAM HEROUT, Ph.D.

BRNO 2010

Abstrakt

Práce se věnuje oblasti detekce pohyblivých objektů ve video sekvenci. V práci jsou detailně popsány detekční metody založené na principu odečítání pozadí (Background subtraction). Jsou zde představeny strategie modelování pozadí a aktualizace modelu pozadí. Práce se také částečně zabývá problematikou sledování objektů ve videu, kde je představen jeden z možných přístupů. Součástí práce je aplikace pro demonstrování činnosti detekčních metod a sledovací metody. Na výsledcích jsou ukázány hlavní přednosti a zápory naimplementovaných metod.

Abstract

This thesis is devoted to detection of moving object in video sequence. In this thesis are described in detail detection methods based on principle of Background subtraction. There are introduced strategies of background modeling and background model updating. This thesis deal also with problems of tracking objects in video, there is introduced one of possible approaches. Part of this thesis is application for function demonstration of detection methods and tracking method. On results are showed main advantages and disadvantages of implemented methods.

Klíčová slova

detekce pohyblivého objektu, počítačové vidění, odečítání pozadí, modelování pozadí, aktualizace pozadí, detekce popředí, sledování objektů

Keywords

detection of moving object, computer vision, Background subtraction, background modeling, background update, foreground detection, object tracking

Citace

Radim Buchtela: Detekce pohyblivého objektu ve videu, bakalářská práce, Brno, FIT VUT v Brně, 2010

Detekce pohyblivého objektu ve videu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Adama Herouta, Ph.D.. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Radim Buchtela
17. května 2010

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Adamovi Heroutovi, Ph.D. za odborné vedení, jeho ochotu a cenné připomínky. Dále bych chtěl poděkovat své rodině a přátelům za podporu a zázemí.

© Radim Buchtela, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Základy zpracování obrazu	5
2.1 Obraz a jeho reprezentace	5
2.2 Barevný model	6
2.3 Obrazové transformace	6
2.3.1 Převod do stupně šedi	6
2.3.2 Prahování	7
2.3.3 Eroze	7
2.3.4 Dilatace	7
2.3.5 Medián	7
3 Metoda Odečítání pozadí (Background subtraction)	8
3.1 Předzpracování	9
3.2 Detekce popředí	9
3.3 Modelování pozadí	10
3.3.1 Nerekurzivní techniky	10
3.3.2 Rekurzivní techniky	12
3.3.3 Citlivost výběru (Selectivity)	16
3.4 Ověření dat	17
4 Sledování objektu (Object tracking)	19
4.1 Formulace problému	19
4.2 Porovnání blobů	19
4.3 Přiřazení objektu k blobu	21
5 Návrh a implementace testovací aplikace	23
5.1 Návrh aplikace	23
5.2 Implementace	26
5.2.1 Třídy <code>BS_detector</code> a <code>Tracker</code>	26
5.2.2 Knihovny <code>OpenCV</code> , <code>cvBlob</code> a <code>TinyXML</code>	27
5.3 Ovládání aplikace	27
6 Testování	29
6.1 Hodnota detekčního prahu	29
6.2 Hodnota učící konstanty	31
6.3 Detekce pohyblivých objektů ve venkovní scéně	31
6.4 Detekce pohyblivých objektů ve vnitřní scéně	32

6.5	Sledování pohyblivých objektů	32
6.6	Výpočetní náročnost	33
7	Závěr	34
A	Obsah přiloženého DVD	37

Kapitola 1

Úvod

První snahy o detekci pohybu v obraze se objevují v průběhu 70. let, k jeho prudkému rozvoji však dochází v posledních dvou desetiletí a stává se jedním z typických úkolů vědní disciplíny – *počítačové vidění* (computer vision). Detekce pohyblivého tělesa ve video sekvenci bývá prvním důležitým krokem k extrakci informace v mnoha studiích zabývajících se počítačovým viděním, jako je například monitorování dopravního provozu, sledování lidí, analýza pohybu na vymezené ploše a tvorba statistik, bezpečnostní videa sledující kriminalitu, vandalismus či pohyb na vymezeném prostoru. Kvalita výsledků získaných v tomto kroku se promítne v podobě přesnějšího tvaru a pozice detekovaného objektu.

Základním úkolem detekce pohybu je oddělení popředí, tj. pohybujícího se tělesa, od pozadí a získání jeho přesných hranic. Proces detekce je často komplikován přítomností řadou nežádoucích vlivů, obzvláště pokud pracujeme s reálnou scénou. Mezi tyto vlivy patří šum, který je obsažen v každém snímku video sekvence v náhodných rozpoloženích, změna modelu pozadí (přeparkování vozidla), nestacionární objekty pozadí (kymácející se stromy, vlnící se vodní hladina) či změna obrazu způsobená chvěním kamery, která je běžná v aplikacích pracujících s venkovní scénou zatíženou větrem nebo vibracemi půdy. Pro detekční systém jsou také velmi obtížně detekovatelné objekty popředí mající podobnou barvu jako pozadí či objekty popředí, které znehybněly a není je možné odlišit od objektů pozadí.

Cílem této bakalářské práce je snaha prostudovat strategie detekce pohyblivých objektů ve video sekvenci snímané stacionární kamerou. Přičemž důraz bude kladen na možnost nasazení detektoru na reálnou scénu, což předpokládá určitou schopnost adaptace pro různé nežádoucí vlivy. Při posuzování účinnosti detekčního algoritmu se především zaměříme 4 hlavní hodnotící faktory: přesnost v detekci hranic objektu (přesnost), stabilita detekce v čase (časová stabilita), schopnost detekovat změny malých amplitud (citlivost) a schopnost poskytnutí vysoké přesnosti pod měnícími se podmínkami (robustnost).

Přístupů k problematice detekce pohybu existuje mnoho. Mezi nejznámější metody se řadí: metoda *Porovnání hran*, *Porovnání histogramů*, *Optical flow*, *Local Binary Patterns* či jejich kombinace. A však jedním z nejvíce frekventovaných přístupů se statickou kamerou je založen na metodě *Odečítání pozadí* (*Background subtraction*) a jejích variantách. Které stojí v čele pozornosti této bakalářské práce.

Práce je rozčleněna do sedmi kapitol. V 2. kapitole jsou popsány základní obrazové transformace využívané v různých fázích detekčního procesu. Je zde také věnována pozornost problematice získání digitálního obrazu a jeho vhodné reprezentace.

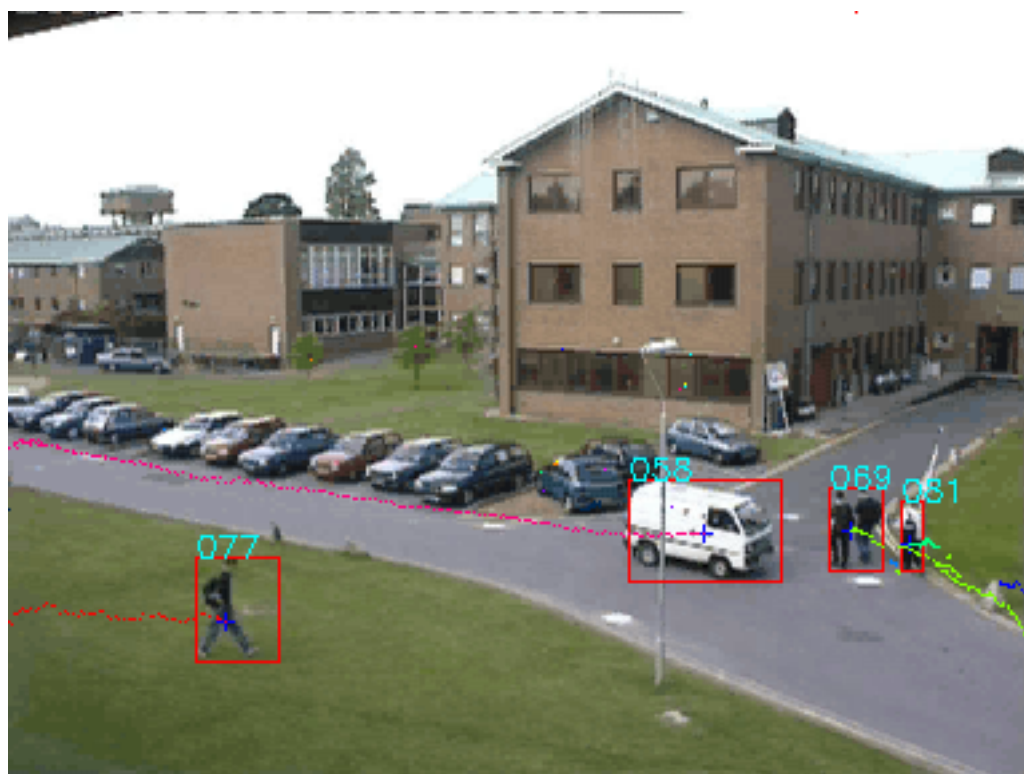
Kapitola 3 se zabývá popisem jednotlivých technik detekce pohyblivých objektů na základě odečítání pozadí. Jsou zde rozebrány jednotlivé fáze, kterými detekční metody

při svém výpočtu procházejí. Přitom největší pozornost byla věnována na fázi modelování pozadí a detekce popředí.

Kapitola 4 popisuje jeden z možných přístupů sledování objektů ve video sekvenci, u něhož byl kladen důraz na možnost sledování více objektů v jeden okamžik.

Kapitola 5 se zabývá návrhem testovací aplikace, která slouží k demonstrování činnosti některých zde popsaných metod detekce pohybu a jedné sledovací metody. Je zde také popsán konkrétní způsob implementace a použité technologie.

V 6. kapitole je předvedena účinnost detekce jednotlivých metod na sadě testovacích video sekvencí. Velká část testování byla zaměřena na vliv změny klíčových parametrů detekčních algoritmů na úspěšnost detekce.



Obrázek 1.1: Příklad detekce pohyblivých objektů

Kapitola 2

Základy zpracování obrazu

Zpracování obrazu je rozsáhlá oblast stojící na rozhraní přinejmenším dvou vědních disciplín. Přibližuje se k počítačové grafice, která se zabývá tvorbou, úpravou a analýzou obrazových dat. Druhou oblastí je teorie signálů, ze které využívá oblast zabývající se popisem přechodu od spojitého obrazu k diskrétnímu.

Obrazová data jsou nejčastěji reprezentována pomocí rastru. Rastr je pravidelná síť obrazových bodů (pixelů), organizovaná jako dvourozměrná matice bodů. Každý bod nese specifickou informaci o své barvě popř. další doplňující informace. Druhým způsobem reprezentace dat je pomocí vektoru, kde jsou data popsána matematickou funkcí.

V našem případě se dále budeme zabývat jen rastrovou grafikou, jelikož na tomto způsobu reprezentace obrazových dat pracuje většina snímacích přístrojů. Kvalita záznamu je určena především parametrem rozlišení a barevná hloubka. Rozlišení udává celkový počet bodů obrazu nejčastěji ve formě *pocet_sloupcu* \times *pocet_radku*. Barevná hloubka udává počet bitů použitých k popisu barvy obrazového bodu. Informace o digitalizaci obrazu a jeho reprezentace popsané v této kapitole jsou čerpány z knihy od J. Žáry a spol. [14].

2.1 Obraz a jeho reprezentace

Naším cílem je získat obraz reálného světa do digitální podoby, která je vhodná pro uložení v paměti počítače. Tento proces se nazývá digitalizace. Je to přechod od spojitého obrazu popsaného spojitou funkcí $f(x, y)$ k diskrétnímu obrazu popsaného diskrétní funkcí $I_{i,j}$, kde x, y jsou souřadnice v prostoru, $x, y \in (-\infty, \infty)$, obecně se však budeme věnovat obrazové funkci mapující jen určitou oblast prostoru, $x \in \langle X_{min}, X_{max} \rangle$, $y \in \langle Y_{min}, Y_{max} \rangle$, oborem hodnot H obrazové funkce je libovolné reálné číslo vyjadřující velikost jasové intenzity světla, $H = (-\infty, \infty)$, i, j jsou souřadnice bodu v matici o rozměrech $I \times J$, $i \in \{0, 1, \dots, I-1\}$, $j \in \{0, 1, \dots, J-1\}$, a oborem hodnot K diskrétní funkce jsou hodnoty jasových úrovní, $K = \{0, 1, \dots, K_{max}\}$. Při digitalizaci převádíme nekonečné množství bodů a nekonečný počet barev na obraz o konečném množství bodů a o konečném množství barev dle zvolené vzorkovací funkce. Digitalizace probíhá ve dvou nezávislých krocích, jimiž jsou kvantování a vzorkování.

Principem kvantování je diskretizace oboru hodnot obrazové funkce $f(x, y)$. Obor hodnot je rozdělen na intervaly, jimž je přidělena jediná, zástupná hodnota. Její hodnotou nejčastěji bývá průměr ze zvoleného intervalu. Při kvantování dochází ke ztrátě informace z důvodu nahrazení množiny hodnot jedinou hodnotou. Ztrátu informace můžeme spatřovat v podobě skokových změn u původně hladkých přechodů. Proto při kvantování musí

být počet kvantovacích úrovní dostatečně velký, aby byly přesně vyjádřeny jemné detaily obrazu, nevznikaly falešné obrysy a aby se citlivost zařízení blížila citlivosti lidského oka.

Vzorkováním spojité funkce $f(x, y)$ se rozumí zaznamenávání hodnot - vzorků, v předem daných intervalech, do matice $I \times J$. Čím jemnější je vzorkování (čím větší matice $I \times J$) a kvantování, tím lépe je aproximován původní spojitý obrazový signál. Vzorkování tedy určuje výsledné rozlišení obrazu, jež nám přímo úměrně udává, kolik informací bude obraz obsahovat.

2.2 Barevný model

Barevný model popisuje základní barvy a způsob míchání těchto základních barev do výsledné barvy. Barva je v přírodě dána směsí světla různých vlnových délek, které se barevné modely snaží co nejvěrněji napodobit. V praxi se používají modely, u kterých je zvolen vhodný kompromis mezi přesností podání barevného dojmu a složitostí konkrétního modelu. Mezi nejčastější používané modely patří RGB, CMY, HSV, YUV. Zdroj [1].

V praxi je nejpoužívanějším barevným model RGB. Model RGB se skládá ze tří základních barev červená (R, red), zelená (G, green) a modrá (B, blue). Pro dosažení dalších barev se používá aditivní míchání intenzit těchto tří barev. Počet barevných odstínů závisí na počtu kódovacích bitů. V současné době se nejběžněji využívá vyjádření barevných složek pomocí 3 bytů, kde pro každou složku R, G, B je vyhrazeno 8bitů. Toto kódování dovoluje vyjádřit $2^8 \cdot 2^8 \cdot 2^8 = 16\,777\,216$ různých barevných odstínů.

2.3 Obrazové transformace

Výstup každého detekčního algoritmu je obecně zatížen šumem, obzvláště pokud pracujeme s reálnými scénami. Příčinami šumu je primárně vnitřní šum samotného snímacího přístroje, nežádoucí odrazy, objekty, jejichž barva se částečně či plně shoduje s pozadím, náhodné stíny a změny v osvětlení, ať již přírodní či umělé. K potlačení šumu využíváme různě složité obrazové transformace. Další neméně důležitou oblastí je transformace vstupního snímku do vhodnější formy pro další zpracování, která může znamenat významný posun ke zlepšení výkonu detekčního algoritmu.

2.3.1 Převod do stupně šedi

Při této transformaci dochází k převodu barevné informace na jednu odpovídající hodnotu ve stupni šedi. Odstín šedi je vyjádřen pomocí 8bitového čísla, tudíž v šedotónovém obrazu může každý obrazový bod nabývat 256 různých úrovní intenzity. Intenzita je postupně vypočtena pro každý bod barevného obrazu podle vztahu:

$$I = 0,299R + 0,587G + 0,144B \quad (2.1)$$

kde R je hodnota červené složky, G je hodnota zelené složky a B je hodnota modré složky.

Tento vztah vznikl empiricky s přihlédnutím na specifické vlastnosti lidského oka, které vnímá různým způsobem intenzitu jednotlivých barevných složek, přičemž nejcitlivější je na zelenou barvu, což je patrné ze vzorce, kde se na výsledné hodnotě podílí největší vahou.

Převod do stupně šedi se v problematice detekce pohyblivého tělesa v obraze úspěšně využívá k urychlení detekčního algoritmu, jelikož nemusíme procházet tři barevné složky každého obrazového bodu, ale postačí nám pouze jeden průchod.

2.3.2 Prahování

Operace prahování spočívá v převedení obrazu o více jasových úrovních na obraz, ve kterém se vyskytují pouze dvě jasové úrovně, A a B . Výsledný obraz má většinou binární charakter, $A = 0$, $B = 1$, a je dán vztahem:

$$f(x) = \begin{cases} A & , \text{je-li } x < \tau \\ B & , \text{je-li } x \geq \tau \end{cases} \quad (2.2)$$

kde x vstupní jasová hodnota, τ je zvolený práh, A a B jsou výstupní hodnoty. Je-li vstupní hodnota x menší než zvolený práh τ , je nahrazena novou hodnotou A . Je-li vstupní hodnota x obrazového bodu větší nebo rovna hodnotě prahu τ , je nahrazen hodnotou B . Vhodná volba prahu má zásadní vliv na výslednou kvalitu obrazu.

V problematice detekce pohybujícího se tělesa ve videu je metoda prahování nejčastěji aplikována na výsledek odečítání pozadí od aktuálního snímku video sekvence, kdy chceme pročistit místa s minimálními hodnotami rozdílů.

2.3.3 Eroze

Eroze pracuje nad monochromatickým obrazem, tak že prochází každý bod obrazu a v jeho okolí vyhledává nejmenší hodnotu a tou je nahrazena hodnota aktuálního bodu. Za okolí bývá nejčastěji volena matice o rozměrech 3×3 .

Eroze se využívá pro odstranění osamocených bodů, které bývají zapříčeny šumem. Použití eroze má i svou negativní stránku a tou je narušení tvaru samotného objektu po jeho obvodu. To však lze částečně navrátit aplikováním dilatace.

2.3.4 Dilatace

Dilatace pracuje na stejném principu jako eroze, jen s tím rozdílem, že při procházení každého bodu nevyhledává v jeho okolí nejmenší, nýbrž největší hodnotu, kterou je nahrazena hodnota aktuálního bodu. Za okolí bývá nejčastěji volena matice o rozměrech 3×3 .

Aplikování dilatace se projevuje růstem objektů a zaplněním děr a úzkých zálivů uvnitř objektu. Dilataci bývá zvykem aplikovat po erozi.

2.3.5 Medián

Medián je střední cestou mezi použitím eroze a dilatace. Zde se v okolí každého bodu vyhledává střední hodnota, kterou je nahrazena aktuální hodnota bodu.

Při aplikování mediánu dochází k nárůstu či úbytku objektu v závislosti na hustotě jeho bodů v dané oblasti.

Kapitola 3

Metoda Odečítání pozadí (Background subtraction)

Metoda Odečítání pozadí je jednou z neznámějších a nejvyužívanějších metod v oblasti detekce pohybu v obraze. Mezi její hlavní přednosti patří rychlost a snadnost implementace. Díky těmto přednostem se metoda těší trvalé popularitě a snah stálého vylepšování. V současné době existuje již několik set variant o různé složitosti.

Princip metody ční v oddělení popředí od pozadí na základě odečítání, bod po bodu, aktuálního snímku od pozadí, které je vzato jako referenční snímek. Na podobném principu pracuje i lidský mozek, kdy hledáme rozdíly mezi obrázky na základě jejich porovnávání.

Třebaže existuje velké množství algoritmů založených na odečítání pozadí, u většiny z nich můžeme pozorovat stejné fáze, kterými při svém výpočtu procházejí. V práci [6] jsou popsány jednoduchým diagram datových toků, který se skládá ze čtyř hlavních kroků: *předzpracování (pre-processing)* - úprava vstupního obrazu do vhodnější formy pro další zpracování, *modelování pozadí (background modeling* též *background maintenance)*, *detekce popředí (foreground detection)*, *ověření dat (post-processing* též *data validation)* - sloužící k eliminaci těch bodů, které nekorespondují s pohyblivým objektem.

Při analýze metod založených na principu odečítání pozadí se hlouběji zaměříme na atributy: detekce popředí a modelování pozadí. Než se však pustíme do samotného popisu, je potřeba si definovat sadu entit a vztahů, se kterými se budeme potýkat po zbytek kapitol. V práci [5] jsou popsány následovně:

- *Blob* – sada spojených bodů tvořící souvislou oblast.
- *Popředí (foreground)* – binární obraz získaný odečtením aktuálního snímku od modelu pozadí při aplikování prahové hodnoty. V tomto obrázku jsou teoreticky obsaženy pouze body spojené s reálným pohybujiící se objektem.
- *Pozadí (background)* – obraz, který zachycuje statickou scénu. To znamená bez přítomnosti pohybujiících se objektů.
- *Viditelný pohybujiící se objekt (moving visual object, MVO)* – sada spojených bodů popředí, které souvisí s reálným pohybujiícím se objektem.
- *Šum popředí (foreground noise, FN)* – sada bodů, které se chybně objevili v popředí, ale nesouvisí s reálným pohybujiícím se objektem.

- *Šum pozadí (background noise, BN)* – sada bodů, které se chybně neobjeví v popředí, ale souvisí s reálným pohybujícím se objektem. Tento typ šumu se objeví vždy, pokud se barva pohybujícího se objektu shoduje s barvou pozadí a práh není dostatečně citlivý, aby je správně oddělil. Všechny body, $b(x, y)$, patřící do tohoto regionu, r_{BN} , mají vlastnost: $|I_t(x, y) - B_t(x, y)| < \tau, \forall x, y, b(x, y) \in r_{BN}$ a τ zvolený práh.
- *Stín (shadow, Sh)* – typ šumu popředí, často doprovází reálné pohybující se objekty. Všechny body, $b(x, y)$, tohoto regionu, r_{Sh} , mají vlastnost: $(I_t(x, y) - B_t(x, y)) < 0 \forall x, y, b(x, y) \in r_{Sh}$.
- *Odras (reflection, Rf)* – typ šumu popředí, obecně se vyskytuje v menším měřítku než stín. Všechny body tohoto regionu, r_{Rf} , mají vlastnost: $(I_t(x, y) - B_t(x, y)) > 0 \forall x, y, b(x, y) \in r_{Rf}$.
- *Duch (ghost)* – typ šumu popředí. Objevuje se v místě, kde se nachází pohybující se objekt, který je znehybněn, nebo pokud je pohybující se objekt součástí pozadí a v daný okamžik se uvede v pohyb. Všechny body tohoto typu mají vlastnost: $|I_t(x, y) - B_t(x, y)|$ je dotatečně velký, aby byl bod klasifikován jako popředí.
- *Kolísání (fluctuation, Fl)* – typ šumu popředí. Fluktuací neboli také kolísáním hodnot se rozumí malé změny v osvětlení, které jsou zachyceny mezi dvěma následujícími snímky. Tyto změny jsou dány fyzikálními vlastnostmi optických senzorů u snímáčího zařízení. Pro všechny pixely tohoto regionu, r_{Fl} , platí: $(I_t(x, y) - B_t(x, y)) \approx 0, \forall x, y, b(x, y) \in r_{Fl}$.

3.1 Předzpracování

Cílem detekčního algoritmu je detekovat významné změny v průběhu video sekvence se zanedbáním běžných bezvýznamných změn. Abychom bezvýznamné změny dokázaly potlačit, k tomuto účelu se využívají různé druhy filtrů pro vyhlazení vstupního snímku. Touto technikou se dá redukovat velká část šumu obsaženého v obrazu. Filtry pro vyhlazení lze také úspěšně využít pro redukcii šumu produkovaného vnějším okolím, jako je sněžení či déšť. U systémů běžících v reálném čase se pro zlepšení výkonu sahá ke změně velikosti snímků, snímkovací frekvence videa nebo ještě častěji k převodu barevného spektra videa do stupně šedi. V současné době je však, čím dál tím populárnější zpracovávat snímky reprezentované v RGB nebo HSV barevném prostoru, které vykazují lepší výsledky u identifikace objektů v méně kontrastních lokacích.

3.2 Detekce popředí

V této etapě se snažíme porovnat vstupní snímek video sekvence s modelem pozadí a identifikovat kandidáty na popředí ze vstupního snímku. V této práci pro popis každého pixelu o pozici (x, y) daného snímku v čase t a s hodnotou pixelu např. skalární ve stupni šedi, $I_t^{(x,y)}$, nebo RGB vektoru (r, g, b) , $I_t^{(x,y)}(r, g, b)$, bude dále použita zjednodušená notace $M_t(x, y)$. Potom můžeme říci, že každý pixel je označen jako popředí, když:

$$|I_t(x, y) - B_t(x, y)| > \tau \quad (3.1)$$

kde $I_t(x, y)$ je obraz aktuálního snímku, $B_t(x, y)$ je model pozadí a τ je práh s předdefinovanou hodnotou. Výsledek detekce popředí závisí na zvolené hodnotě prahu. Pokud je

hodnota prahu velmi nízká, tak detekované popředí obsahuje i části pozadí. V opačném případě, nastavení příliš vysokého prahu, dochází pouze k částečné detekci popředí. Optimálně zvolená hodnota prahu závisí na typu scény a celkové kvalitě video sekvence.

V novějších studiích se setkáváme s tendencemi opouštět princip *globálního prahování* (Global Thresholding), tj. použití předem definované prahové hodnoty, která je aplikovaná na každý pixel. Výhodnější je použití principu *lokálního prahování* (Local Thresholding), tj. použití prahu, který je určen pro každý pixel zvlášť. Tato strategie je velmi výhodná, když různé regiony mají různé osvětlení, protože objekty, které se vyskytují v tmavých regionech se obecně nevystavují tak velkému šumu jako objekty ve světlejších regionech [11]. Globální prahová hodnota má často za následek zničení objektu, který vstoupil do tmavého regionu. Nutno však podotknout, že strategie lokálního prahování by byla téměř bezvýznamná, pokud by nebyla rozšířena o schopnost aktualizace prahové hodnoty v průběhu detekce. Lokální (adaptivní) prahování využívají sofistikovanější metody jako *Klouzavý Gaussův průměr* či *Směsice Gaussových křivek*.

3.3 Modelování pozadí

Modelování pozadí je druhou důležitou etapou každého algoritmu založeného na odečítání pozadí. Mnoho výzkumů je věnováno rozvoji modelu pozadí, který je odolný vůči změnám prostředí v pozadí, ale zároveň citlivý k identifikaci všech pohybujících se objektů. Modelování pozadí můžeme rozdělit podle dvou různých kritérií. Buď podle fáze, ve které dochází k výpočtu modelu pozadí, to je pak označováno jako statické a dynamické, nebo podle techniky výpočtu modelu pozadí, to je označováno jako nerekurzivní a rekurzivní [10, 6].

Statické modelování je modelování, při kterém je model pozadí předpočítán v učicí fázi, nejběžněji zprůměrováním několika po sobě jdoucích snímků, aby bylo docíleno co nejkvalitnější podoby. V průběhu samotné detekce se model pozadí již nikterak nemění, nepřepočítává se, což je výhodou v ušetření výpočetního výkonu. Tato technika však vyžaduje existenci sekvence snímků bez pohybujících se objektů. Další nevýhody jsou, že nedokáže reagovat na dlouho trvající změny scény a špatně reaguje na změny osvětlení ať již náhlých nebo pozvolných v rámci denního 24hodinového cyklu.

Při dynamickém (adaptivním) modelování je model pozadí po dobu běhu výpočtu aktualizován, což umožňuje lepší reakci na změny osvětlení a na dlouho trvající změny. To se však promítne ve složitosti, kdy musíme řešit otázku vhodného nastavení parametrů rychlosti učení. Při tomto druhu modelování pozadí se objevuje nový nežádoucí jev a tím je vznik duchů. Tento jev vzniká v místě obrazu, kde se nachází nehybný objekt, který je součástí pozadí, a v určitý moment se uvede v pohyb.

3.3.1 Nerekurzivní techniky

Nerekurzivní techniky modelují pozadí s využitím bufferu, v němž je uloženo N předchozích snímků a výsledná podoba pozadí je vypočítána podle aktuální variace snímků uvnitř bufferu. Nerekurzivní metody jsou vysoce adaptivní, jelikož nejsou závislé na historii mimo snímky v bufferu. Nevýhody ve využití bufferu mohou tkvět ve zvýšených nárocích na velikost paměti. Nejznámější techniky jsou popsány níže.

Rozdíl dvou snímků (Frame differencing)

Pravděpodobně nejjednodušší technika, při které je snímek t vzat jako model pozadí pro snímek v čase $t + 1$.

$$B_{t+1}(x, y) = Frame_t(x, y) \quad (3.2)$$

Tato technika je využívána v nenáročných aplikacích ať již ve smyslu nároků na kvalitu výsledků či výpočetních nároků. Technika je spíše využívána jako základ, prvotní stupeň, pro další experimentování. Technika je citlivá na šum a změny v osvětlení. Pokud pro porovnání aktuálního snímku je použit jen jeden předchozí snímek, má to za následek nedetekování vnitřních částí pohybujícího se objektu. Tento jev je patrnější pro velké pomalu se pohybující objekty. Dá se říci, že toto jeden z hlavních neduhů této metody.

Průměr posledních N snímků

Při této technice je obraz modelu pozadí spočten jako průměr všech N snímků uvnitř bufferu:

$$B_{t+1}(x, y) = \frac{1}{N} \sum_{i=t-N+1}^t Frame_i(x, y)^1 \quad (3.3)$$

Smyslem této metody je vypočtení pozadí, které se blíží statické scéně aktuálního snímku. Tato metoda je odolná vůči šumu a malým pozvolným změnám v osvětlení. Vhodné nasazení této metody by bylo na vnitřní (indoorové) scény. Zde si je nutné uvědomit, že velikost bufferu, N , bude hrát zásadní roli ve schopnostech detektoru. Čím bude velikost bufferu větší, tím získáme vyšší odolnost modelu pozadí vůči vnějším jevům, jako jsou šum a pomalu se pohybující objekty. Naopak čím bude velikost bufferu menší, tím získáme vyšší adaptivitu systému, tj. schopnost rychle se přizpůsobit světelným změnám ve scéně. Za přijatelný kompromis mezi časovou stabilitou a adaptivitou je uváděna velikost bufferu obvykle okolo 5 snímků. Zřejmý problém s touto technikou je, že na výpočtu modelu pozadí se mimo pozadí podílí i popředí. Pokud se ve scéně vyskytují pomalu se pohybující objekty, metoda v tomto místě selhává.

Medián posledních N snímků

Při této technice je obraz modelu pozadí spočten jako medián všech N snímků uvnitř bufferu. To předpokládá, že hodnota pixelu zůstane v pozadí, pokud se vyskytuje ve více než polovině snímků v bufferu. Tato metoda je odolná vůči šumu, jelikož není ovlivňována extrémními hodnotami. Oproti průměrování medián do modelu pozadí přidává „prvek určitosti“, jelikož pixel pozadí je zde označen, buď jako jeden či druhý objekt, nikoliv jejich průměr. Příkladem může být scéna, ve které se nachází keř v čase t , $t + 1$, v čase $t + 2$ do scény vstoupí pes a zdržuje se zde po dobu $t + 3 + n$. Buffer, pro jeden konkrétní pixel, vykreslující naši situaci může vypadat následovně $\{13, 11, 203, 202, 205\}$ ², kde první dvě hodnoty se vztahují k prvnímu objektu (keř) a poslední tři hodnoty se vztahují k druhému objektu (pes). V tomto případě jako hodnota pozadí bude vybráno číslo 202, které jasně

¹uvedený vzorec lze zapsat i v rekurzivní podobě: $B_{t+1}(x, y) = B_t(x, y) + \frac{1}{N}(Frame_t - Frame_{t-N+1})$, kde N je velikost bufferu, $Frame_t$ aktuální snímek a $Frame_{t-N+1}$ je nejstarší snímek v bufferu. Ač metoda má i vzorec v rekurzivní podobě, tak stále spadá do kategorie Nerekurzivní techniky, jelikož při svém výpočtu stále potřebuje buffer o N snímcích. Vzorec v rekurzivní podobě byl uveden pro potřeby efektivnější implementace.

²hodnoty v bufferu jsou zvoleny záměrně mírně odlišné, což simuluje přítomnost šumu. V ideálním případě by buffer vypadal názorně $\{13, 13, 203, 203, 203\}$

deklaruje přítomnost psa. V případě průměrování by hodnota pixelu pozadí byla 126,8, což lze s těžší interpretovat.

V zásadě zde nejde o nic jiného, než že se snažíme odhadnout hodnotu pozadí pro následující snímek. K tomu se používají právě výše zmíněné techniky, někdy taky označované jako filtry. Jejichž úkolem je odhadnutí hodnoty pozadí a odfiltrování vnějších jevů (šum, změny osvětlení, pohybující se objekty), které ovlivňují naměřené hodnoty pozadí. Každý nový snímek nám dává novou informaci o hodnotě pozadí, což lze využít pro přesnější odhad hodnoty pozadí a získání informace o kvalitě tohoto odhadu. Nejjednodušší situace nastává, když víme, že hodnota pozadí má konstantní hodnotu (tj. s časem se nemění). V tomto případě se dá s úspěchem použít Průměr či Medián. Bohužel v reálné scéně s tímto předpokladem nemůžeme počítat. Proto byly zavedeny modifikované verze metody Průměru a Mediánu, které odhadují hodnotu pozadí vždy jen přes n -posledních snímků. Tento přístup je úspěšně aplikován pro odhad pozadí, jehož hodnota se v čase zpozdolna mění. Pokud však pozadí má tendenci k rychle se měnícím hodnotám, bývají nasazeny i propracovanější filtry jako je *Wienerův filtr* navržený Norbertem Wienerem publikovaný v roce 1949 nebo často využívaný *Kalmanův filtr*, který se však řadí již do druhé kategorie.

3.3.2 Rekurzivní techniky

Rekurzivní techniky při výpočtu modelu pozadí nevyužívají buffer. Místo toho rekurzivně aktualizují model pozadí pro každý snímek. Evidentní výhody těchto technik jsou v nižších paměťových nárocích.

Klouzavý průměr (Running average)

Model pozadí je při této technice rekurzivně aktualizován dle vztahu:

$$B_{t+1}(x, y) = \alpha I_t(x, y) + (1 - \alpha)B_t(x, y) \quad (3.4)$$

kde $\alpha \in (0,0; 1,0)$ je učicí konstanta³, která je velmi malá, typicky $\alpha \in (0,005; 0,1)$. Určuje, jakou měrou se na výsledné podobě modelu pozadí, $B_{t+1}(x, y)$, bude podílet vstupní snímek, $I_t(x, y)$. Čím je učicí konstanta větší, tím se bude aktuální snímek více podílet na podobě modelu pozadí. Tím lze dosáhnout rychlejší adaptivity systému. Na druhou stranu, čím je učicí konstanta menší, tím bude model pozadí stabilnější a méně náchylnější na změny ve scéně (šum, pohybující se objekty). Tento jednoduchý filtr je použit, buď jako hlavní hnací motor pro modelování pozadí, nebo bývá často využíván pro aktualizaci jen určité složky komplexnějšího modelu pozadí, jako je např. SG, MoG.

Klouzavý Gaussův průměr (Running Gaussian average)

Tato technika je založena na jedné Gaussově křivce zobrazující normální (Gaussovo) rozložení pravděpodobnosti $\eta(\mu, \sigma^2)$ hodnoty pozadí, kde složky μ a σ^2 jsou aktualizovány pomocí klouzavého průměru:

$$\mu_{t+1}(x, y) = \alpha I_t(x, y) + (1 - \alpha)\mu_t(x, y) \quad (3.5)$$

$$\sigma_{t+1}^2(x, y) = \alpha(I_t(x, y) - \mu_t(x, y))^2 + (1 - \alpha)\sigma_t^2(x, y) \quad (3.6)$$

³často se také můžeme setkat s ekvivalentní podobou: $B_{t+1}(x, y) = B_t(x, y) + \alpha(I_t(x, y) - B_t(x, y))$

kde μ (mí) je střední hodnota modelu pozadí, σ^2 (sigma) je rozptyl střední hodnoty. Rozptyl (též střední kvadratická odchylka, variance nebo také disperze) vyjadřuje, jak moc kolísá hodnota pozadí kolem střední hodnoty. Potom pro detekci popředí můžeme využít vzorec (3.1) v modifikované podobě:

$$|I_t(x, y) - \mu_t(x, y)| > k\sigma \quad (3.7)$$

kde k je konstanta, obvykle $k = 2,5$. Hodnota 2,5 nám z charakteristiky Gaussovy křivky říká, že hodnoty korespondující s pozadím o střední hodnotě μ a rozptylu σ z 99% leží v intervalu $(\mu - 2,5\sigma; \mu + 2,5\sigma)$ a tím bude správně detekováno. Pouze 1% z naměřených hodnot korespondujících s pozadím nebude správně detekováno, jelikož se vyskytuje mimo uvedený interval. Díky použití pružného prahu, $k\sigma$, lze znatelně minimalizovat počet falešných detekcí (pozadí chybně označeno jako popředí) vyplývajících z nevhodně zvoleného globálního prahu či ze zvýšené fluktuace v určitých místech scény. Tato technika je také schopna částečně reagovat na pravidelně se opakující děje (o zvýšené frekvenci opakování), jako je například šelest listů stromu, blikačka. Na druhou stranu u vysoce frekventovaných míst, u kterých dochází k velkým skokovým rozdílům intenzity (např. místa s velkým počtem pohybujících se objektů) může dojít k navýšení hodnoty rozptylu nad únosnou úroveň, což má za následek žádnou či jen částečnou detekci pohybujících se objektů. Proto je vhodné si zvolit strop, max. hodnotu, které rozptyl může nabývat. Zde bych chtěl podotknout, v případě, kdy se ve scéně objeví místo o zvýšené kolísavosti intenzity, bude citlivost detektoru snížena zvětšením rozptylu, čímž zamezíme falešným detekcím. Zvýšením rozptylu (rozumějme též prahu) je však částečně snížena i citlivost k detekci pohybujících se objektům mající podobnou barvu jako pozadí. Proto musíme řešit otázku, zda je pro nás výhodnější eliminace falešných detekcí za cenu nižší citlivosti nebo ponechání falešných detekcí za cenu lepší detekce pohybujících se objektů, popřípadě nalezení rozumného kompromisu mezi nimi.

Směsice Gaussových křivek (Mixture of Gaussians, MoG)

Metoda Směsice Gaussových křivek (též gaussovek nebo také gaussianů) byla poprvé představena pány Ch. Staufferem a W. E. L. Grimsonem v práci [11], jejichž cílem bylo vytvořit robustní, adaptivní metodu, která je schopna dostatečně se přizpůsobit změnám v osvětlení, pohybujícím se objektům, opakujícím se pohybům nestacionárních objektů pozorovaných ve scéně, jako jsou větve stromů, jejichž pohyb je závislý na větru ve scéně, nebo světla semaforu. Tyto změny pozadí způsobují, že hodnota intenzity pixelu se v průběhu času výrazně liší. Kdybychom chtěli pokrýt pouze pozvolné světelné změny ve scéně, vystačili bychom s jednou gaussovou. Nasazení jedné gaussovky na komplexnější scénu je však nedostačující. Proto ve své práci navrhuji použití směsice gaussovek pro modelování změn.

Pro model pozadí je použit multimodální přístup, kde model pozadí je složen z více modulů pozadí. Model je zde navržen jako směs K gaussových distribucí, která adaptivně modeluje každý pixel X . Potom pravděpodobnost pozorované hodnoty aktuálního pixelu může být formulovaná vztahem:

$$P(X_t) = \sum_{i=0}^K \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (3.8)$$

kde X_t je proměnná, která reprezentuje aktuální pixel ve snímku I_t , K je počet distribucí, obvykle je rovno 3, 4 nebo 5, t představuje čas (např. index snímku), $\omega_{i,t} \in [0,0; 1,0]$ je váha

i -té gaussovky ve směsi gaussovek v čase t , $\mu_{i,t}$ a $\Sigma_{i,t}$ je střední hodnota a kovariační matice i -té gaussovky ve směsi v čase t . η je gaussova hustota rozdělení pravděpodobnosti:

$$\eta(X_t, \mu_t, \Sigma_t) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_t|}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \quad (3.9)$$

kde $|\Sigma_{k,t}|$ je determinant kovariační matice, $(X_{t+1} - \mu_{i,t})^T$ je zápis transponovaného vektoru. $\eta(X_t, \mu_{k,t}, \Sigma_{k,t})$ vyjadřuje s jakou pravděpodobností bod X_t náleží dané gaussovcce (resp. k pozadí). Z důvodů nižších výpočetních nároků je pány Ch. Staufferem a W. E. L. Grimsonem kovariační matice navržena jako diagonální:

$$\Sigma_{k,t} = \sigma_k^2 I \quad (3.10)$$

kde I je jednotková matice o příslušné dimenzi⁴. Tato definice předpokládá, že složky RGB vektoru jsou navzájem nezávislé.

Základem tedy je, že každý pixel je modelovaný směrnicí K gaussovek. Poprvé při inicializaci modelu směsice gaussovek jsou parametry jednotlivých gaussovek manuálně nastaveny. Poté jsou již automaticky aktualizovány. Nejprve jsou gaussovky sestupně seřazeny podle hodnoty ω/σ . Toto seřazení vychází z faktu, že pixel náležící pozadí má vysokou váhu (nejvýše však 1) a malou hodnotu rozptylu kvůli tomu, že se pozadí ve scéně nachází delší dobu (tím vyšší váha) a vyznačuje se konstantnějšími hodnotami v průběhu času (tím menší rozptyl).

Potom prvních B distribucích je vybráno jako model pozadí, kde:

$$B = \underset{b}{\operatorname{argmin}} \left(\sum_{k=1}^b \omega_k > T \right) \quad (3.11)$$

kde T je práh, který určuje míru minimální části dat, které jsou zahrnuty do pozadí. Když je hodnota T velmi malá, je model pozadí vždy unimodální. Pro model pozadí je vždy vybrána ta nejlepší (nejpravděpodobnější) distribuce. Když je hodnota T vysoká, pro model pozadí je vybráno více distribucí, které zachycují i opakující se pohyby ve scéně, jako jsou pohyby větví, světla semaforu. Vyšší hodnotu je vhodné nastavit na video sekvenci zachycující venkovní scénu. V literatuře se obvykle doporučuje hodnotu T nastavit na 0,75.

Potom když nový snímek přichází v čase $t+1$, je hodnota každého pixelu tohoto snímku porovnávána s existujícími gaussovými distribucemi, dokud není nalezena shoda. Pixel se shoduje s distribucí, pokud platí:

$$\sqrt{(X_{t+1} - \mu_{i,t})^T (X_{t+1} - \mu_{i,t})} < k \sigma_{i,t} \quad (3.12)$$

kde $(X_{t+1} - \mu_{i,t})^T$ je zápis transponovaného vektoru, $\sigma_{i,t}$ a $\mu_{i,t}$ je standardní (směrodatná) odchylka a střední hodnota i -té gaussovky v čase t , k je konstanta, obvykle $k \in (2,0; 3,0)$. Na základě výsledku porovnání dochází k aktualizaci příslušných hodnot dle scénáře:

⁴Názorněji pro případy, kdy hodnota pixelu je definovaná jako:

$$\text{skalár: } \eta(X_t \text{ val}, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)\sigma^2}} e^{-0,5 \frac{(\text{val} - \mu)^2}{\sigma^2}}$$

$$\text{RGB vektor: } \eta(X_t(r, g, b), \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^3 \sigma^2 \sigma^2 \sigma^2}} e^{-0,5 \frac{(R - \mu_R)^2}{\sigma^2} - 0,5 \frac{(G - \mu_G)^2}{\sigma^2} - 0,5 \frac{(B - \mu_B)^2}{\sigma^2}}$$

Když X_t se shoduje s i -tou distribucí, parametry této distribuce jsou aktualizovány následovně:

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha \quad (3.13)$$

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho X_{t+1} \quad (3.14)$$

$$\sigma_{i,t+1}^2 = (1 - \rho)\sigma_{i,t}^2 + \rho(X_{t+1} - \mu_{i,t})^T(X_{t+1} - \mu_{i,t}) \quad (3.15)$$

kde

$$\rho = \alpha\eta(X_{t+1}, \mu_i, \Sigma_i) \quad (3.16)$$

α je předdefinovaný učicí parametr, obvykle $\alpha \in (0,001; 0,1)$. Tento postup aktualizace je aplikován pouze na první nalezenou shodu, pokud jsou po první shodě nalezeny ještě další shody, příslušné distribuce jsou aktualizovány již dle (3.17), (3.18) a (3.19).

Když X_t se neshoduje s i -tou distribucí, váha distribuce je snížena a parametry této distribuce jsou aktualizovány následovně:

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} \quad (3.17)$$

$$\mu_{i,t+1} = \mu_{i,t} \quad (3.18)$$

$$\sigma_{i,t+1}^2 = \sigma_{i,t}^2 \quad (3.19)$$

Pokud se X_t neshoduje s žádnou z K distribucí, potom je nejméně pravděpodobná distribuce (např. s nejmenší vahou ω či poměrem ω/σ) nahrazena novou distribucí:

$$\omega_{new,t+1} = \text{malá hodnota váhy} \quad (3.20)$$

$$\mu_{new,t+1} = X_{t+1} \quad (3.21)$$

$$\sigma_{new,t+1}^2 = \text{velká hodnota rozptylu} \quad (3.22)$$

Po výsledných operacích jsou váhy všech K distribucí normalizovány:

$$\omega_i = \frac{\omega_i}{\sum_{i=0}^K \omega_i} \quad (3.23)$$

jelikož musí platit, že součet vah všech K distribucí je roven jedné. Podle Ch. Staufferovi a W. E. L. Grimsonovi práce [11] je nutné pro běh systému nastavit pouze dva parametry α a T .

U tohoto systému můžeme pozorovat dva hlavní nedostatky. Za prvé, Ch. Stauffer a W. E. L. Grimson inicializují váhu, střední hodnotu a rozptyl každé gaussovky pomocí K-means algoritmu, což v inicializační fázi vyžaduje trénovací sekvenci bez popředí. P. KaewTraKulPong a spol. ve své práci [8] navrhli metodu pro inicializační fázi, která zlepšuje rychlost učení a dovoluje rychlejší přechod k stabilnímu modelu pozadí. Během inicializační fáze (prvních L snímků), učicí parametr α je proměnný a řídí se vztahem: $\alpha = 1/t$, kde t je aktuální počet snímků. Po uplynutí L snímků, $\alpha < \alpha_{min}$ též $t > L$, je učicí parametr nastaven jako konstanta: $\alpha = \alpha_{min} = 1/L$. Například pro $\alpha = 0.005$ je L rovno 200 snímkům. Za druhé, systém se vyznačuje velmi pomalou adaptivitou. Problémem je, že učicí parametr ρ je velmi malý kvůli složce hustoty pravděpodobnosti. Což vede k pomalé aproximaci střední hodnoty a rozptylu. P. KaewTraKulPong a spol. navrhuje složku hustoty pravděpodobnosti zcela vypustit z definice ρ . Z. Zivkovic [13] tuto myšlenku rozšířil a navrhuje ρ definovat jako $\rho = \alpha/\omega$.

Podrobnější informace o metodě Směsice Gaussových křivek je možné nalézt v rozsáhlé práci [4] od T. Bouwmanse a spol., která mapuje techniky MoG založené na modelu od Ch. Stauffera a W. E. L. Grimsone [11]. Kompletní analýzu ohledně nastavení parametrů u modelu MoG lze nalézt v práci [12].

3.3.3 Citlivost výběru (Selectivity)

Při modelování pozadí mnoho autorů navrhuje přístup aktualizace pozadí podle klasifikace jednotlivých bodů. Vychází z logické úvahy, že body klasifikované jako popředí by se na podobě pozadí měly podílet odlišnou vahou než body klasifikované jako pozadí.

Nulová selektivita - tato varianta je zde uvedena jen pro úplnost. Při této variantě je model pozadí aktualizován všemi body bez ohledu na klasifikaci bodu. Pro Klouzavý průměr (3.4) by vzorec vypadal následovně:

$$B_{t+1}(x, y) = \alpha I_t(x, y) + (1 - \alpha)B_t(x, y) \quad \text{pro všechny } (x, y) \quad (3.24)$$

Úplná selektivita - při této variantě je model pozadí aktualizován jen body, které nejsou klasifikovány jako popředí, v opačném případě model pozadí zůstává nezměněn. Mezi charakteristické rysy tohoto přístupu patří: model pozadí není po celou dobu běhu ovlivněn novými objekty ve scéně, nemožnost reakce na dlouho trvající změny ve scéně (světelné změny, znehybnění pohybujících se objektů), chybné vyhodnocování, pokud model pozadí byl inicializován snímkem obsahující pohybující se objekty. Poslední uvedená vlastnost má ve většině případů za důsledek vznik duchů, což má negativní vliv na celkovou kvalitu detekce. Pro Klouzavý průměr (3.4) by vzorec vypadal následovně:

$$B_{t+1}(x, y) = \begin{cases} \alpha I_t(x, y) + (1 - \alpha)B_t(x, y) & \text{kdy } (x, y) \notin F_t(x, y) \\ B_t(x, y) & \text{kdy } (x, y) \in F_t(x, y) \end{cases} \quad (3.25)$$

Částečná selektivita - při této variantě je model pozadí aktualizován oběma typy bodů, při čemž typ bodu určuje váhu, jakou se bude podílet na modelu pozadí. Při aplikování podílové selektivity na Klouzavý průměr (3.4) by vzorec vypadal následovně:

$$B_{t+1}(x, y) = \begin{cases} \alpha I_t(x, y) + (1 - \alpha)B_t(x, y) & \text{kdy } (x, y) \notin F_t(x, y) \\ \beta I_t(x, y) + (1 - \beta)B_t(x, y) & \text{kdy } (x, y) \in F_t(x, y) \end{cases} \quad (3.26)$$

kde $\alpha, \beta \in [0, 0; 1, 0]$. Tento přístup umožňuje dostatečně reagovat na dlouho trvající změny ve scéně, jimiž mohou být světelné změny o velké intenzitě či znehybnění pohybujících se objektů. Z pohledu eliminace duchů si metoda taktéž nestojí špatně. Čím je hodnota α menší, tím se pozadí rychleji aktualizuje. A proto jsou duchové rychleji eliminovány. Na druhou stranu, pokud je hodnota β příliš velká, tak se může objevit efekt v práci [5] označovaný jako *wake-effect*. Efekt se objevuje v popředí v opačném směru pohybujícího se objektu. Tento efekt je znatelnější u větších pohybujících se objektů, pomalejšího pohybu a větší hodnoty β . Pokud má být systém vysoce účinný v eliminaci duchů, tak tato strategie (3.26) v [5] není doporučována. Místo toho autoři navrhují přístup založený na konceptu *Pernamence memory*.

Pernamence memory - přístup založený na konceptu *Pernamence memory*, PM , vyžaduje existenci nové matice dat $PM_t(x, y)$ o stejné velikosti jako je matice popředí, každá hodnota $PM_t(x, y)$ je asociována s příslušným bodem matice popředí. Potom v každé instanci $PM_t(x, y)$ v čase t je definovaná matice dat, kde hodnota $PM_t(x, y)$ je získána z hodnoty předchozí instance $t - 1$ a z binárního vstupu $F_t(x, y)$ podle předpisu:

$$PM_t(x, y) = \begin{cases} PM_{t-1}(x, y) + C_0 \text{ (až do } C_{max}) & \text{kdy } F_t(x, y) = 1 \\ 0 & \text{jinak} \end{cases} \quad (3.27)$$

kde inicializovaná matice $PM_0(x, y) = 0$, C_0 je konstanta, zde $C_0 = 1$, kterou je matice PM inkrementována, C_{max} je maximální hodnota, která může být uložena v PM . Když hodnota bodu matice PM v čase t , $PM_t(x, y)$, dosahuje maximální hodnoty C_{max} , potom bod pozadí umístěný ve stejné pozici, $B_t(x, y)$, je aktualizován hodnotou z aktuálního snímku, $I_t(x, y)$. Celkový systém aktualizace modelu pozadí je popsán rovnicí (3.28).

$$B_{t+1}(x, y) = \begin{cases} \alpha I_t(x, y) + (1 - \alpha) B_t(x, y) & \text{kdy } (x, y) \notin F_t(x, y) \\ I_t(x, y) & \text{kdy } (x, y) \in F_t(x, y) \text{ a} \\ & PM_t(x, y) = C_{max} \\ B_t(x, y) & \text{kdy } (x, y) \in F_t(x, y) \text{ a} \\ & PM_t(x, y) < C_{max} \end{cases} \quad (3.28)$$

Zvolená hodnota C_{max} udává počet snímků, které musí uplynout k eliminaci duchů, kteří se objevili v popředí. Čím menší C_{max} , tím je eliminace duchů rychlejší. Nic méně hodnota konstanty bude záviset převážně na typické rychlosti pohybujících se objektů ve scéně. Například pokud se objekt pohybuje pomalu, je nutné hodnotu C_{max} nastavit větší. Jinak při uplynutí času C_{max} by hrozilo, že model pozadí by mohl být aktualizován hodnotou příslušející ještě pohybujícímu se objektu.

3.4 Ověření dat

V této fázi dochází k ověření platnosti výstupních dat, které jsou reprezentovány binární maskou popředí. Běžným přístupem je odstranění izolovaných bodů aplikováním jedním z trojice morfologických filtrů eroze, dilatace, medián či jejich vhodnou kombinací. V praxi je však tento proces nedostačující, proto často bývá ještě doplněn metodou vytváření blobů (sady propojených bodů), které slouží k identifikaci s reálným objektem a k eliminaci těch blobů, které jsou příliš malé, aby korespondovaly s reálným objektem.



(a) Aktuální snímek, $t = 874$



(b) Model pozadí



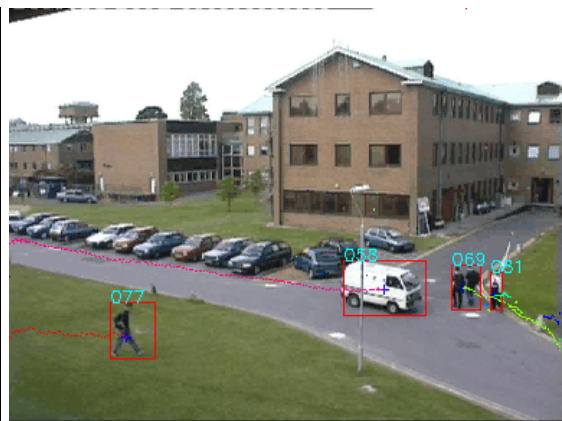
(c) Maska popředí



(d) Vyčištěná maska popředí



(e) Extrakce popředí



(f) Výsledek detekce doplněný o sledování objektů

Obrázek 3.1: Úkázka jednotlivých fází detekce

Kapitola 4

Sledování objektu (Object tracking)

Sledování objektu přes několik snímků video sekvence je jedním z dalších důležitých úkolů v oblasti počítačového vidění, které často navazuje a pracuje s výstupními daty získanými při procesu detekce pohyblivého tělesa. Tato disciplína se zabývá získáním ucelené informace o pohybu objektu ve video sekvenci. Cílem je nalezení podobnosti ve vlastnostech objektů v jednotlivých po sobě jdoucích snímcích. To lze také popsat jako snahu o přiřazení sledované trajektorie objektu k objektu nalezeného v následujícím snímku video sekvence. Trajektorie vyjadřuje polohu objektu během jeho pohybu v každém momentu čase t .

Abychom mohli sledovací systém použít na komplexní scénu, zaměříme na oblast sledování vícero objektů ve scéně (multi-object tracking), který oproti sledování jednoho objektu přináší řadu nových problémů. Systém musí být schopen řídit zánik, rozdělení, sloučení a další složité případy související s vícenásobným výskytem objektů. Informace k problematice sledování objektů byly čerpány z prací [7, 9], .

4.1 Formulace problému

Základním problémem je přiřazení sady objektů $R^{t-1} = \{r_1^{t-1}, r_2^{t-1}, \dots, r_M^{t-1}\}$ předchozího snímku k blobům $B^t = \{b_1^t, b_2^t, \dots, b_N^t\}$ vyskytujících se v aktuálním snímku. Blob (někdy nazýván cíl) je základní entitou, která představuje detekovaný objekt. Nutné je podotknout, že jeden blob může být tvořen i více objekty, takovým blobem je např. slupina lidí. Každý blob je možné popsat sadou vlastností, jako je pozice, velikost, barva, textura, rychlost a tvar.

4.2 Porovnání blobů

Nejběžnějším přístupem je využití tabulky porovnání, přesněji matice porovnání (Match matrix). Pokud budeme předpokládat, že v aktuálním snímku se vyskytuje N blobů a v předchozím snímku M blobů, tak bude požadována matice o velikosti $M \times N$. Kde řádky budou představovat objekty a sloupce představují bloby detekované v aktuálním snímku. Matice porovnání nám vyjadřuje míru shody mezi jednotlivými bloby a objekty. Míra shody je nejčastěji vyjádřena jako euklidovská vzdálenost mezi objektem a blobem, resp. jejich sadou vlastností. V této práci je blob definovaný jako trojice: $B = \{P, C, S\}$, kde P je pozice blobu, C je barva blobu a S je velikost blobu. Potom matice porovnání $MM_{i,j}$ je

vypočtena dle předpisu:

$$MM_{i,j} = \sqrt{(X_r - X_b)^2 + (Y_r - Y_b)^2} + \sqrt{(\overline{R}_r - \overline{R}_b)^2 + (\overline{G}_r - \overline{G}_b)^2 + (\overline{B}_r - \overline{B}_b)^2} + \sqrt{(S_r - S_b)^2} \quad (4.1)$$

kde pozice blobu je popsána souřadnicemi svého centroidu, X a Y , barva je vyjádřena jako průměrné hodnoty jednotlivých složek RGB vektoru, \overline{R} , \overline{G} , \overline{B} , spočtených přes oblast blobu a velikost zde představuje celkový počet pixelů tvořící blob, S .

Tato technika však trpí vážnými nedostatky. Jelikož jednotlivé vlastnosti mohou nabývat různě velikých hodnot, jedna vlastnost se může na výsledku podílet větší vahou než jiná. Proto v práci [2] navrhuji jednotlivé vlastnosti normalizovat do intervalu $[0,0; 1,0]$. Například velikost blobu podělit celkovou velikostí snímku. Podobně hodnoty jednotlivých složek RGB vektoru podělit hodnotou 255 a souřadnice centroidu blobu podělit rozměry snímku. Matice porovnání je potom vypočtena jako normalizovaná euklidovská vzdálenost dle předpisu:

$$MM_{i,j} = 1 - \sqrt{\frac{1}{6}(\Delta X/Xdim)^2 + (\Delta Y/Ydim)^2 + (\Delta \overline{R}/255)^2 + (\Delta \overline{G}/255)^2 + (\Delta \overline{B}/255)^2 + (\Delta S/velSnimku)^2} \quad (4.2)$$

kde ΔX , ΔY vyjadřuje vzdálenost mezi blobem a objektem na příslušné x -ové a y -ové souřadnici, $\Delta \overline{R}$, $\Delta \overline{G}$ a $\Delta \overline{B}$ vyjadřuje rozdíl průměrných hodnot jednotlivých barevných RGB složek, ΔS je rozdíl velikosti ¹.

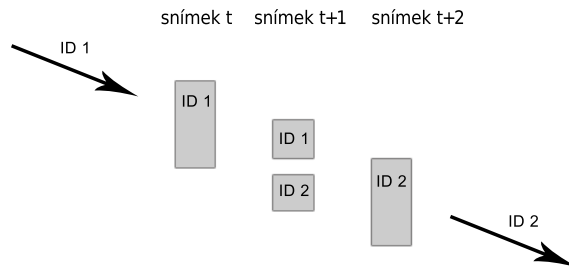
Při vyjádření vzdálenosti dvou porovnávaných blobů jsem upřednostnil metriku založenou na obdélníku vymezuujícího hranice blobu před metrikou počítající s centroidy z důvodu získání relevantnějších informací. Ohraničující obdélník (Bounding box) je definován jako nejmenší obdélník, který obsahuje celou oblast blobu. V případě, že se obdélníky v sousedních snímcích částečně překrývají, pak je jejich vzdálenost rovna nule. Překryv zde představuje jednu z klíčových indicií pro nalezení identity objektu mezi sousedními snímky.

U přiřazení objekt-oblob, u nichž je rozdíl vzdálenosti abnormálně veliký, $\Delta X > X_{max} \vee \Delta Y > Y_{max}$, je míra shody automaticky nastavena na nulu. Jelikož tato přiřazení lze z fyzikálního hlediska označit jako nereálná.

Jedním z častých jevů, které komplikují sledování objektů, je rozpadnutí objektu na více blobů. Tento jev je způsoben špatnou extrakcí objektu při procesu detekce pohyblivého tělesa, kde část objektu je nesprávně detekována jako pozadí, nebo zapříčiněn stacionárními tělesy ve scéně, která překrývají část objektu např. pouliční lampa, kmen stromu. Problém rozpadnutí objektu lze částečně řešit spojením blízkých blobů do jednoho, podmínka pro spojení je formulovaná: $\Delta Y \leq \tau_Y \wedge \Delta X \leq \tau_X$, kde ΔX , ΔY je vzdálenost mezi dvěma bloby na příslušné x -ové a y -ové souřadnici, τ_Y a τ_X jsou prahy maximální vzdálenosti. V mém případě se osvědčila hodnota $\tau_Y = \tau_X = 4$.

Řešení pomocí spojení blízkých blobů bohužel není dostatečné, jelikož práh maximální vzdálenosti nemusí zahrnout všechny bloby. Potom z rozpadených blobů pouze jeden převzme identitu objektu z předchozího snímku, zbylé bloby jsou identifikovány jako nové objekty. U těchto objektů však kvůli jejich podobným vlastnostem často dochází k vzájemnému přehazování identity mezi jednotlivými snímky, jak je ilustrováno na obrázku 4.1. Pro tuto situaci je charakteristické, že části rozpadlého objektu mají krátkou životnost, pouze několik málo snímků, poté je objekt opět tvořen pouze jedním blobem. Tento problém je

¹operace odčítání od jedničky je ve vzorci definovaná spíše z didaktických důvodů, aby hodnoty v matici bylo možno interpretovat jako míru shody mezi porovnávanými objekty a ne jako míru odlišnosti.



Obrázek 4.1: Příklad přerušení identity objektu na základě rozpadnutí objektu

řešen zvýhodněním stabilních objektů, respektive znevýhodněním nových objektů, u nichž je míra shody s porovnávanými bloby již předem automaticky snížena o určitou hodnotu, v mém případě se jedná o hodnotu 0,05.

4.3 Přiřazení objektu k blobu

Přiřazení objektu k blobu na základě vypočtené matice porovnání není triviální záležitostí. Tato oblast je v literatuře známa jako *Přiřazovací problém* (*Assign problem*), kde se snažíme o přiřazení blobů k objektům, tak aby každý blob byl přiřazen jednomu objektu a naopak.

Nejjednodušší strategií je, že se prochází matice po řádcích, v každém řádku je nalezena „nejlepší shoda“, na jejímž základě dochází k přiřazení objekt-blob. Tato technika však trpí mnohými nedostatky. Výsledek přiřování není stabilní, závisí na pořadí procházených řádků, což často degraduje optimální výsledek. Příklad přiřazení je ilustrován na obrázku 4.2b.

Častější strategií nalezení optimálního přiřazení je založena na maximalizaci součtu matice, nalezení největšího součtu přiřazení objekt-blob. Tato úloha je definovaná na prostoru $n!$, kde počet možných řešení (kombinací) je dán permutací. Například pokud budeme mít matici o velikosti 3×3 , potom počet kombinací přiřazení objekt-blob, které je nutné projít, je roven faktoriálu $3! = 3 \cdot 2 \cdot 1 = 6$, jmenovitě $r1-r2-r3$, $r1-r3-r2$, $r2-r1-r3$, $r2-r3-r1$, $r3-r1-r2$, $r3-r2-r1$. V případě tabulky na obrázku 4.2c je nejlepší přiřování nalezeno u kombinace $r1-r3-r2$, jejíž součet je roven $0,9 + 0,5 + 0,7 = 2,1$. Nevýhodou této techniky je, že s rostoucí velikostí matice se rapidně zvyšuje výpočetní náročnost. Například pro matici 5×5 to znamená projít 120 kombinací, pro matici 12×12 již 479 001 600 kombinací. Pokud

	B1	B2	B3
R1	0,9	0,2	0,3
R2	0,1	0,8	0,7
R3	0,1	0,5	0,2

(a)

	B1	B2	B3
R1	0,9	0,2	0,3
R2	0,1	0,8	0,7
R3	0,1	0,5	0,2

(b) *Suma* = 1,9

	B1	B2	B3
R1	0,9	0,2	0,3
R2	0,1	0,8	0,7
R3	0,1	0,5	0,2

(c) *Suma* = 2,1

Obrázek 4.2: a) je příklad matice porovnání. b) je příklad přiřazení objekt-blob metodou „nejlepší shoda“. c) je příklad přiřazení objekt-blob metodou maximalizace součtu.

se bude ve video sekvenci v jeden moment nacházet větší počet zájmových objektů, může to pro sledovací systém znamenat výrazné zpomalení. Proto jsem zavedl omezení velikosti matice na maximální velikost 7×7 , v níž jsou upřednostněny objekty s větší velikostí.

Zlepšení výkonnosti sledovacího systému lze docílit nasazením specializovaných algoritmů na prohledávání matice, které využívají heuristické přístupy a aproximační metody řešení. Mezi ně například patří iterativní algoritmus *Softassign* nebo známější *Maďarský algoritmus* (též znám jako *Kuhn-Munkresův algoritmus*), který byl poprvé publikován v roce 1955 Haroldem Kuhnem. Tento algoritmus pracuje s časovou složitostí $O(N^3)$ oproti původní faktoriálové časové složitosti $O(n!)$.

Kapitola 5

Návrh a implementace testovací aplikace

Tato kapitola se zabývá návrhem a implementací aplikace, jejíž snahou je demonstrování činnosti některých detekčních algoritmů spolu se sledovacím algoritmem popsaných v předchozích kapitolách. Při návrhu aplikace byl hlavně kladen důraz na funkčnost a efektivitu detekčních algoritmů před uživatelským rozhraním samotné aplikace.

Mimo demonstrování činnosti naimplementovaných detekčních algoritmů, je také cílem poskytnout informace, co se děje v průběhu jednotlivých fází detekčního procesu. Tyto informace jsou poskytovány v podobě vizuálních dat, která nám dovolují získat lepší představu o principech detekce pohyblivého tělesa.

Navrhnutá aplikace byla implementována pomocí programovacích jazyků C a C++ s použitím volně dostupné knihovny OpenCV, což z ní činí multiplatformní aplikaci. Knihovna OpenCV zde zajišťuje jednoduchý přístup k obrazovým datům video sekvence. Jako vstupní datový proud je použit videosoubor uložený ve formátu .avi. Vzhledem k požadavku na schopnost aplikace pracovat v reálném čase je také možné použít přímý datový proud z připojené kamery.

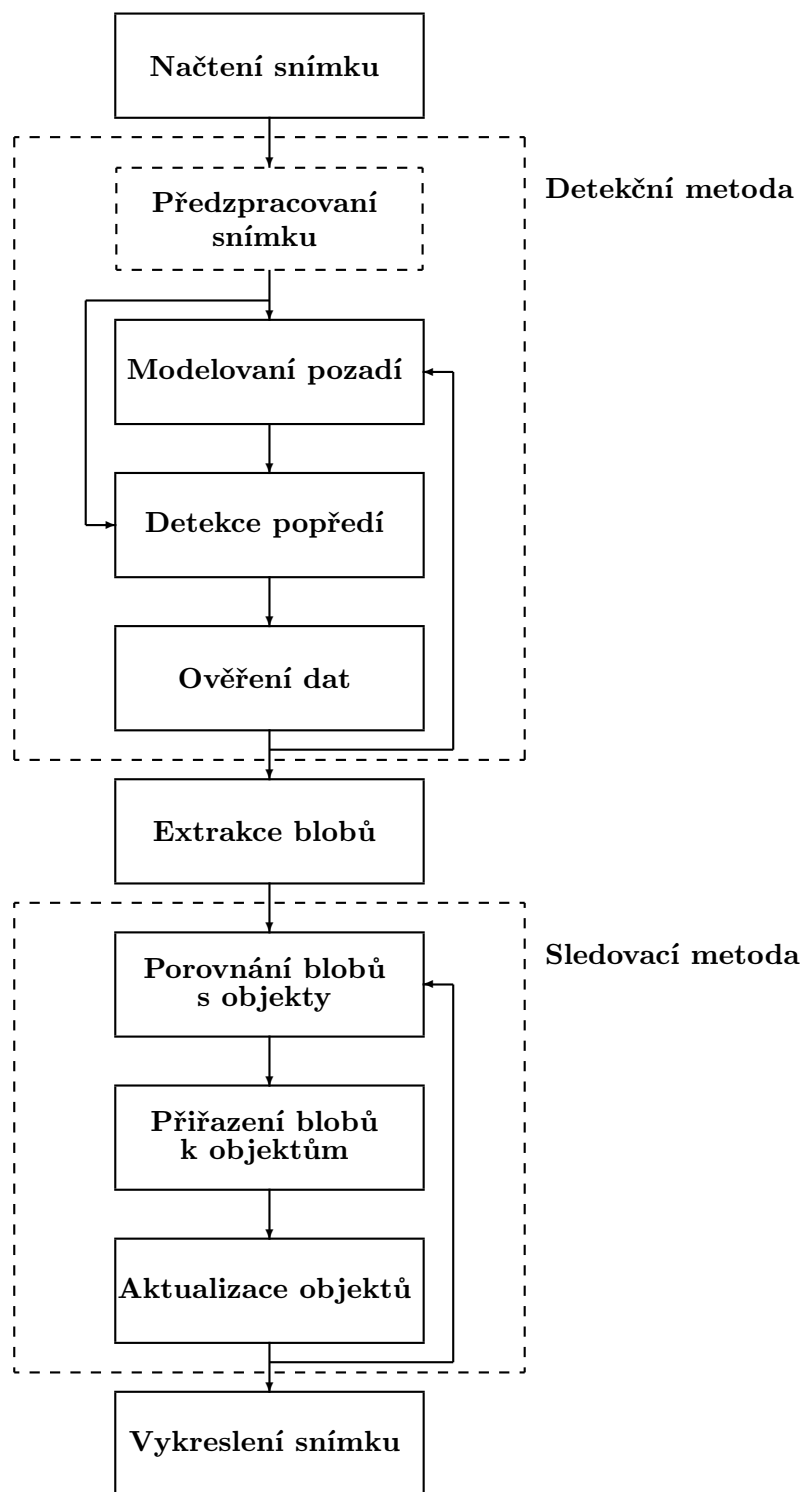
5.1 Návrh aplikace

Aplikace byla navržena tak, aby byla schopna uspokojit naše požadavky ohledně sběru, zpracování a vizualizace výstupních dat. Při návrhu aplikace bylo nutné řešit problémy jako je zajištění přístupu k obrazovým datům, efektivní manipulace s daty a možnost jejich vizualizace, způsob komunikace mezi detekčním a sledovacím modulem a v neposlední řadě také volba vhodné formy prezentace výstupních dat.

Na základě vztahů mezi jednotlivými komponenty byla navržena struktura aplikace, která je popsána diagramem datových toků na obrázku [5.1](#).

Vstupem aplikace je načtená video sekvence. Zdrojem této video sekvence může být jednak připojená kamera nebo videosoubor uložený ve formátu .avi. Díky možnosti připojení různých zdrojů lze lépe analyzovat vlastnosti detekčního algoritmu. Pokud se zabýváme analýzou algoritmu pracujícího v reálném čase, pak lze jako zdroj využít připojenou kameru. Pro hlubší analýzu algoritmů je přínosnější pracovat s uloženým videosouborem, nad kterým lze provádět opětovná vyhodnocování na základě různého nastavení detekčního algoritmu.

Činnost detekčního algoritmu je zde rozdělena do třech kroků: modelování pozadí, detekce popředí a ověření finálních dat. Jelikož všechny zde implementované detekční algo-



Obrázek 5.1: Diagram datových toků.

ritmy jsou založeny na metodě odečítání pozadí, tak popisovaný scénář je pro všechny tyto algoritmy shodný.

Výběr vhodné strategie modelování pozadí je jeden z klíčových faktorů úspěšnosti celého detekčního procesu. U modelování pozadí je snaha omezení náchylnosti ke změnám v pozadí, ale zároveň udržení citlivosti k identifikaci pohybujících se objektů. Mezi techniky modelování pozadí jsem zařadil Průměr posledních N snímků, Klouzavý průměr, Klouzavý Gaussův průměr, Směsice Gaussových křivek, které jsou diskutovány v sekci 3.3. Jedná se o různě úspěšné techniky, jak odhadovat pozadí, jehož hodnota se v čase pozvolna mění.

Na základě dostupného modelu pozadí dochází k detekci popředí, již snahou je nalezení pohybujících se objektů v aktuálním snímku. Princip detekce spočívá v odečítání aktuálního snímku od modelu pozadí pixel po pixelu. Výsledkem je binární maska o stejném rozměru jako aktuální snímek. V úspěšnosti detekce popředí hraje velkou roli zvolená hodnota prahu. U ní je často potřeba měnit hodnotu v závislosti na typu zvolené video sekvence. Proto je nutné, aby při chodu aplikace byla dostupná možnost si zvolit svou hodnotu prahu. Naproti tomu metoda Klouzavý gaussův průměr je již navržena tak, že si dokáže obsluhovat hodnotu prahu zcela automaticky.

Výsledkem detekce popředí je binární maska, která nám popisuje možné kandidáty na popředí. Tato maska je však silně degradovaná přítomným šumem, což má za následek velké množství falešných detekcí. Běžným přístupem je odstranění izolovaných bodů v masce aplikováním různých filtrů. Pro tento účel je nejvýhodnější aplikovat jeden z trojice morfologických filtrů - dilatace, eroze a medián či jejich vhodná kombinace.

Po ověření platnosti dat v masce popředí je pro další chod aplikace nezbytně nutné vydolovaná data převést na jinou reprezentaci. Takovou vhodnou reprezentací jsou pro nás blosy, jedná se o spojenou sadu pixelů tvořící souvislou oblast. V našem případě se na blosy již můžeme dívat jako na jednotlivé pohyblivé objekty ve scéně.

Ze sady bodů, z kterých je blob tvořen, lze vyextrahovat další důležité informace o vlastnostech blobu. Mezi tyto vlastnosti můžeme zařadit polohu centroidu blobu (geometrický střed blobu), hranice blobu vyjádřena pomocí obdélníku, počet bodů tvořící blob, průměrné hodnoty jednotlivých složek RGB vektoru spočtených přes oblast blobu.

V tuto chvíli jsme schopni názorně prezentovat výsledky detekčního algoritmu na jednotlivých snímcích video sekvence. Výsledky detekčního algoritmu lze demonstrovat na prvotní binární masce popředí, popřípadě binární masce popředí zbavené šumu. Dalším nástrojem pro prezentování výsledků je vizuální označení blobů, které lze provést jednak vykreslením hranic blobu pomocí obdélníku nebo obarvením samotné oblasti blobu.

Dalším krokem je navržení sledovacího algoritmu, u něhož jsou požadavky na co nejmenší časovou náročnost výpočtu a na schopnost sledování více objektů na jednou. U sledování objektů ve video sekvenci se snažíme získat ucelenou informaci o pohybu objektu. Principem sledovacího algoritmu je párování blobů mezi předchozím a aktuálním snímkem na základě jejich podobnosti. Sledovací proces se skládá z následujících kroků: výpočet podobnosti mezi blosy z předchozího a aktuálního snímku, přiřazení blobů a aktualizace sledovaných objektů.

Podobnost mezi dvěma blosy je zde vypočítána jako euklidovská vzdálenost mezi vlastnostmi blobů. S důrazem na co nejmenší výpočetní náročnost byla sada vlastností zvolena jako trojice: pozice, barva a velikost blobu. Před výpočtem euklidovské vzdálenosti jsou však nejprve hodnoty všech vlastností normalizovány, aby se na výsledné hodnotě podílela každá vlastnost stejnou váhou. Finální přiřazení je provedeno na základě maximalizace součtu jednotlivých přiřazení. Výběr přiřazení je volen s důrazem na maximalizaci zisku podobnosti z pohledu celkové scény, nikoliv pouze z pohledu jednotlivců.

Data produkovaná ve sledovací fázi jsou prezentována v podobě vykreslení trajektorie u každého sledovaného objektu. Trajektorie zde představuje sadu bodů, kterými objekt prošel v průběhu předchozích snímků. Dalším důležitým nástrojem pro prezentaci výsledků v této fázi je vykreslení identifikačního čísla u každého sledovaného objektu.

5.2 Implementace

Aplikace je implementována pomocí programovacích jazyků C a C++ ve svobodném vývojovém prostředí Dev-C++. Pro práci s obrazovými daty zde byla použita volně šiřitelná knihovna OpenCV, která poskytuje nástroje pro přístup k video sekvenci, funkce pro manipulaci s obrazovými daty a jednoduchý aparát pro tvorbu GUI.

Z pohledu navržené aplikace je pro nás vhodné části aplikace implementovat jako samostatné komponenty, což nám přinese větší nezávislost a jednodušší údržbu těchto celků. K tomu účelu je využito vlastností objektového programování, kterými disponuje jazyk C++.

5.2.1 Třídy `BS_detector` a `Tracker`

Třída `BS_detector`, kde `detector` představuje název konkrétní detekční metody, implementuje vlastnosti detekce pohybu založené na technice odečítání pozadí, které jsou ve třídě popsány sadou atributů a operací. Mezi důležité atributy patří model pozadí a sada proměnných, které ovlivňují kvalitu detekční metody, mezi ně patří učící konstanty, práh detekce a inicializační konstanty. Proces detekce je zde popsán pomocí operací, které lze nad třídou vykonávat. Pro správný chod detekční metody musí nejprve dojít k nastavení proměnných, to zde zajišťuje konstruktor třídy. Konstruktor zde může být volán bez parametru, na jehož základě dojde k nastavení proměnných defaultními hodnotami, nebo s jedním parametrem, který zajišťuje nastavení proměnných z konfiguračního souboru. Tato inicializace však není kompletní, jelikož je potřeba alokovat paměť pro model pozadí, k tomu slouží metoda `initBackgroundModel()`. Samotný proces detekce je zajišťován metodou `subtract()`, která provádí odečítání aktuálního snímku od modelu pozadí a následnou aktualizaci modelu pozadí. Výsledkem operace je surová binární maska popředí, kterou je možno v dalších krocích pročistit od přítomného šumu. V třídě `BS_detector` je implementovaná metoda `renderBackground()`, pomocí níž lze vykreslit aktuální stav modelu pozadí.

Vlastnosti metody pro sledování objektů jsou popsány třídou `Tracker`. Mezi sadu jejich atributů patří seznam aktuálních tracků, který je implementován pomocí indexového kontejneru `<vector>` poskytovaného knihovnou STL, počet tracků, který zajišťuje generování identifikačního čísla pro nový track, a maximální počet blobů, kterým lze omezit počet sledovaných blobů v každém snímku. Třída `Tracker` disponuje přetíženým konstruktorem podobně jako třída `BS_detector`. Z pohledu činnosti je pro nás nejdůležitější operace `updateTracks()`, která tvoří jádro sledovacího procesu. Operace provádí spárování tracků s aktuálně detekovanými bloby, na základě této operace dochází k aktualizaci seznamu tracků, jednotlivé tracky jsou buď smazány (v případě nenalezení vhodného kandidáta na spárování), aktualizovány (v případě nalezení vhodného kandidáta na spárování) nebo nově vytvořeny (v případě nalezení nového objektu ve scéně). Pro vykreslení výsledků sledovacího procesu je dostupná operace `renderTracks()`, která umožňuje kombinovat vykreslování ID čísla nad sledovanými bloby či vykreslování celkového pohybu ve scéně pomocí trajektorií.

V práci je také použita třída `BlobExtractor`, která zde tvoří objektovou nadstavbu knihovny `cvBlob`.

5.2.2 Knihovny OpenCV, cvBlob a TinyXML

Při implementaci aplikace bylo využito některých již existujících knihoven, při jejichž volbě hrála roli především možnost snadné modifikace a integrace do naší aplikace.

- knihovna OpenCv 1.0.0
- knihovna cvBlob 0.9.5
- knihovna TinyXML 2.6.1

Nejdůležitější knihovnou v rámci naší aplikace je knihovna OpenCV, jedná se o volně šiřitelnou knihovnu určenou pro manipulaci s obrazem původně vyvíjenou společností Intel. Knihovna poskytuje nástroje pro přístup k video sekvenci, při čemž jako zdroj může sloužit videosoubor uložený ve formátu `.avi` nebo připojená kamera, řadu funkcí pro efektivní manipulaci s obrazovými daty a jednoduchý aparát pro tvorbu GUI.

Knihovna `cvBlob` vznikla jako alternativa k robustnější knihovně `cvBlobsLib` vyvíjené v rámci OpenCV. Knihovna obsahuje základní funkce pro práci s bloby, jako je extrakce blobu a jeho vlastností, filtrace blobů dle velikosti, vizuální lokalizace blobu. V tomto projektu byla nasazena pro extrakci blobů z binární masky popředí. Knihovně se v našem projektu dostalo malých změn, kdy funkce pro extrakci blobu byla rozšířena o možnost spojení blízkých blobů do jednoho blobu.

Poslední z uvedené trojice knihoven je malá knihovna `TinyXML`, která disponuje základními funkcemi pro zpracování xml souboru. V tomto projektu byla využita na čtení konfiguračního souboru, který je součástí naší aplikace.

5.3 Ovládání aplikace

Aplikace je navržena jako konzolová aplikace komunikující pomocí příkazového řádku. Aplikace se spouští příkazem:

```
> BSdetektor.exe [nazev_videosouboru.avi]
```

kde `nazev_videosouboru.avi` je nepovinný parametr. Pokud bude aplikace spuštěna bez parametru, pokusí se aplikace získat přístup k video sekvenci z připojené kamery. V případě absence zdroje video sekvence bude aplikace ukončena.

Po úspěšném spuštění se zobrazí úvodní text doplněný o informace o video sekvenci. Pro spuštění detekčního procesu je uživatel vyzván ke zvolení konkrétní detekční metody a volbě zapnutí/vypnutí funkce pro sledování objektů. Svou volbu je nutné vždy potvrdit stisknutím tlačítka **Enter**. Aplikaci je možné v každém kroku ukončit stisknutím tlačítka **Q**. Po volbě detekční metody a volbě sledování objektů je zahájen proces detekce, který je signalizován sadou nově vytvořených oken. Kde každé okno ilustruje jednu z fází detekčního procesu:

- Input – ukazuje aktuální snímek video sekvence
- Output – ukazuje finální výstup detekčního procesu, popř. doplněný o stav sledování

- Background – ukazuje stav modelu pozadí (u multimodální metody MoG jen ten nejpravděpodobnější modul)
- Foreground – ukazuje masku popředí
- Foreground filtering – ukazuje masku popředí vyčištěnou od izolovaných bodů

Proces detekce lze pozastavit a opětovně spustit stisknutím tlačítka P. Ukončení aplikace při probíhajícím procesu detekce je možno vyvolat stisknutím tlačítka Esc. Součástí aplikace je také konfigurační soubor, který dovoluje vlastní nastavení parametrů u implementovaných metod.

Kapitola 6

Testování

V této kapitole se budu snažit shrnout výsledky jednotlivých detekčních algoritmů a poukázat na jejich hlavní přednosti, popřípadě nedostatky. U testování se zaměřím především na vliv změny klíčových parametrů detekčních algoritmů na úspěšnost detekce.

Testovací data jsou tvořena sadou video sekvencí snímající reálnou scénu v rozlišení 384×288 a snímkovací frekvenci 30 snímků za vteřinu.

Výsledky zpracování jsou ověřovány jednak na základě vizuálního porovnávání anebo pomocí srovnávacího skriptu. V oblasti počítačového vidění je snaha napodobit lidské vnímání, proto se dá říci, že vyhodnocení správnosti na základě vizuálního porovnání má největší relevanci. Bohužel tato technika je velmi pomalá. Proto na ověřování většího kvantu dat je nasazeno strojové porovnávání výsledků detekčního algoritmu s referenčními daty. Jelikož však shoda získaných dat s referenčními daty není často binární problém, je nutné výsledky získané pomocí strojového vyhodnocení brát s určitou rezervou. Výsledky účinnosti detekčních metod budou prezentovány pomocí metriky:

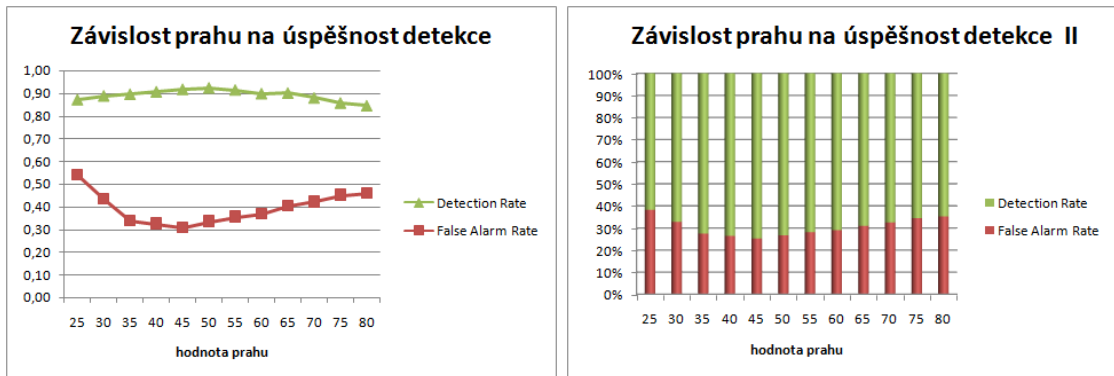
- FN (false negative) vyjadřuje počet nedetekovaných objektů,
- TP (true positive) vyjadřuje počet správně detekovaných objektů,
- FP (false positive) vyjadřuje počet chybně detekovaných objektů,
- FAR (False Alarm Rate) = $FP / (FP + TP)$,
- DR (Detection Rate) = $TP / (TP + FN)$,

kde Detection Rate (DR) vyjadřuje míru počtu správných detekcí k celkovému počtu všech potencionálně možných detekcí, False Alarm Rate (FAR) vyjadřuje míru počtu chybně provedených detekcí k celkovému počtu všech provedeným detekcí. Objekt je správně detekovaný, pokud se jeho obdélníková hranice shoduje s referenčním objektem. Shoda je zde definovaná, tak že těžiště detekovaného obdélníku leží uvnitř referenčního obdélníku. Více informací ohledně metrik a algoritmů pro vyčíslení výkonu detekčních a sledovacích metod lze nalézt v práci [3].

6.1 Hodnota detekčního prahu

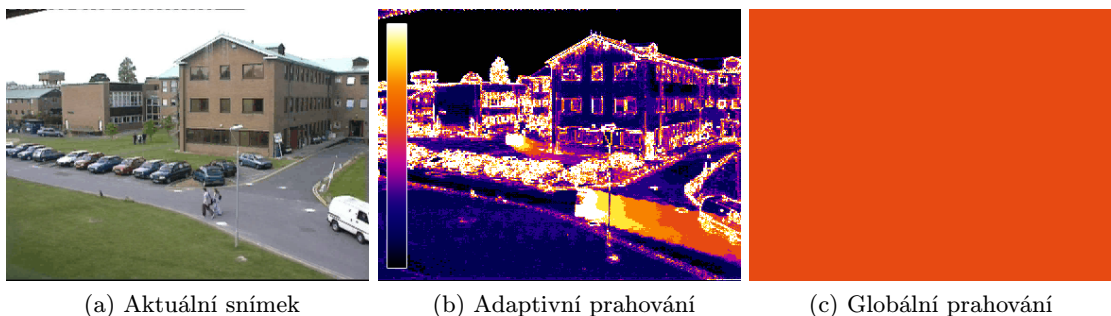
Jeden z klíčových atributů u detekčních metod popsaných v této práci je hodnota prahu. Hodnota prahu určuje míru citlivosti detektoru ke změnám ve scéně. Čím je hodnota prahu větší, tím se detektor stává více imunní vůči změnám ve scéně. Při tomto testu se zaměříme

na vliv změny prahu na účinnost detektoru a pokusíme se určit optimální hodnotu. Jako testovací data byly vzaty hodnoty z intervalu [25; 80], získané hodnoty budou prezentovány dvěma grafy.



Obrázek 6.1: Grafy vyjadřující závislost prahu na úspěšnost detekce

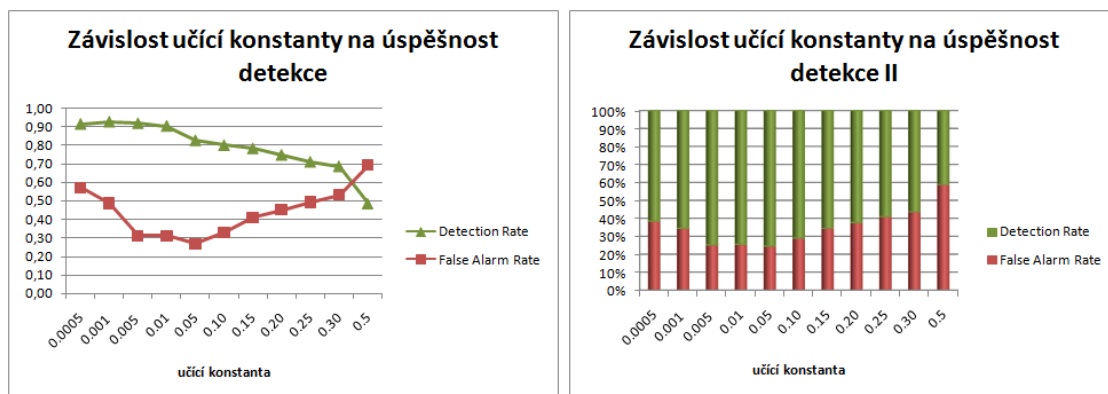
Při testech vyšlo najevo, že volba prahu není triviální záležitostí, jelikož volba je podmíněna přítomností různých jevů a jejich četností ve video sekvenci (jako je šum, odrazy, stíny, tmavá místa ve scéně, nestacionární objekty pozadí, přítomnost pohyblivých objektů s barvou podobné pozadí). Při volbě prahu se často setkáme s nepříjemnou situací, kdy úspěšná eliminace jednoho druhu jevů má za následek zvýšení náchylnosti k jiným nežádoucím jevům. Proto se při výběru snažíme najít určitou rovnováhu mezi jednotlivými eliminacemi nežádoucích jevů. Při testech se ukázalo, že detektor vykazuje nejlepší výsledky pro hodnoty prahu v intervalu [40; 50]. Nutné je podotknout, že hodnota prahu je silně ovlivněna typem snímané scény. Např. u scény, v níž se vyskytují místa se zvýšenou kolísavostí jasové intenzity, je nutné hodnotu prahu zvýšit, tak aby se minimalizoval počet falešných detekcí. Zvýšením prahu je však částečně snížena i citlivost k detekci pohybujících se objektů mající podobnou barvu jako pozadí. V těchto případech se osvědčila strategie využívající lokální (adaptivní) prahování, které umožňuje jednak v oblastech o zvýšené kolísavosti jasové intenzity snížit práh citlivosti a jednak zvýšit práh citlivosti ve stabilních oblastech, jak je ilustrováno na obrázku 6.2.



Obrázek 6.2: Ukázka adaptivního a globálního prahování

6.2 Hodnota učící konstanty

Dalším důležitým atributem, na kterém závisí úspěšnost detekce, je hodnota učící konstanty. Učící konstanta určuje rychlost aktualizace modelu pozadí. Čím je učící konstanta větší, tím se bude vstupní snímek více podílet na podobě modelu pozadí. Pro testování vlivu velikosti učící konstanty na úspěšnost detekce byly zvoleny hodnoty z intervalu $[0,0005; 0,5]$.



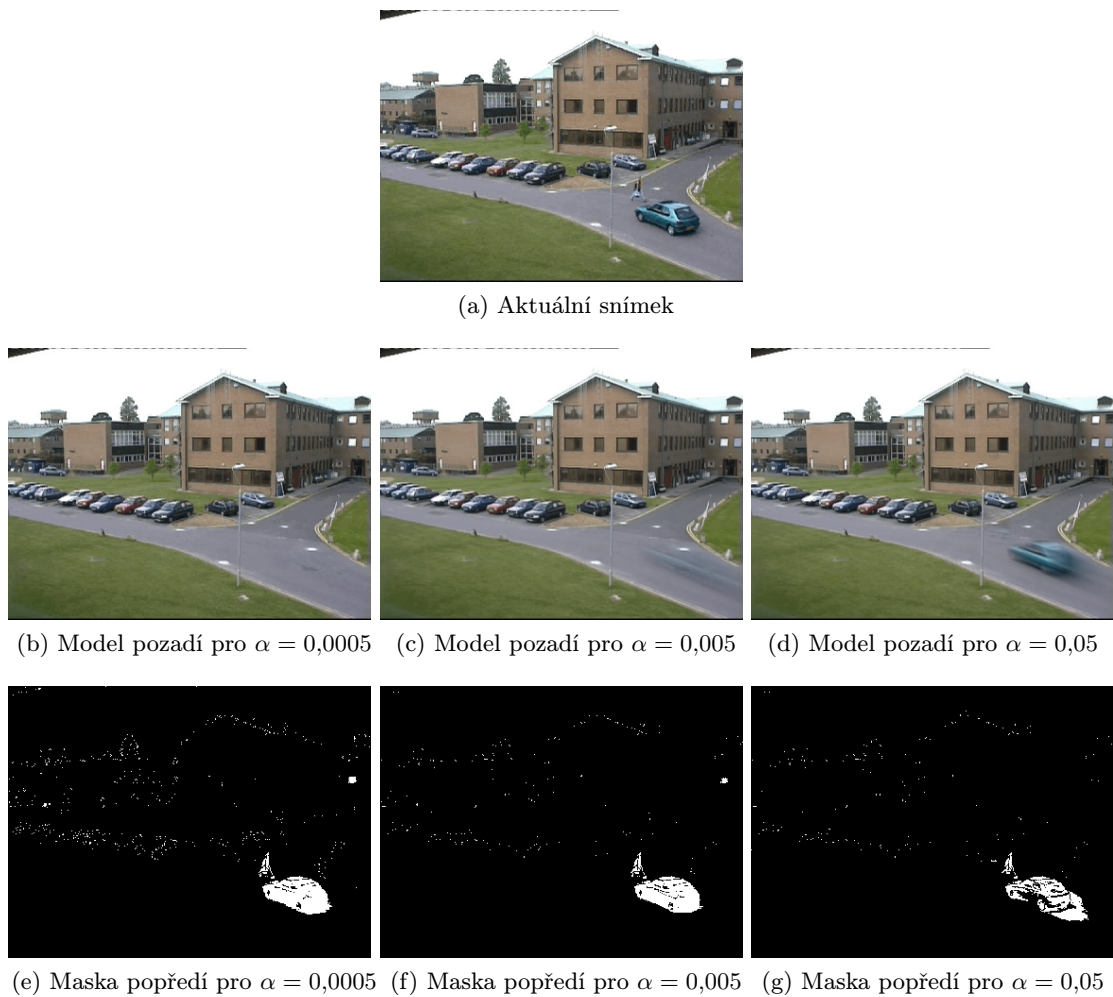
Obrázek 6.3: Grafy vyjadřující závislost učící konstanty na úspěšnost detekce

Při testování se potvrdil přínos ve využití učící konstanty pro modelování pozadí. Která umožňuje lepší reakci na změny osvětlení a na dlouho trvající změny ve scéně. Při velké míře adaptivity však často docházelo k rychlému přesunu pohyblivých objektů do pozadí, což v určitých chvílích způsobilo žádnou či jen částečnou detekci. Tento jev byl patrnější u velkých a pomalu se pohybujících objektů. Naopak čím byla učící konstanta menší, tím byl model pozadí stabilnější vůči změnám ve scéně. Při testu se jako přijatelný kompromis mezi časovou stabilitou a adaptivitou pro učící konstantu ukázal interval $[0,005; 0,05]$. U aktualizace pozadí se osvědčila strategie selektivního modelování pozadí, při níž dochází k aktualizaci pozadí podle klasifikace bodů vstupního snímku. Pokud je obrazový bod klasifikován jako popředí, podílí se na modelu pozadí menší mírou než body klasifikované jako pozadí.

6.3 Detekce pohyblivých objektů ve venkovní scéně

Detekce pohyblivých objektů ve venkovní scéně je poměrně komplikovaný úkol kvůli velkým změnám v osvětlení a velké četnosti výskytu nestacionárních objektů pozadí (větve stromů, traviny, keře, světla semaforu), jejichž pohyb by neměl být detekován. V tomto místě se jako nejvhodnější ukázalo nasazení multimodálního modelu pozadí – zde Směsice Gaussových křivek, která využívá několik gaussovek pro popis pohybu nestacionárních objektů pozadí.

Při použití unimodálního modelu pozadí, jako je např. Klouzavý Gaussův průměr, je nutné pro eliminaci falešných detekcí způsobené nestacionárními objekty pozadí zvýšit práh detekce, což však vede k celkovému snížení citlivosti detektoru.



Obrázek 6.4: Ukázka vlivu učící konstanty na model pozadí a následnou úspěšnost detekce

6.4 Detekce pohyblivých objektů ve vnitřní scéně

Detekce pohybu ve video sekvenci, jež zachycuje vnitřní scénu, je o poznání jednodušší, jelikož se vyznačuje stabilnějším prostředím, avšak není to vždy podmínkou. Hladký průběh detekce ve vnitřním prostředí mohou narušit takové jevy, jako je pulsující zářivka, lesklé obložení chodeb či vržené stíny. Pro eliminaci těchto jevů se často využívají stejné mechanismy jako u venkovní scény.

6.5 Sledování pohyblivých objektů

Jelikož sledovací algoritmus byl navržen s důrazem na co nejmenší výpočetní náročnost, je úspěšnost sledovacího algoritmu silně závislá na kvalitě dat získaných v detekčním procesu. Algoritmus zvládá sledovat více objektů na jednou a implementační omezení na maximální počet sledovaných objektů v jeden okamžik se na testovaných video sekvencích nejeví jako příliš velké omezení. Vliv tohoto omezení je navíc ještě snížen přednostním výběrem větších objektů pro sledování.

Při sledování dochází k velkému nárůstu počtu registrovaných objektů v průběhu celé video sekvence, což se projevuje na vysokém identifikačním čísle ID. Na velikosti tohoto čísla se převážně podílejí falešně detekované objekty (jejich životnost je jen pár snímků), přerušená identita objektů, která je způsobená nekvalitní extrakcí objektů v detekčním procesu a malou schopností sledovacího algoritmu řešit vzájemné překrývání objektů. Dalším faktorem, který se podílí na vysokém ID čísle je absence možnosti uchovávat si ztracené objekty po několik dalších snímků.

Ač při vývoji sledovacího algoritmu bylo provedeno několik úspěšných kroků pro snížení počtu sledovaných falešných objektů, tak je to stále nedostačující. V průběhu dalšího vývoje je proto prioritní pokračovat ve zdokonalování sledovací metody a to i případě, že by to znamenalo přehodnocení výpočetních nároků na sledovací algoritmus.

6.6 Výpočetní náročnost

Pro nasazení detekční metody v reálném čase je nezbytné, aby detekční algoritmus byl schopen zpracovat alespoň 3 až 5 snímků za vteřinu. Výpočetní náročnost jednotlivých metod je ukázána v tabulce 6.1. Pokud detektor není primárně určen pro nasazení v náročných venkovních scénách, je z pohledu výpočetní náročnosti vhodné sáhnout po unimodálních metodách jako je např. Klouzavý Gaussův průměr, jejichž výpočetní nároky jsou znatelně menší než u multimodálních metod.

Detekční metoda	Počet snímků	Čas zpracování	Snímků za vteřinu
Průměr posledních 8 snímků	2688	3:47	12
Klouzavý průměr	2688	1:24	32
Klouzavý průměr PM	2688	1:24	32
Klouzavý Gaussův průměr	2688	1:24	32
Klouzavý Gaussův průměr PM	2688	1:26	31
Směsice Gaussových křivek	2688	8:32	5

Tabulka 6.1: Výpočetní náročnost detekčních metod

Kapitola 7

Závěr

Cílem této bakalářské práce bylo popsat principy detekce pohyblivých objektů ve videu. Oblast detekce pohyblivých objektů se ukázala jako rozsáhlá oblast, proto se tato práce blíže zaměřila pouze na jednu část a tou jsou detekční metody založené na principu odečítání pozadí, při čemž největší pozornost byla věnována přístupům k modelování pozadí a jejich aktualizace a technikám detekce popředí, které se ukázaly jako klíčové z pohledu celkové kvality detekce.

Většina popsaných metod byla naimplementována a následně otestována na řadě video sekvencí. Výsledky ukázaly, že techniky pracující na principu odečítání pozadí se právem těší velké popularitě v odborné komunitě zabývající se počítačovým viděním. Z pohledu výpočetní náročnost/kvalita se na prvním místě z testovaných metod umístila metoda Klouzavý Gaussův průměr, která disponuje kvalitním aparátem, který zajišťuje dostatečnou adaptivitu k okolí a dostatečně vysokou citlivost v detekci.

V práci je také popsána jedna multimodální detekční metoda MoG, která ukazuje nový přístup k modelování pozadí, kterým lze poměrně úspěšně řešit problém pohybujícího se pozadí, aniž by byla razantně snížena citlivost detekce. Nevýhodou této metody jsou zvýšené výpočetní nároky.

Budoucí rozšíření může probíhat více směry. Jeden z prioritních je rozšíření a zdokonalení sledovacího systému především o predikci polohy sledovaného objektu pro následující snímek, např. pomocí *Kalmanova filtru*, možnost uchovávat si ztracené objekty po několika dalších snímků pro navázání identity, vylepšení řízení překryvu sledovaných objektů. Dalším rozšířením by mohla být klasifikace detekovaných objektů, která by nám poskytla komplexnější informace o snímané scéně. V neposlední řadě by také nebylo od věci rozšířit záběr této práce a pokračovat ve zkoumání dalších metod založených na odečítání pozadí, popřípadě na jiných přístupech, které by poskytovalo podrobné a ucelené informace o této bezesporu velmi zajímavé oblasti.

Literatura

- [1] Wikipedie: Otevřená encyklopedie: Barevný model [online]. 2009-09-19 [cit. 2010-05-04]. Dostupný z WWW: http://cs.wikipedia.org/wiki/Barevný_model.
- [2] Ali, M. A.; Indupalli, S.; Boufama, B.: Tracking Multiple People for Video Surveillance [online]. 2007 [cit. 2010-05-04]. Dostupný z WWW: <http://computer-vision.org/4security/pdf/windsor.pdf>.
- [3] Bashir, F.; Porikli, F.: Performance evaluation of object detection and tracking systems. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2006.
- [4] Bouwmans, T.; El Baf, F.; Vachon, B.: Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. *Recent Patents on Computer Science* 1, ročník 3, 2008: s. 219–237.
- [5] Carmona, E. J.; Martínez-Cantos, J.; Mira, J.: A new video segmentation method of moving objects based on blob-level knowledge. *Pattern Recognition Letters*, ročník 29, č. 3, February 2008: s. 272–285.
- [6] Cheung, S.-C. S.; Kamath, C.: Robust techniques for background subtraction in urban traffic video. *Visual Communications and Image Processing 2004*, ročník 5308, 2004: s. 881–892.
- [7] Collins, R.: Lecture 28: Intro to Tracking [online]. 2007 [cit. 2010-05-04]. Dostupný z WWW: <http://www.cse.psu.edu/~rcollins/CSE486/lecture28.pdf>.
- [8] KaewTraKulPong, P.; Bowden, R.: An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. *Proceedings 2nd European Workshop on Advanced Video Based Surveillance Systems (AVBS01)*, September 2001.
- [9] Narayana, M.: Automatic Tracking of Moving Objects in Video for Surveillance Applications [online]. 2007 [cit. 2010-05-04]. Dostupný z WWW: <http://vis-www.cs.umass.edu/~narayana/NarayanaMSThesisDefense.ppt>.
- [10] Shireen, Y. E.; Khaled, M. E.-S.; Sumaya, H. A.: Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art. *Recent Patents on Computer Science*, ročník 1, 2008: s. 32–54.
- [11] Stauffer, C.; Grimson, W. E. L.: Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 2, Januar 1999: s. 246–252.

- [12] Zang, Q.; Klette, R.: Parameter Analysis for Mixture of Gaussians Model. *CITR Technical Report 188*, 2006.
- [13] Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, ročník 2, 2004: s. 28–31.
- [14] Žára, J.; Beneš, B.; Sochor, J.; aj.: *Moderní počítačová grafika*. Computer Press, druhé vydání, 2005, iISBN 80-251-0454-0.

Dodatek A

Obsah přiloženého DVD

- Zdrojové soubory programu
- Knihovny potřebné k přeložení aplikace
- Přeložená aplikace pro Windows
- Konfigurační soubor
- Manuál k ovládání aplikace
- Testovací videa
- Videokázky činnosti detektoru
- Text bakalářské práce v .pdf a .tex