



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 POHYBU V OBRAZE Z KAMERY

MOTION DETECTION IN VIDEO

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR POLANSKÝ

VEDOUcí PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Polanský Petr**

Obor: Informační technologie

Téma: **Detekce pohybu v obraze z kamery**
Motion Detection in Video

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s problematikou odečítání pozadí a detekce pohybu v obraze.
2. Vytvořte nástroj pro pořizování datové sady s různými druhy pohybu.
3. Pořídte a anotujte vhodnou datovou sadu s různými druhy a intenzitami pohybu z více kamer.
4. Vyhledejte a vyvíňte algoritmy pro detekci pohybu ve videu ze stacionární kamery.
5. Integrujte vyvinuté algoritmy do aplikace, jež v reálném čase bude detekovat pohyb a výsledky detekce vhodně zasílat/vizualizovat.
6. Na datové sadě vyhodnoťte vlastnosti vytvořené aplikace a srovnajte ji s alternativami.
7. Zhodnoťte dosažené výsledky a navrhnete možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 až 5.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Herout Adam, doc. Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božská Schova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem této práce je zhodnotit účinnost detekce pohybu pomocí Gaussova rozdělení pravděpodobnosti. Algoritmus při detekci pohybu vytvoří krátké video zachycující pohyb a vhodně jej vizualizuje. Vizualizace je provedena grafem intenzity bílých pixelů mapující pohyb. Systém je vhodný hlavně pro méně rušné oblasti, kde pohyb je významnou změnou prostředí. Výsledky této práce ukazují, jaký vliv na detekci mají okolní podmínky a poloha kamery.

Abstract

The objective of this work is evaluate motion detection using Gaussian Mixture Model. When algorithm detects motion, it creates short video capturing this motion and visualizing it properly. Visualization is made by white pixels intensity graph. System is applicable on less busy area when motion is more noticeable. Results of this work shows how surrounding environment and camera position influence detection.

Klíčová slova

detekce pohybu, odečítání pozadí, rozlišení pohybu, směsice Gaussových křivek, eroze, dilatace, otevření, uzavření, oblast zájmu, ROI, prahování, ROC křivka, Cannyho hranový detektor, OpenCV, intenzita pohybu, podmínky sledování, druhy kamer

Keywords

motion detection, background subtraction, Gaussian Mixture Model, erosion, dilation, opening, closing, region of interest, ROI, threshold, ROC curve, Canny edge detector, OpenCV, motion intensity, surveillance conditions, camera types

Citace

POLANSKÝ, Petr. *Detekce pohybu v obraze z kamery*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Herout Adam.

Detekce pohybu v obraze z kamery

Prohlášení

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

.....
Petr Polanský
17. května 2016

Poděkování

Tímto bych chtěl poděkovat hlavně prof. Heroutovi za jeho vstřícnost, ochotu a trpělivost.

© Petr Polanský, 2016.

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	2
2	Sledovací systémy	3
2.1	Podmínky sledování	3
2.2	Druhy sledovacích kamer	4
3	Techniky pro detekci pohybu v obraze	6
3.1	Existující řešení problematiky	6
3.2	Odečítání pozadí	9
3.3	Prahování	13
3.4	Cannyho hranový detektor	13
3.5	Morfologické operátory	15
3.6	Specifikace oblasti zájmu	17
4	Návrh detekce pohybu v obraze	18
4.1	Nalezení změny v obraze	18
4.2	Vizualizace výsledků	19
4.3	Testování	19
5	Implementace a vyhodnocení	25
5.1	Implementace	25
5.2	Spuštění aplikace	26
5.3	Vyhodnocení výsledků	26
6	Závěr	28
	Literatura	29
A	Obsah CD	30

Kapitola 1

Úvod

Detekce pohybu je v dnešní době aktuálním tématem. Vždyť jednoduchou cenově dostupnou kameru s ukládáním záznamu na disk je možné koupit v každém obchodě s elektronikou. Kamerový systém však lze využít i jinak než jen pro ochranu majetku. Kamerové systémy dokáží mimo jiné odlišit člověka od jiného objektu, správně detekovat registrační značku auta nebo rozpoznat obličej člověka. Pokročilejší kamerové systémy přináší mnoho funkcí zejména pro velká obchodní centra. Například dokáží z pozorování zákazníků vysledovat, u kterých regálů tráví zákazníci nejvíce času. Kamerové systémy také umožňují majiteli spravovat více kamer současně. Pro jednodušší detekci pohybu se používají fotopasti, které pořídí snímek zachycující pohyb.

Ve své práci jsem se zaměřil na sledování osob v uzavřených prostorách, kde výskyt pohybu je třeba dobře detekovat zejména z hlediska bezpečnosti osob a majetku. V druhé kapitole jsme popsali, jaká práva a povinnosti plynou pro majitele kamerových systémů. Dále jsem popsal typy často používaných kamer a existující řešení, kterými jsem se inspiroval. Popsal jsem také metody, které k detekci pohybu slouží. Ve své práci jsem popsal nejen postup k dosažení co nejlepších výsledků detekce pohybu, ale také různé anomálie, se kterými se algoritmus dokáže či nedokáže vypořádat. Výstupem programu jsou krátká videa, která zobrazují zachycený pohyb a pomocí grafu informují o intenzitě pohybu.

Kapitola 2

Sledovací systémy

V době internetu a obrovské rozšířené technologii jsou systémy sledující pohyb velkým tématem s mnoha klady i zápory. Téměř na každé ulici si můžete všimnout instalovaného kamerového systému. V dnešní době je navíc tato ochrana soukromí či majetku cenově dostupná téměř každému.

2.1 Podmínky sledování

Kamerový systém je uznáván Úřadem pro ochranu osobních údajů (dále jen Úřad) za legitimní a legální možnost k ochraně zdraví osob a majetku. Na místě, které je sledované kamerami, musí být upozornění jako je např. na obrázku 2.1, které kolemjdoucí informuje, že tento prostor je sledován kamerovým systémem. Provozování kamerového systému je považováno za zpracování osobních údajů, pokud je pořizován záznam těchto záběrů. V tomto případě je nutné, aby byl systém zapsán do registru zpracování osobních údajů vedeného Úřadem a na majitele kamerového systému se vztahuje oznamovací povinnost. V případě, že majitel kamerového systému za účelem ochrany svého majetku monitoruje svůj soukromý pozemek, oznamovací povinnost se na něj nevztahuje. Není přípustné, aby kamery snímaly prostory sloužící k intimním činnostem, jako jsou např. veřejné umývárny a toalety. Záznamy kamer musí sloužit pouze pro předem specifikované účely, popřípadě pro účely související s veřejným zájmem, mezi které patří boj s pouliční kriminalitou. Je třeba zajistit ochranu snímacích zařízení a nosičů se záznamy, aby nedošlo k odcizení či



Obrázek 2.1: Ukázka jak může vypadat upozornění na sledování v dané oblasti

zničení záznamu. Lhůta, po kterou může být záznam uchovávan, by neměla být delší, než je nezbytně nutné pro splnění účelu kamerového sledování. Zákon sice nespecifikuje, jaká je přiměřená doba uchovávání záznamu, nicméně Úřad za takovou dobu zpravidla považuje třídní uchovávání. I v případě, že je sledování prováděno obecní nebo státní policií, nelze snímky zveřejňovat nebo vkládat na webové stránky, a to ani na webové stránky města. Skryté sledování soukromých prostor je pak možné pouze v nezbytně nutných případech a za podmínek stanovených trestním právem. V případě monitorování zaměstnanců je zaměstnavatel podle ust. § 316 odst. 3 zákoníku práce povinen zaměstnance přímo informovat o rozsahu kontroly a způsobech jejího provádění, pro jaký účel budou osobní údaje zpracovávány, kdo a jak je bude zpracovávat a komu mohou být zpřístupněny.

2.2 Druhy sledovacích kamer

Analogové kamery

Tyto kamery využívají pro přenos obrazu analogový signál nejčastěji koaxiálním kabelem. Data se zpracovávají až v nahrávacím zařízení (DVR)¹, kde dojde k digitalizaci a případnému uložení dat na pevný disk. Výhoda této technologie spočívá v možnosti kombinace s kamerami různých výrobců, protože většina z nich jsou kompatibilní. Dále jsou ze všech typů kamer nejcitlivější na světlo a jejich pořizovací cena také patří k nejnižším ze všech typů kamer.

HD-SDI

HD-SDI² kamerové systémy jsou nejčastěji vhodné pro přestavbu starých analogových systémů. Na druhou stranu jsou velmi citlivé na kvalitu vedení. Pokud se na trase vyskytuje mnoho spojek či konektorů, dochází k velkým útlumům signálů. V dnešní době lze po koaxiálním kabelu přenášet jak obraz, tak i napájení.

IP kamery

IP kamery se liší oproti předcházejícím hlavně tím, že kamera v sobě má zabudovaný procesor, čímž může docházet ke zpracování obrazu již na straně kamery a k nahrávacímu zařízení se dostane již zpracovaný signál. Nejčastěji jsou používány kompresní formáty MJPEG, MPEG-4 a H.264. Nastavení kamer probíhá přes webové rozhraní v internetovém prohlížeči, kde je také možné sledovat zachycený obraz z kamery. Nejčastěji zařízení operují s rozlišením obrazu 1,3 Mpix nebo FullHD, které je 2Mpix. Pro přenos se používá LAN síť nebo WiFi rozhraní. WiFi připojení má však rovněž své nevýhody, jako jsou například ztráty paketů při přenosu. Pokud jsou kamery zapojeny do stávající počítačové sítě LAN, není třeba řešit vzdálenost kamer, ale spíše propustnost sítě. Uložení záznamu z kamery je možné několika způsoby. Záznam lze ukládat na vnitřní SD³ kartu v kameře, která je však omezená svojí kapacitou a tím pádem je omezená i délka záznamu. Dále je možné nechat záznam ukládat přímo do PC pomocí softwaru, který k tomuto účelu obvykle výrobci poskytují, nebo ukládat na FTP server, čímž nedojde k poškození dat v případě poškození fyzického úložiště. Díky vysokému rozlišení jsou IP kamery vhodné např. pro identifikaci osob, automobilů včetně jejich registračních značek nebo pro rozpoznání lidských obličejů.

¹Digital Video Recorder

²High Definition Serial Digital Interface

³Secure Digital

Nevýhody IP kamer spočívají ve zpoždění obrazu kvůli zpracování v kameře. Nekompatibilita kamer různých výrobců je částečně vyřešena ONVIF standardem. Může také docházet k horší datové propustnosti u sdílených tras, kde kamery používají stejné trasy přenosu.

Kapitola 3

Techniky pro detekci pohybu v obraze

V této kapitole se chytám popsat, jaká řešení tohoto tématu již existují, a kterými jsem se rozhodl inspirovat. Dále popíši jednotlivé algoritmy, které souvisí s touto tematikou detekce pohybu.

3.1 Existující řešení problematiky

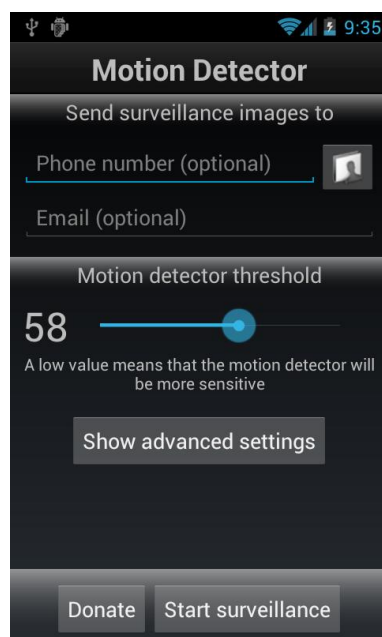
Pro účely ochrany majetku nebo sledování pohybu osob byla vytvořena spousta mechanismů, jak toho docílit. Patří mezi ně statické kamery, které se instalují na domy. Jednodušší detekci pohybu lze docílit pomocí fotopastí, které pořídí snímek okolí v případě, že se v něm vyskytne pohyb. Tato metoda se využívá např. pro doložení existence některých vzácných druhů zvířat. Dále existují termokamery, jejichž výstupem je infračervený snímek, zachycující intenzitu tepla na snímku. Já jsem se rozhodl popsat existující softwarové programy, které dokáží problém identifikovat a popřípadě i rozlišit.

Motion Detector Pro

Tato mobilní aplikace je vytvořena pro operační systém Android. V případě detekce pohybu, pomocí vestavěné kamery v mobilu, je uživateli poslán snímek s pohybem na emailovou adresu nebo pomocí textové zprávy na jiné telefonní číslo. Jak je ukázáno na obrázku 3.1, uživatel si může nastavit úroveň prahu detekce, která ovlivňuje, na jak moc velký pohyb chce uživatel dostat upozornění. Mezi další funkce patří uložení snímku do paměti telefonu nebo na virtuální uložení. Po nejnovější aktualizaci dokáže program detekovat pohyb také pomocí přední kamery telefonu.

AXIS Video Motion Detection

Axis Communication (dále jen Axis) je jedna z největších firem zabývajících se kamerovou bezpečností. Nabízí velké množství kamer, které jsou specifické pro určitou činnost, z nichž některé jsou ukázány na obrázku 3.2. Například kamery do dopravních prostředků jsou odolné vůči prachu, vodě, otřesům i výkyvům teplot. Dále poskytují kamery se zvýšenou odolností vůči větru pomocí elektronické stabilizace obrazu a kamery pro dokonalou detekci i v noci jsou vybaveny nočním viděním. Některé jejich kamery používají metodu Zipstream, pomocí které jsou objekty ve videu rozlišeny na základě jejich důležitosti. Tento



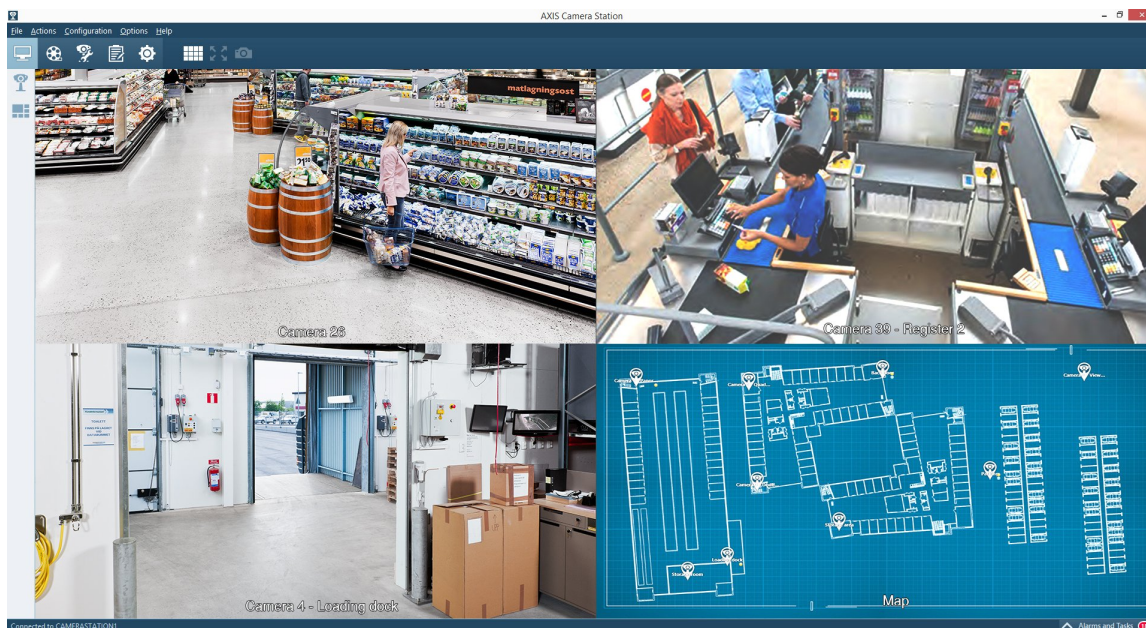
Obrázek 3.1: Nastavení způsobu detekce u Motion Detector Pro, mezi které patří velikost prahu nebo telefonní číslo pro poslání upozornění.

přístup slouží pro lepší využití šířky pásma a paměti. Důležité detaily, mezi které patří obličej člověka, tetování nebo státní poznávací značka, jsou odděleny a uloženy ve vysokém rozlišení, na rozdíl od bílých zdí, trávníků nebo vegetace, které nejsou ve většině případů podstatné.

Kromě zaručení bezpečnosti a ochrany, poskytuje firma Axis i služby na podporu managementu. Například pokud zákazník dlouho stojí před některým z regálů, systém informuje prodavače, že zákazník je nejspíš nejistý a potřeboval by poradit. Zejména ve velkých obchodech se hodí i informace pro prodavače, že v regálech chybí zboží, nebo že se tvoří velké fronty u pokladen a je potřeba zpřístupnit další. Dále dochází ke shromažďování statistik zájmu o jednotlivé regály v obchodě podle doby, kterou zákazníci u nich tráví. Aplikace pro sledování záznamu se jmenuje Axis Camera Station a je ukázána na obrázku 3.3. Veškeré



Obrázek 3.2: Některé z typů kamer, které poskytuje firma Axis Communication, vlevo standardní kopulovitá, uprostřed termínální a vpravo velmi diskrétní, sestávající se ze zdířky pro LAN, napájení a slotu pro paměťovou kartu.



Obrázek 3.3: Ukázka AXIS Camera Station, která kromě záznamů z více kamer zároveň, zobrazuje i plánec budovy s označenými místy, kde se vyskytují kamery.

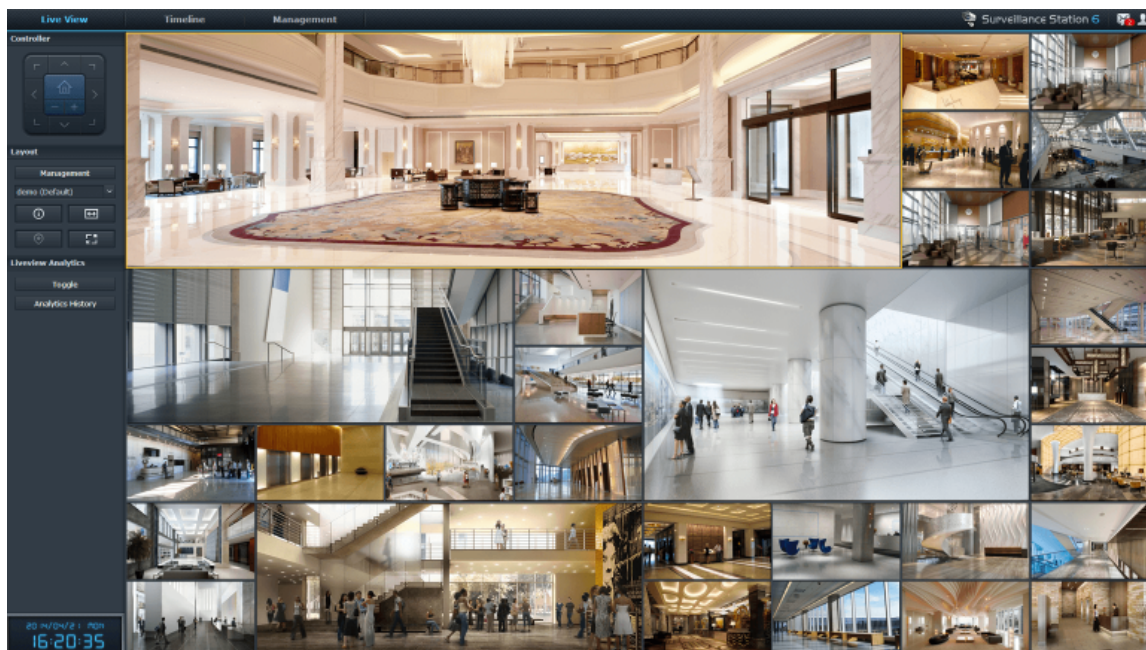
záznamy z kamer je možné sledovat na mobilu či tabletu a to i záznamy kamer z jiných poboček konkrétní firmy. Mezi další služby patří detekce překročení virtuální hranice např. při přezení plotu nebo sledování specifické osoby.

Synology

Společnost Synology poskytuje obsluhu pro kamerové systémy a jejich spravování. Jejich Surveillance Station, jak je možné vidět na obrázku 3.4, umožňuje současně sledovat záznamy z různých kamer. Přehrát lze video i po jednotlivých snímcích pro lepší detekci změn v obraze. Snímky záznamu lze později upravovat změnami jasu, kontrastu a ostrosti, pro lepší obraz.

Nahrávání lze nastavit ve více režimech. Uživatel může nahrávání nastavit ručně nebo může nastavit, aby se spustilo automaticky po vyvolání určité události. Pokud se přeruší spojení se serverem, může kamera nahrávat na svoji SD kartu. Po obnovení síťového spojení se videa přepokopírují na server. Záznamy pořízené uvnitř pohybujícího se vozidla se mohou ukládat na úložiště uvnitř kamery. Až se dopravní prostředek na konci dne vrátí do stanice, všechna videa budou stažena pomocí bezdrátového přenosu na server. V případě odpojení určité kamery od sítě je možné nastavit, aby se spustilo nahrávání jinou kamerou, snímající stejný prostor. Všechny záznamy jsou archivovány a opatřeny neviditelným digitálním podpisem. Nástroj Evidence Integrity Authenticator může ověřit a potvrdit, že se záznamem nebylo žádným způsobem manipulováno a jedná se o originál. Toto je vhodné zejména u soudních sporů, kde jsou jako přípustné brány pouze ověřené důkazy.

Synology navíc spolupracuje s prodejci IP kamer a podporuje více než 4000 kamer různých výrobců. Integruje pokročilé funkce kamer včetně digitálního vstupu, obousměrného zvuku, vícenásobných datových proudů, TV standardů, zorného pole, ovládání přenosové rychlosti a protokolů RTSP/RTP. Stejně jako u Axis je možné záznamy sledovat na mobilu nebo tabletu na operačních systémech iOS a Android.



Obrázek 3.4: Ukázka Synology Surveillance Station ukazující sledování záznamů z více kamer současně.

3.2 Odečítání pozadí

Tato technika se používá při detekci pohybu pro odlišení popředí snímku, které obsahuje pohyb a pozadí, které je neměnné. Obrázek 3.5 ukazuje, jak může vypadat pozadí a popředí. Existuje mnoho algoritmů jak pomocí odečítání pozadí docílit detekce pohybu. Jednotlivé algoritmy se liší svojí složitostí, rychlostí, reakcí na nečekané změny v obraze a paměťovou náročností. Některé další techniky včetně detailnějšího zhodnocení jejich vlastností popsal Piccardi [9].



Obrázek 3.5: Vlevo je znázorněn snímek, který by mohl být určen jako pozadí. Snímek vpravo je aktuální snímek, na kterém se vyskytuje pohyb člověka a bude tedy posouzen jako snímek popředí.

Metoda rozdílu dvou snímků

Metoda detekuje pohyb pomocí porovnání snímku, který je brán jako pozadí a snímku, který se v čase liší. Tento princip funguje pouze v případě, kdy všechny pixely v popředí se pohybují a všechny pixely pozadí jsou statické.

$$|I(x, y, t) - B(x, y, t - 1)| > Th \quad (3.1)$$

Rovnice (3.1) popisuje tento postup, přičemž $I(x, y, t)$ značí hodnotu intenzity pixelu na konkrétní pozici (x, y) v čase t , od které je odečtena hodnota pixelu pozadí $B(x, y, t)$ ze stejné pozice. Nastavením proměnné Th říkáme, k jak moc velkému rozdílu intenzit mezi oběma pixely musí dojít, aby byl vyznačen pohyb.

Metoda se používá hlavně v jednodušších aplikacích, jelikož je velmi citlivá na šum a změny světla. Tohoto přístupu využívá Lipton [6], který navíc rozlišuje jednotlivé objekty na lidi nebo vozidla a následně sleduje trajektorie takto zachycených objektů.

Mean filter

V tomto případě je vybráno N snímků, uložených v bufferu, ze kterých je sestaveno pozadí pomocí střední hodnoty. Z každého snímku z bufferu jsou vybrány hodnoty pixelů ze stejné pozice (x, y) a z nich je vypočítána střední hodnota, která představuje hodnotu pixelu pozadí na stejné souřadnici (x, y) .

$$B(x, y, t) = \frac{1}{N} \sum_{i=0}^{N-1} I(x, y, t - i) \quad (3.2)$$

$$|I(x, y, t) - \frac{1}{N} \sum_{i=0}^{N-1} I(x, y, t - i)| > Th \quad (3.3)$$

Rovnice (3.2) popisuje tvorbu pozadí a rovnice (3.3) odečítání takto vytvořeného pozadí od aktuálního snímku.

Median filter

Princip tohoto postupu, popsaného rovnicemi (3.4) a (3.5), spočívá v použití hodnot mediánů posledních N snímků pro model pozadí. V pozdějším vylepšení metody se dle Piccardi [9] navrhuje vypočítat medián ze sady hodnot skládající se z posledních N snímků a z W posledních vypočítaných mediánů.

$$B(x, y, t) = \text{median}\{I(x, y, t - i)\} \quad (3.4)$$

$$|I(x, y, t) - \text{median}\{I(x, y, t - i)\}| > Th, \quad (3.5)$$

kde $i \in \{0, \dots, N - 1\}$ určuje z kolika předchozích snímků bude medián vypočítán.

Hlavní nevýhoda tohoto postupu spočívá v nutnosti použití vyrovnávací paměti na uložení starých snímků. S možným řešením tohoto problému přišli McFarlane a Schofield [7], kteří navrhuje, aby bylo v bufferu uchováno pouze aktuální pozadí a intenzity pixelů jsou inkrementovány o 1, pokud je současný pixel na dané pozici světlejší nebo dekrementovány o 1, pokud je tmavší. Další nevýhoda je patrná při pomalu se pohybujícím objektu se stejnou intenzitou pixelů, jakou má pozadí, což může mít za následek splynutí pohybujícího se objektu s pozadím.

Klouzavý Gaussův průměr

Klouzavý Gaussův průměr patří mezi rekurzivní techniky, tedy na rozdíl od předchozích postupů pro výpočet pozadí nepoužívá buffer, ale rekurzivně aktualizuje model pozadí. Je to založeno na správném umístění jedné Gaussovy funkce hustoty pravděpodobnosti (Gpdf)¹ na každý pixel posledních N snímků. Gpdf je charakterizovaná pro každý pixel rozptylem σ^2 a střední hodnotou μ , které jsou aktualizovány v čase pomocí klouzavého průměru následovně:

$$\mu_{t+1} = \alpha I_t + (1 - \alpha)\mu_t \quad (3.6)$$

$$\sigma_{t+1}^2 = \alpha(I_t - \mu_t)^2 + (1 - \alpha)\sigma_t^2 \quad (3.7)$$

$$|(I_t - \mu_t)| > k\sigma \quad (3.8)$$

Funkcí (3.8) lze určit, zda pixel patří mezi popředí nebo mezi pozadí. k je práh nejčastěji nabývající hodnotu 2,5 a slouží pro určení, zda pixel leží v dané Gaussově křivce. α je učící parametr. Pokud (3.8) platí, je pixel označen jako součást popředí v opačném případě jako součást pozadí. Mezi nevýhody tohoto algoritmu patří zhoršená detekce v místech s velkým množstvím pohybu vyskytující se naráz.

Směsice Gaussových křivek

Principem tohoto postupu je určení, zda pixel obrazu je součástí popředí či pozadí pomocí směsi N Gaussových křivek, které modelují každý pixel. Nejčastěji je jako N zvolena hodnota mezi 3 a 5. V jakémkoliv čase t má konkrétní pixel (x_0, y_0) svoji historii:

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}. \quad (3.9)$$

Pro správné reagování na změny okolí, můžeme historii aktualizovat přidáním nových a odstraněním starých hodnot v průběhu času. Pravděpodobnost, že aktuální pixel nabývá dané hodnoty, je vyjádřena vztahem:

$$P(X_t) = \sum_{i=1}^N \omega_{i,t} \mathcal{N}(X_t | \mu_{i,t}, \Sigma_{i,t}), \quad (3.10)$$

kde $\Sigma_{i,t}$ je kovarianční matice i -té Gaussovy křivky a $\omega_{i,t}$ je váha křivky v čase t , přičemž součet vah všech křivek musí být roven 1. $\mathcal{N}(X_t | \mu_{i,t}, \Sigma_{i,t})$ je Gaussova funkce hustoty pravděpodobnosti konkrétního pixelu X v čase t , která se určí pomocí vztahu:

$$\mathcal{N}(X_t | \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_{i,t}|^{1/2}} e^{-\frac{1}{2}(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})}. \quad (3.11)$$

Pro aktualizaci parametrů je možné využít algoritmus EM², který používá například Dar-Shyang Lee [5]. Tento přístup však podle Šonka, Hlaváč, Boyle [8] může výpočet výrazně zpomalovat. Jiný postup porovnává pixel s každou křivkou ze směsice křivek, která modeluje daný pixel. Pokud je odchylka od střední hodnoty křivky menší než 2,5, potom se pixel s danou křivkou shoduje. Pokud je nalezeno více shod, potom je vybrána ta nejlepší. Rekurzivní provádění slouží pro neustálou aktualizaci parametrů. Váha, rozptyl a střední hodnoty jsou upraveny následovně:

$$\omega_{i,t+1} = (1 - \rho)\omega_{i,t} + \alpha(M_{i,t+1}), \quad (3.12)$$

¹Gaussian probability density function

²algoritmus Expectation-Maximization

kde $M_{i,t+1} = 1$ pro Gaussovou křivku, se kterou došlo ke shodě, a $M_{i,t+1} = 0$ pro ostatní neshodující se křivky.

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho X_{t+1} \quad (3.13)$$

$$\sigma_{i,t+1}^2 = (1 - \rho)\sigma_{i,t}^2 + \rho(X_{t+1} - \mu_{i,t+1})^2, \quad (3.14)$$

kde

$$\rho = \alpha \mathcal{N}(X_{t+1} | \mu_{i,t}, \sigma_{i,t}^2) \quad (3.15)$$

a α je učicí parametr, kde $1/\alpha$ určuje jakou rychlostí se budou parametry měnit. Pokud není nalezena shoda ani s jednou křivkou, potom je Gaussova křivka s nejmenší vahou (nebo poměrem ω/σ) v daném pixelu odstraněna a nahrazena novou s parametry:

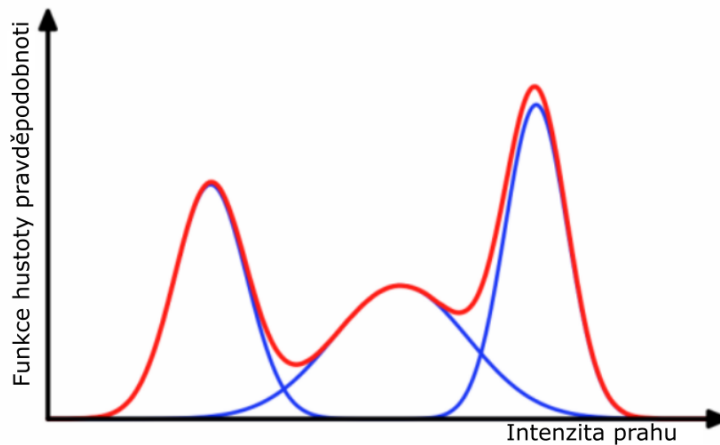
$$\omega_{t+1} = \omega_{min} \quad (3.16)$$

$$\mu_{t+1} = X_{t+1} \quad (3.17)$$

$$\sigma_{t+1}^2 = \sigma_{max}^2, \quad (3.18)$$

kde ω_{min} znamená vybrání malé hodnoty váhy ω a σ_{max}^2 značí vybrání velké hodnoty σ^2 s ohledem na ostatní křivky v pixelu. V tento okamžik již je každý pixel přiřazen konkrétní křivce a zbývá určit, zda jednotlivé křivky a tím i pixely, které se s nimi shodují, patří do pozadí nebo do popředí. Gaussovy křivky jsou seřazeny sestupně podle poměru ω/σ a pomocí rovnice (3.19) je vybráno prvních B křivek pro tvorbu modelu pozadí. Práh T určuje, jak velká část seřazených křivek bude vybrána jako pozadí a zbylá část bude použita na tvorbu popředí.

$$B = \operatorname{argmin}_b \left(\sum_{i=1}^b \omega_i > T \right) \quad (3.19)$$



Obrázek 3.6: Zobrazení vykreslení tří Gaussových křivek, kde každá je vyznačena modrou barvou. Červená barva lemující všechny křivky signalizuje celkový součet hustot pravděpodobností jednotlivých křivek, které jsou ještě vynásobeny svými vahami.

Výhodou takového postupu je rozdílná velikost prahu každého pixelu. Je umožněno objektům stát se součástí pozadí bez jakékoliv újmy na existujícím pozadí. Hlavní nevýhodou

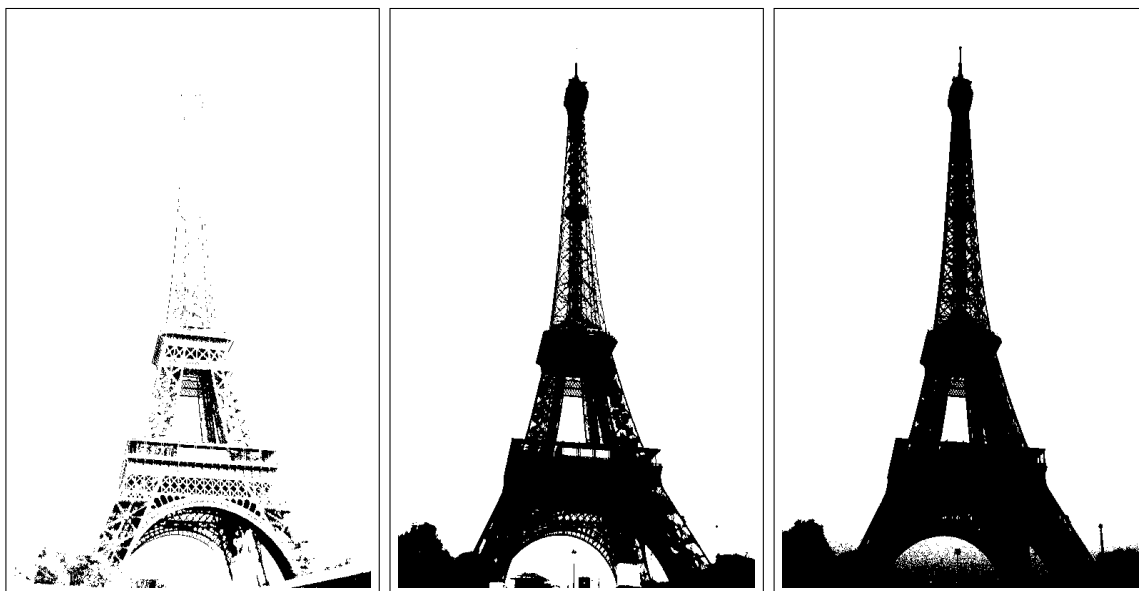
je neschopnost se kvalitně vypořádat s náhlými silnými změny světelných podmínek. Na obrázku 3.6 je ukázáno, jak může vypadat směs tří Gaussových křivek modelující jeden pixel v jednodimenzionálním prostoru. Tento algoritmus je používán a podrobněji popsán v Stauffer, Grimson [10].

3.3 Prahování

Jedná se o jednu z metod segmentace obrazu. Výsledkem prahování je binární obraz, který obsahuje pouze pixely 0 nebo 1. O tom, jakou z těchto hodnot bude pixel nabývat, se rozhoduje podle hodnoty prahu T .

$$f(x) = \begin{cases} 1 & \text{pro } x \geq T \\ 0 & \text{pro } x < T \end{cases} \quad (3.20)$$

Jestliže je stejný práh použit v celém obrázku, operace se nazývá globální prahování. Pokud se hodnota T liší pro různé části obrazu, pak se operace nazývá adaptivní prahování.

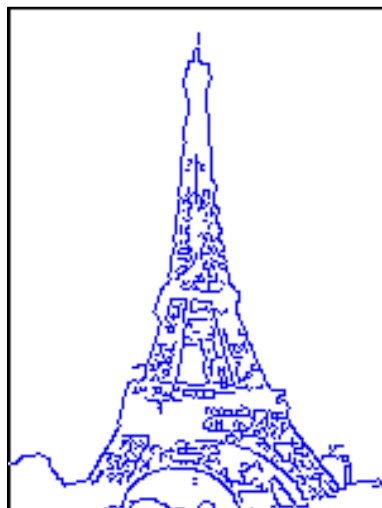


Obrázek 3.7: Vliv velikosti prahu na výsledné vykreslení obrázku. Jednotlivé obrázky jsou vytvořeny po aplikaci prahování s prahem o hodnotě 20, 50 a 100.

Na obrázku 3.7 je znázorněno, k jak velké změně může dojít vlivem při zvolení konkrétní hodnoty prahu.

3.4 Cannyho hranový detektor

Jedna z hlavních možností, jak rozlišit velikost objektu ve snímku, je pomocí detekce hran. Mezi nejpoužívanější postup z této kategorie popsal John Canny v roce 1986 [2]. V této práci mimo jiné stanovil, jaká kritéria by tento algoritmus měl splňovat. Patří sem snaha o co nejmenší chybnou detekci, dobrá lokalizace hran a zabránění v detekování stejné hrany vícekrát. Samotný algoritmus se nejčastěji skládá ze 4 kroků. Nejprve je pomocí Gaussova filtru odstraněn šum ve snímku. Velikost Gaussova jádra ovlivňuje výkon detektoru. Dále



Obrázek 3.8: Výsledek aplikace Cannyho hranového detektoru, který závisí na velikosti malého a velkého prahu. V tomto případě byla pro malý prah zvolena hodnota 20 a pro velký 255.

dojde k nalezení velikosti gradientu a jeho směru pomocí dvojice konvolučních matic (3.21) a (3.22). K tomuto účelu je použit podobný postup, na jehož principu funguje Sobelův operátor [8]. Matice G_x udává horizontální směr a matice G_y vertikální směr.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (3.21)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad (3.22)$$

Z matic je pomocí funkce (3.23) vypočítána velikost gradientu a pomocí (3.24) jeho směr.

$$G = \sqrt{G_x^2 + G_y^2} \quad (3.23)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (3.24)$$

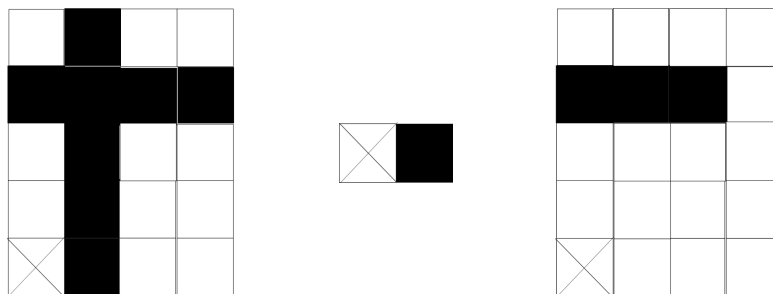
Výsledek je zaokrouhlený k jednomu ze čtyř možných úhlů (0° , 45° , 90° , 135°). Dále jsou pomocí techniky non-maximum suppression odstraněny pixely, které nejsou součástí detekovaných hran. V důsledku toho zůstanou pouze tenké hrany. V posledním kroku je provedeno prahování s hysterezí. K tomuto účelu jsou vybrány malý a velký práh. Pokud má gradient pixelu větší hodnotu než je hodnota většího prahu, pixel je brán jako součást hrany. Pokud je gradient pixelu menší než je hodnota menšího prahu, pak je pixel zamítnut. Pokud je gradient pixelu mezi hodnotami obou prahů, tento pixel bude akceptován, pokud je připojen k pixelu, který má hodnotu větší, než je hodnota většího prahu. Výsledek potom může vypadat podobně, jako je ukázáno na obrázku 3.8.

3.5 Morfologické operátory

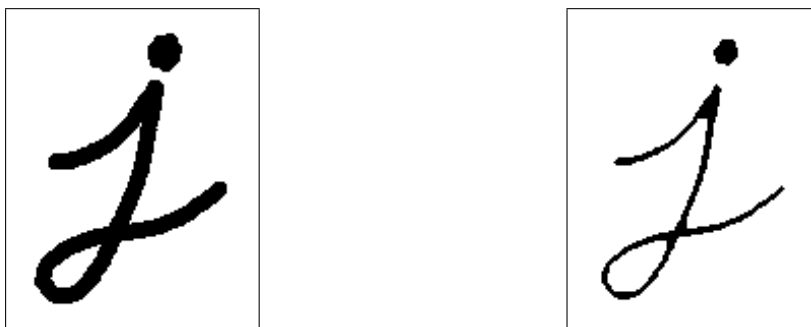
Algoritmy, které detekují pohyb, velmi často zobrazují šum nebo nepatrný pohyb, který nemusí být vždy žádoucí. Pro vyhlazení obrazu jsem se rozhodl použít morfologické operace, které mi umožní odstranit šum a zvýraznit podstatné oblasti zájmu. Základní prvky toho algoritmu jsou binární obraz a strukturální element. Postup spočívá v aplikaci elementu na každý pixel obrazu, čímž vznikne nový obraz. Element může mít různou velikost a tvar, např. čtverec, kruh, elipsa atd. Dalším prvkem, který ovlivňuje výsledek je počet iterací operace. Při větším počtu provedení operací dochází k lepším výsledkům, avšak také dojde ke zpomalení celkového běhu aplikace. Pokud je na obraz použito otevření resp. uzavření, další aplikace otevření resp. uzavření na obraz již nebude mít vliv. Důkladnější popis těchto operací je popsán v Haralick, Sternberg, Zhuang [4].

Dilatace

V případě dilatace je pixelu přiřazena nejvyšší hodnota intenzity jeho sousedů, aplikací strukturálního elementu, jak je ukázáno na obrázku 3.9. Výsledkem této operace dojde ke zmenšení oblastí v popředí jako je tomu na obrázku 3.10.



Obrázek 3.9: Princip použití strukturálního elementu, který slouží pro zaplnění děr v obraze. Úplně vlevo je počáteční obraz, uprostřed se nachází strukturální element a vpravo je výsledný obraz.

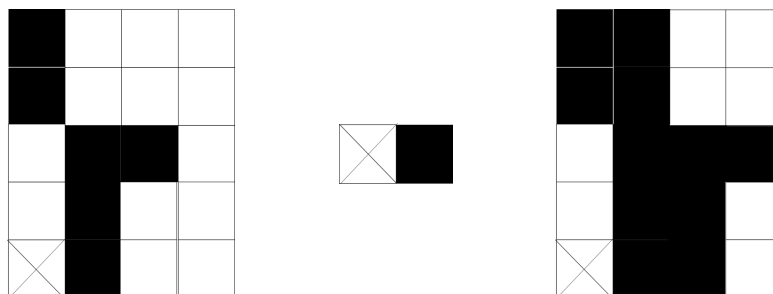


Obrázek 3.10: Ukázka jakým způsobem se změní obrázek po aplikaci dilatace.

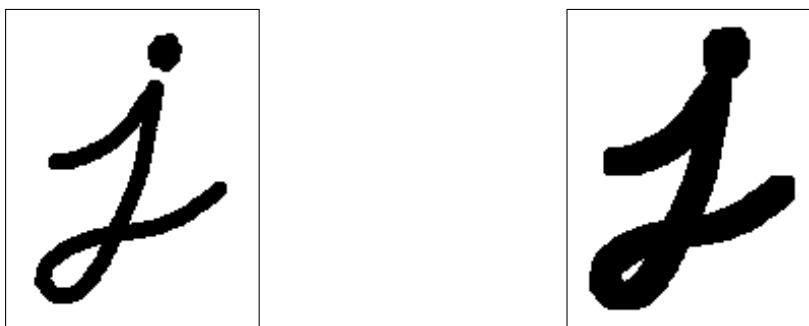
Eroze

Oproti dilataci se eroze liší pouze ohodnocením pixelů. Ze všech sousedů konkrétního pixelu je vybrána nejmenší hodnota, která je pixelu přiřazena. Na obrázku 3.11 je ukázáno, k jaké změně dojde po aplikaci strukturálního elementu. Světlejší oblasti jsou potlačeny a tmavší

zvětšeny. Používá se k odstranění šumu a zjednodušení složitých objektů. Z obrázku 3.12 je patrné, že došlo ke zvýraznění černých oblastí na úkor bílých.



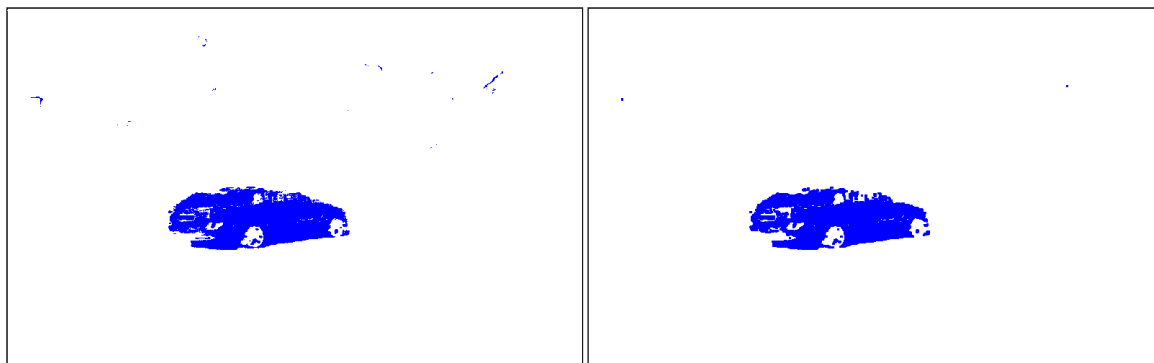
Obrázek 3.11: Ukázka transformace obrazu vlevo pomocí strukturního elementu na obraz vpravo.



Obrázek 3.12: Ukázání změny obrazu po aplikaci eroze, při které dochází ke zvýraznění obrysů objektu.

Otevření

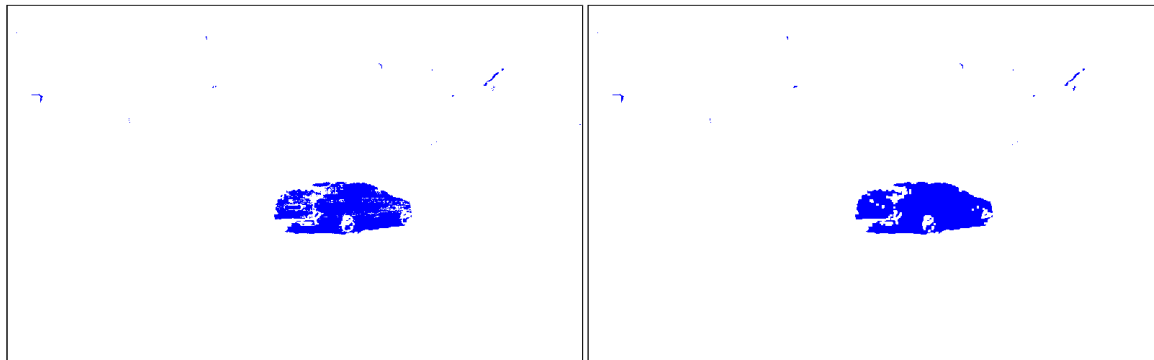
Tato metoda značí použití nejprve eroze, a potom dilatace na stejný obraz. Jak je možné vidět na obrázku 3.13, dojde k eliminaci malých objektů a zjednodušení struktur velkých objektů.



Obrázek 3.13: Ukázka aplikace operátoru otevření, kde z obrázku je patrné odstranění malých osamocených pixelů tvořící nepodstatný šum. Na druhou stranu dojde k i odstranění některých pixelů, které tvoří pohybující se automobil, čímž dojde ke zhoršení jeho struktury.

Uzavření

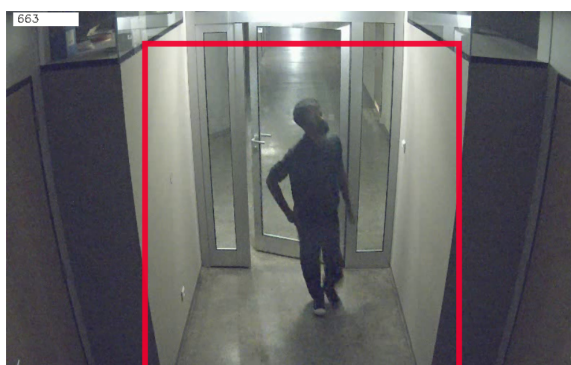
Uzavřením se nazývá postup použití nejprve dilatace a následně eroze. Dojde tak ke spojení objektů, které jsou blízko u sebe. Obrázek 3.14 ukazuje výsledek aplikace metody uzavření.



Obrázek 3.14: Ukázka praktického využití operace uzavření, kde dochází k vyplnění malých děr v pohybujícím se automobile a k celkově větší celistvosti.

3.6 Specifikace oblasti zájmu

Tato technika, která je známá pod zkratkou ROI³, slouží pro zúžení oblasti pro zpracování v obraze na menší a konkrétnější oblast. V počítačové grafice slouží pro odlišné zpracování různých oblastí na snímku. Často se využívá pro zpracování oblastí zájmu ve větší kvalitě, než jiné oblasti. ROI může být definováno prakticky jakýmkoliv geometrickým tvarem. Na obrázku 3.15 je ukázka specifikace ROI pomocí obdélníku. Oblast specifikována ROI je později zpracována a zbylé oblasti jsou zahozeny.



Obrázek 3.15: Využití tvorby oblasti zájmu, kde pouze tato oblast je dále analyzována. V tomto případě ROI vyplňuje oblast, ve které dochází k největší intenzitě pohybu. Díky tomu nedojde k detekci pohybu v levém a pravém dolnímu rohu, což může být využito v případě, kdy jsou v těchto oblastech otevřeny dveře a není nutné, aby algoritmus zachycoval jakýkoliv pohyb vyskytující se v levé nebo pravé místnosti. Samozřejmě je třeba zhodnotit, zda je lepší tento pohyb zachycovat či ignorovat.

³Region Of Interest

Kapitola 4

Návrh detekce pohybu v obraze

Hlavním cílem programu je správná detekce pohybujících se objektů v obraze z kamery. Aplikace je nicméně upravena spíše pro detekci pohybu z videa. Celkový záznam je procházen a v případě detekování pohybu, je vytvořeno cca 1 minutové video, sestávající se z 30 sekund před pohybem a 30 sekund za pohybem. Tento návrh je aplikovatelný na statické kamery, které nejsou ovlivněny otřesy. Pro pohybující se kameru existuje přístup publikovaný Culter a Davis [3].

4.1 Nalezení změny v obraze

Detekci obrazu jsem se rozhodl aplikovat pomocí algoritmu, který popsal Zoran Zivkovic ve svém článku [12]. Díky tomu ze snímku ze vstupního videa vznikne černobílý obrázek. Každý snímek je potom upravován a analyzován zvlášť.

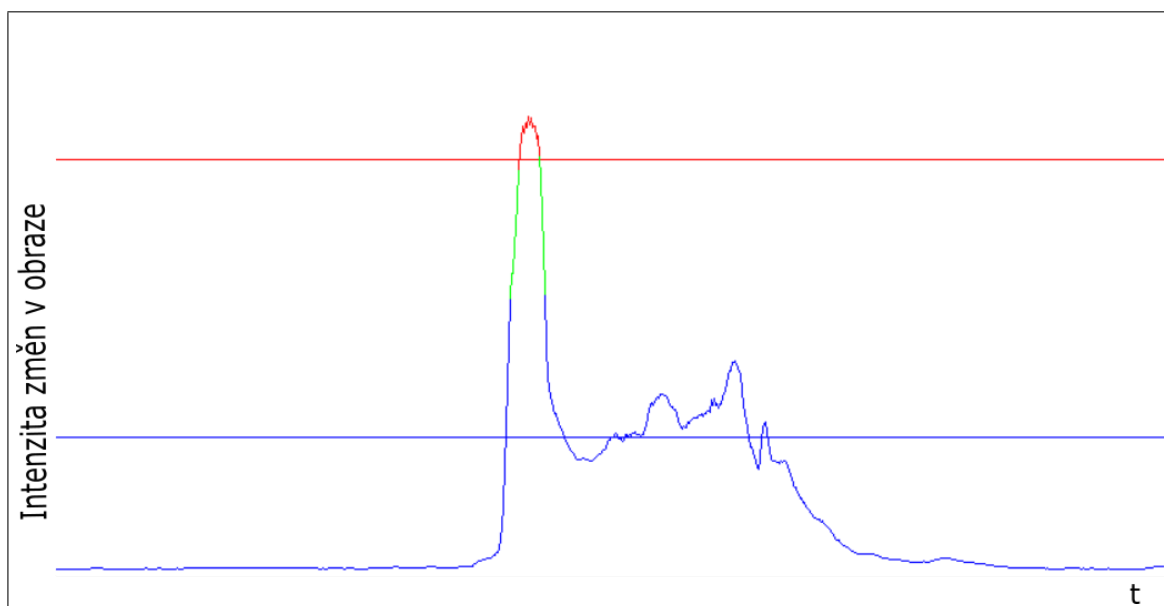
Pohyb je pomocí prahování značen pomocí bíle zbarvených pixelů. Algoritmus detekce po převodu snímku do černobílé zanechává v obraze spoustu malých bílých pixelů, které mohou mít vliv na úspěšnost detekce. Intenzita pohybu je totiž vykreslována pomocí grafu intenzit v čase, který bude popsán v další kapitole. Abych tedy nežádoucí šum odstranil, rozhodl jsem se použít morfologické operátory erozi a dilataci. Konkrétně použiji metodu otevření a to pouze s jednou iterací. Je pravda, že vícero použití obraz více vyhladí, ale dojde i ke značnému zpomalení aplikace, a proto jsem se od tohoto řešení upustil. Dále jsem velikost elementu stanovil na 3×3 počáteční pozici na (1,1).

Prvních 10 sekund je pohyb ignorován, protože je třeba nechat algoritmus správně natrénovat, co je ve videu bráno jako pozadí, aby dokázal lépe odlišit popředí s pohybem. K nalezení hran objektů mi posloužil Cannyho hranový detektor. Jako dolní práh jsem použil číslo 254,9, tedy číslo co nejbližší bílé barvě. Následně jsem jednotlivé hrany spojil do spojitých tvarů a určil velikost jejich plochy pomocí Suzuki, Abe [11]. Po 10 sekundách stabilizace jsou po dobu 30 sekund snímky videa ukládány do vektorů matic, aby byl vždy uchován aspoň půlminutový záznam před výskytem pohybu. Poté už jsou snímky prohledávány s cílem nalézt pohyb. Dokud pohyb není nalezen, dochází k odstranění prvního snímku z vektoru a přidání nového snímku na jeho konec. Samotné rozlišení závažnosti pohybu nejprve přeskočí veškeré kontury, které mají méně než 3 hrany. Každý obrys jsem potom ohodnotil na základě jeho plochy bílých pixelů od 625 do 1500 pixelů. Dále pro lepší odlišení pohybu, který je příliš rychlý a tedy je s velkou pravděpodobností nepodstatný jsem stanovil, že takto velká kontura se musí vyskytnout v alespoň dvou po sobě jdoucích snímcích. Pokud tato podmínka není splněna, není tento pohyb brán v úvahu. Tento postup

jsem aplikoval zejména kvůli prudké změně intenzity pixelů v obraze např. zhasnutí světel. Když je detekován pohyb, je následně ukládáno dalších 30 sekund. Pokud v tento okamžik dojde ve videu k pohybu, již není třeba zkoumat, protože bude i tak zaznamenán. Získaný vektor plný snímků je procházen a ze snímků je vytvořeno video ve formátu „.avi“. V tento okamžik končí fáze detekce pohybu a začíná fáze vizualizace výsledků.

4.2 Vizualizace výsledků

Z vytvořeného videa je nejprve vytvořen graf zobrazující počet bílých pixelů v průběhu videa. Graf, který je zobrazen na obrázku 4.1, sestává z dvojice přímek, které signalizují, k jak moc velkému počtu bílých pixelů došlo. Jednotlivé hodnoty intenzit jsou vyděleny číslem 400, aby byly lépe zobrazitelné v grafu.

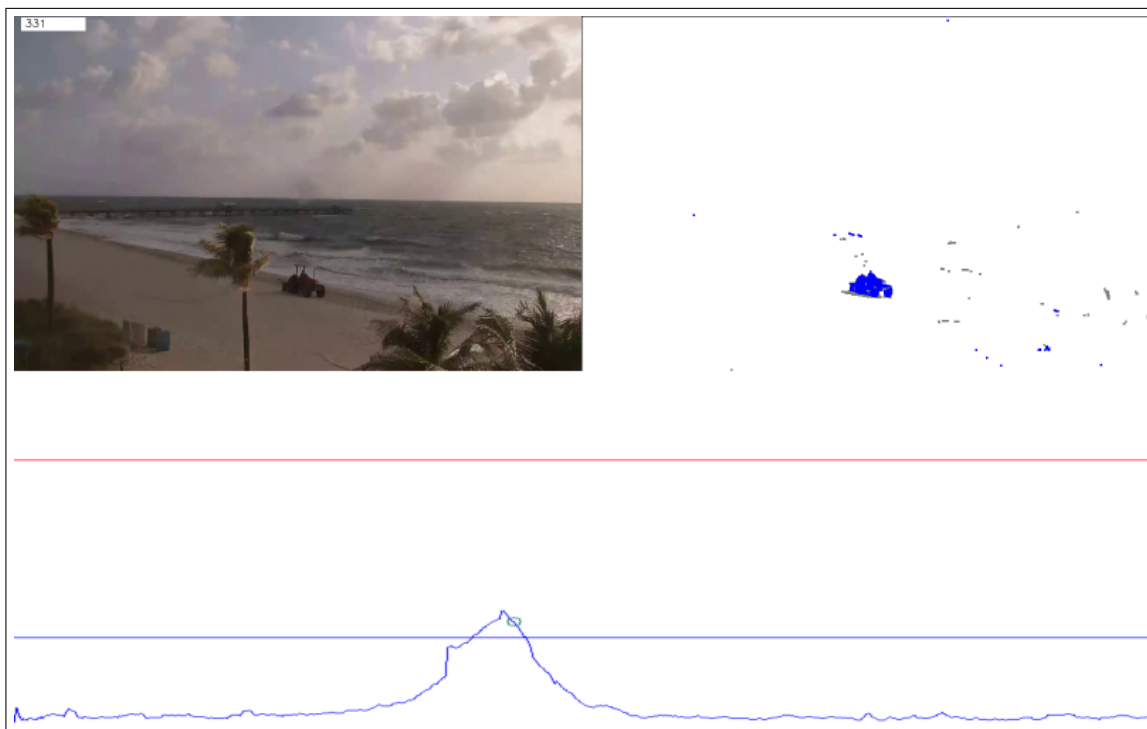


Obrázek 4.1: Graf zobrazuje intenzitu pohybu (bílých pixelů) v čase. Intenzita pohybu menší než modrá přímka vyznačuje obvyklý pohyb v závislosti na poloze kamery. Intenzita jdoucí nad červenou přímkou upozorňuje na neobvyklé chování, které může být způsobené zhasnutím či rozsvícením světla nebo pohybem v příliš velké blízkosti kamery.

Dále je video znovu procházeno, čímž dojde k vložení pohyblivého kolečka, které se při přehrávání videa pohybuje po grafu a ukazuje aktuální intenzitu bílých pixelů konkrétního snímku. Následně je vytvořeno nové video, jak je možné vidět na obrázku 4.2, sestávající se ze snímku videa, černobílého snímku a z grafu.

4.3 Testování

Program jsem testoval na různých situacích, které se běžně vyskytují, ale mohou způsobovat algoritmu značné potíže. V případě detekce v uzavřených prostorech to jsou např. odrazy na skleněných dveřích nebo pohyb v oblasti se slabým osvětlením. Vybral jsem některé z těchto zvláštností a pokusil se algoritmus upravit, aby tyto anomálie neovlivňovali úspěšnost detekce.



Obrázek 4.2: Zobrazení výsledného videa, díky kterému můžeme vidět v jeden okamžik originální záznam videa zachycující pohyb, záznam videa zpracovaný algoritmem a aktuální intenzitu pohybu pomocí grafu. Číslo v levém horním rohu navíc udává aktuální číslo snímku.

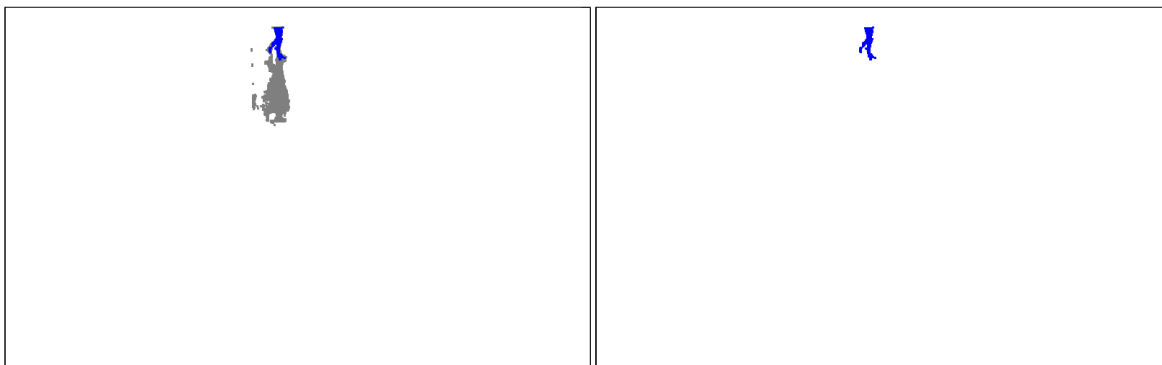
Odfiltrování stínů

Algoritmus jsem původně měl nastavený tak, aby detekoval kromě pohybujících se objektů také stíny, které se u nich vyskytují. Jelikož můj postup spočívá v rozlišení důležitého pohybu podle velikosti obsahu plochy bílých pixelů, které identifikují pohybující se objekt, nastal zde problém v odlišení šedých a bílých pixelů. Bílé pixely jsem lokalizoval pomocí funkce, která vrací počet nečerných pixelů v obraze, tedy počet bílých a šedých pixelů. Procházení obrazu a nahrazování všech šedých pixelů černými nemá smysl, poněvadž dochází k příliš velkému zpomalení aplikace. Místo toho lze pomocí metody `setDetectShadows()` nastavit, aby stíny nebyly zobrazovány. Jak je vidět na obrázku 4.3 stíny mohou ovlivňovat úspěšnost detekce pohybu. V tomto případě byla velikost stínu tak velká, že program považoval stín za důležitý pohyb.

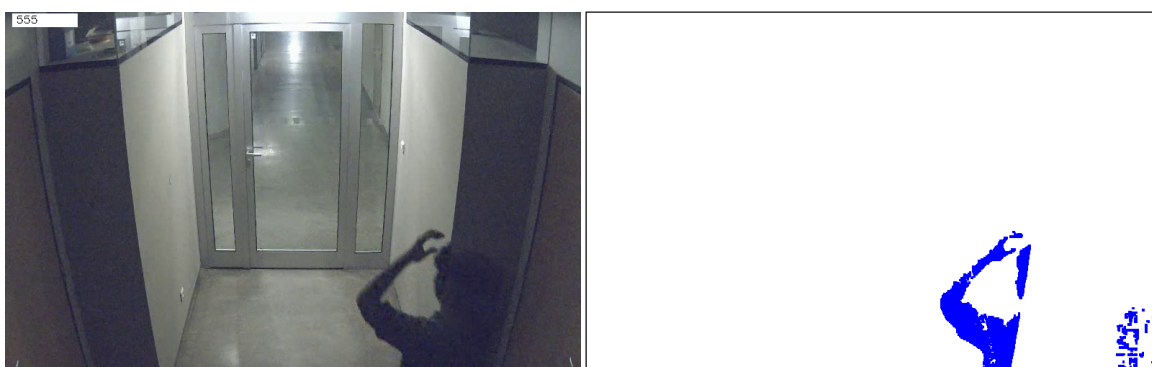
Tmavé objekty

Jak je možno vidět na obrázku 4.4, pohyb tmavých objektů ve tmavém prostředí velmi znesnadňuje detekci. V takovémto případě je objekt, který se ze začátku pohybuje pomalu, klasifikován jako součást pozadí. Díky schopnosti GMM¹ však dochází k aktualizaci hodnot pozadí v čase, a proto je po chvíli objekt správně klasifikován jako součást popředí. V této situaci není tato anomálie nijak významná, poněvadž člověk směřuje do oblasti s lepším osvětlením a ke správné detekci dojde. Pokud by však osvětlení nebylo takto intenzivní,

¹Gaussian Mixture Model – Směsice gaussových křivek



Obrázek 4.3: Ukázka, jaký vliv mohou mít stíny na výslednou detekci. První snímek zachycuje okamžik, kdy velikost stínu spustila detekci pohybu. Druhý snímek zobrazuje tentýž záběr bez zobrazení stínů a v tomto případě správně program nedetekoval pohyb.



Obrázek 4.4: Ilustrace nedokonalé detekce pohybu vlivem splnutí pozadí a popředí.

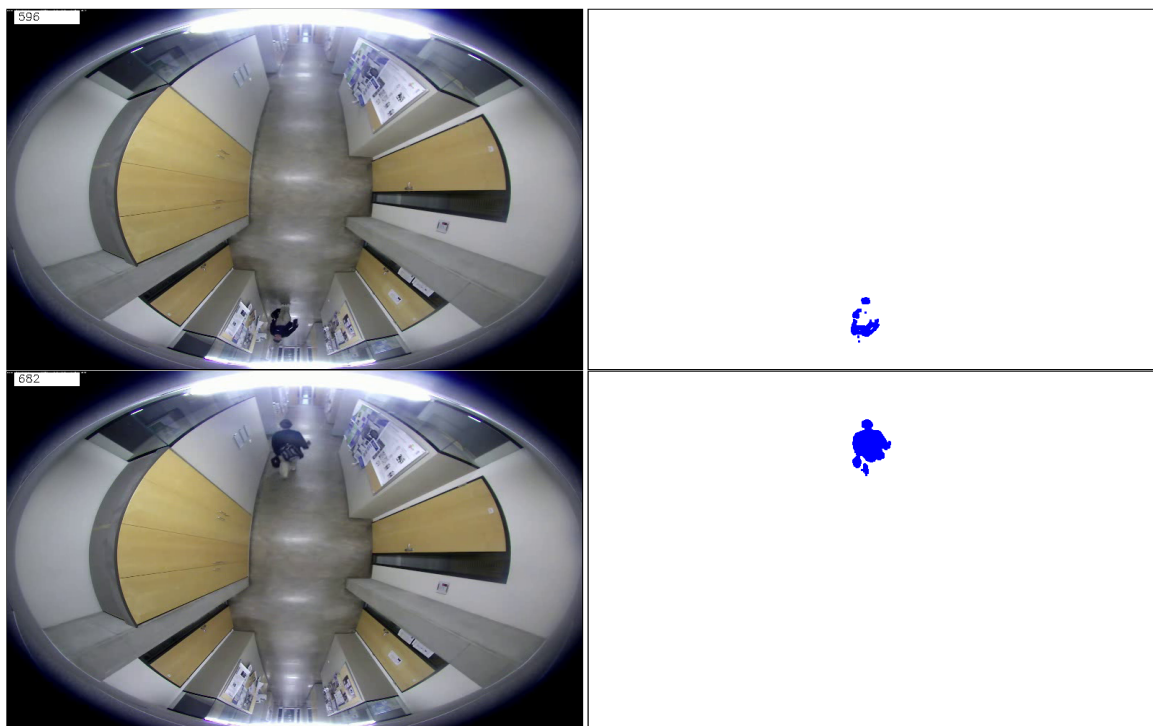
algoritmus by objekt nemusel vůbec detekovat.

Natáčení panoramatického videa

Tento typ kamery dokáže zachytit daleko větší prostor, ale nemusí to být vždy jistota lepší detekce pohybu. Na obrázku 4.5 je ukázáno, jak algoritmus může selhat.

Přínosy specifikace oblasti zájmu

Jak již bylo zmíněno, jedná se o stanovení oblasti v obraze, která má pro naše účely větší význam. Největší uplatnění ROI (Regions Of Interest) má podle mého názoru při sledování venkovního pohybu, kdy velmi často dochází k mylné detekci. Na obrázku 4.6 je vidět, že pohyb ve velké vzdálenosti od kamery může být obtížně detekován. Navíc je lehce zaměnitelný za šum listí ve větru nebo za mořské vlny. Obrázek 4.7 ukazuje použití ROI na takovýto případ. Pohyb moře a oblohy je zcela ignorován a detekce probíhá pouze na důležité oblasti. Jak je vidět, i přes tuto snahu se v obraze vyskytují malé oblasti, které mohou být zaměněny s pohybem člověka. V tomto případě je to způsobeno nízkou kvalitou záznamu z dané kamery a také otřesy kamery způsobené silným větrem. V případě zúžení oblasti zájmu o tuto konkrétní oblast může dojít k opomenutí pohybu v případě, že někdo do této oblasti vstoupí.



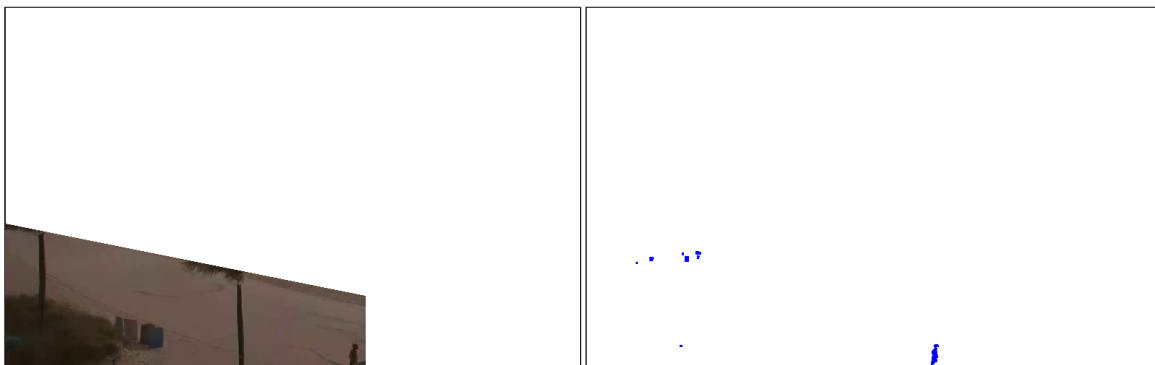
Obrázek 4.5: První dvojice obrázků zachycuje pohyb na snímku číslo 596. Přesně na tomto snímku došlo k první detekci pohybu. Na další dvojici snímků nebyl pohyb identifikován, ale ze snímků vyplývá, že pohyb byl vykreslen lépe než na předchozím snímku.



Obrázek 4.6: Ilustrace jak obtížné může být v určitých situacích detekování pohybu vlivem okolního prostředí nebo počasí. Dole uprostřed se vyskytuje malá postava, kterou by bylo dobré detekovat jako důležitý pohyb. Ve videu, ze kterého je tento snímek, se vyskytuje velmi silný vítr, vlivem něhož algoritmus zachycuje i listy palm a ty jsou poté chybně označeny jako pohyb. Stejný problém nastává i u vln na pláži.

Krátké změny

Velkým problémem algoritmu GMM jsou skokové změny podmínek. Nejčastěji jsou zapříčiněny změnou okolního světla, což lze nejlépe pozorovat v uzavřených prostorech jako je tomu na obrázku 4.8. Díky odfiltrování stínu ze snímku se odstranil problém změny podmínek v pozadí, které nejsou lidským okem na první pohled patrné, ale algoritmus je identifikoval



Obrázek 4.7: Odfiltrování moře a listí palm dojde ke značnému posílení úspěšnosti detekce pohybu. Tentokrát je možné lépe poznat výskyt osoby v dolní oblasti snímku. Jak je možné vidět vlevo dole, tímto přístupem není zajištěna stoprocentní úspěšnost.

jako pohyb. Při zhasnutí či rozsvícení světel přímo před kamerou, dojde k prudké změně v obraze, která obvykle není důležitá pro zaznamenání.

K částečnému potlačení tohoto vjemu jsem stanovil podmínku, že pohyb bude uznán dostatečným, pokud kromě určité úrovně intenzity, bude trvat i určitou dobu. Proto jsem určil, že minimální hladina intenzity, od které je pohyb detekován se musí vyskytovat alespoň na 2 snímcích jdoucích po sobě. Tento interval jsem stanovil s přihlédnutím na negativní výsledky při obyčejných podmínkách. Jsou situace, kdy i změna světelných podmínek není programem správně zachycena. K tomu dochází například za denního světla, čímž není pozadí takto temné a změna podmínek nemá takový efekt.



Obrázek 4.8: Znázornění reakce na zhasnutí světla. První snímek je 340. v pořadí, na kterém je vidět v pravém horním rohu rozsvíceno. Na dalším snímku s pořadovým číslem 346 je již světlo zhasnuté a tato změna se začíná algoritmem projevovat. Největší intenzita se vyskytuje okolo snímku 383 a zotavení se z této události dochází částečně až na snímku 426.

Kapitola 5

Implementace a vyhodnocení

5.1 Implementace

Aplikaci jsem vyvíjel v prostředí Visual Studio 2012. Hlavní kostru programu jsem vytvořil v programovacích jazycích C a C++. Kromě standardních knihoven jsem použil i open source knihovnu OpenCV 3.0. Z té jsem využil hlavně funkce pro práci s obrazem a videem.

Hlavní program sestává ze 3 hlavních funkcí. Jsou to `detectMotion()`, `createVideo()` a `createGraph()`. První z nich je funkce `detectMotion()`, která uskutečňuje celkové procházení videa a hledání pohybu. Využívá k tomu `createBackgroundSubtractorMOG2()`, která je implementovaná v OpenCV na základě článku Zivkoviče [12]. Funkce `BackgroundSubtractor::apply()` vytvoří ze vstupního snímku snímek s odděleným pozadím od popředí. V této funkci je již prahování na obrázek, nicméně na základě testování vstupních videí jsem použil prahování ještě samostatně, protože napomáhá s odstraněním vzniklého šumu. Použil jsem proto vestavěnou funkci `threshold()`. Morfologické operátory mají taktéž v OpenCV předem vytvořené funkce. Pro ustanovení strukturního elementu jsem použil funkci `getStructuringElement()`, která jako parametry má typ elementu, jeho velikost a výchozí pozici. Jak pro erozi tak dilataci jsem použil stejně nastavené elementy. Konkrétní operace jsou volány funkcemi `erode()` a `dilate()`. Cannyho detektor hran je implementován funkcí `Canny()`. Nejdůležitější parametry funkce jsou minimální a maximální práh. Já jsem minimum zvolil 254,9 a maximum na 255. Pro identifikaci pohybujících se objektů jsem použil funkci `findContours()`. Její hlavní přínos pro moji práci spočívá v získání souřadnic všech spojených objektů (blobů) a jejich uložení do vektoru souřadnic. Každý takto získaný blob je posuzován podle svého obsahu, který získám funkcí `contourArea()`. Ve funkci `createVideo()` dochází k vytvoření analyzujícího videa spolu s grafem intenzity pohybu. Zde dochází k volání standardních funkcí, až na `cvtColor()`, která slouží pro převedení vstupního obrazu, grafu a obrazu popředí na stejný kanál. Ve funkci `createGraph()` je použita důležitá funkce `countNonZero()`, která vrací počet nenulových pixelů, které identifikují objekty v popředí a podle toho sestavuji graf zobrazující jejich intenzitu v čase. Pro načtení videa používám funkci `VideoCapture()` a pro zápis do video formátu `VideoWriter()`. Podrobnější popis včetně příkladů použití těchto funkcí lze nalézt knize Bradski, Kaehler [1].

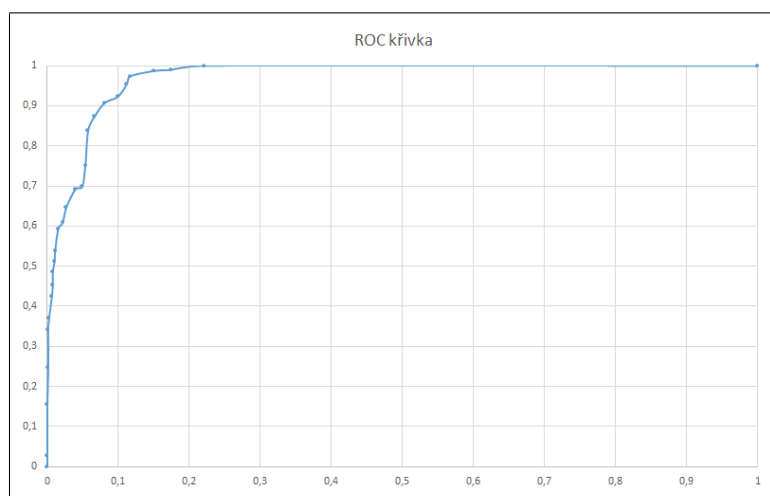
5.2 Spuštění aplikace

Program je vytvořen, aby jej bylo možné spustit pod operačním systémem Windows. Aplikace se spouští přes příkazový řádek ve formátu:

```
BackgroundDetection.exe { cesta_k_souboru } { cesta_kam_se_má_soubor_uložit }
```

Pokud chce uživatel zpracovat video soubor, musí jako první parametr zadat cestu k souboru, který chce analyzovat a jako druhý parametr cestu ke složce, kam chce vytvořený video soubor uložit.

5.3 Vyhodnocení výsledků



Obrázek 5.1: ROC křivka znázorňující vliv prahu na správnost detekce.

Z testů, které jsem provedl, je patrné, že správnou detekci ovlivňuje několik parametrů. Především analyzování pohybu ve venkovním prostředí je velmi obtížné. V uzavřené oblasti ovlivňuje správnou detekci hlavně změna světelných podmínek, která ve venkovní scéně nemá až takový vliv, protože je možná pouze změnami počasí, které obvykle nedosahují takové rychlosti a intenzity, aby měly vliv na detekci. Venkovní prostředí ovšem značně ovlivňují jiné aspekty počasí, jako je vítr nebo déšť. Z provedených testů jsem došel k závěru, že v takových to případech je vhodné použití ROI pro vytyčení oblastí, které nás zajímají z hlediska detekce pohybu. Velký rozdíl v detekci pohybu tento přístup přináší u kamer sledující pohyb u moře nebo u jakékoliv oblasti s přítomností vodní hladiny. Ta vytváří konstantní změnu na pozadí a jejím odfiltrováním může detektor zachycovat objekty s menší plochou. Zvláště v případech, kdy kamera je velmi vzdálená od pohybujících se objektů a například pohyb listí může být mylně identifikován jako pohyb člověka. Je velmi důležité správně zvolit jak velká plocha bílých pixelů bude značit důležitý pohyb. V případě vzdálené kamery se velikost plochy objektu pohybuje mezi 70 až 200 pixely, při kamere ve velké blízkosti je to od plochy 500 pixelů.

Jak už bylo zmíněno, v uzavřených místnostech je detekce pohybu nezávislá na jevech počasí. Rozpoznání objektu ovšem ovlivňují jiné jevy. Může to být slabé osvětlení, kdy tmavé objekty jsou hůře rozpoznatelné a splývají s pozadím. Dále také pohyb za sklem či

jiným průhledným předmětem se stává špatně rozpoznatelný. Z výsledků testování v uzavřených prostorech, jsem z jednoho videa vytvořil ROC křivku 5.1, která zobrazuje úspěšnost detekce v závislosti na minimální ploše pohybujícího se objektu.

Tento program není příliš vhodný pro nasazení v oblastech s vysokou intenzitou pohybu. V tomto případě by vznikala velká spousta videí s pohybem, přičemž by bylo vhodnější nahrávání pohybu bez přerušení. Vhodné je však použití v méně rušných hodinách, kdy není taková intenzita pohybu a konkrétní objekt je lépe rozpoznán.

Kapitola 6

Závěr

Vypracováním této práce jsem vytvořil aplikaci, která detekuje pohybující se objekt ve videozáznamu a vytvoří krátké video, které tento pohyb zobrazuje spolu s grafem intenzity pohybu v čase. Hlavním cílem bylo vhodně uživatele upozornit na pohyb vyskytující se ve videu a poukázat na jeho chování vzhledem k situacím, které mohou ovlivnit správnost detekce.

Práce je přizpůsobena hlavně na detekci pohybu v uzavřených místnostech. Pro venkovní prostory je dobré zakomponovat vytyčení oblasti zájmu pro ignorování šumu vzniklého např. vlivem počasí. Pro tento účel by bylo vhodné program rozšířit, aby dokázal rozpoznat oblasti snímku, kde se periodicky vyskytuje pohyb, který není důležitý pro zaznamenání a automaticky tyto oblasti ignorovat při analýze dalších snímků. V současné době je možné oblast zájmu vložit a uzpůsobit pouze manuálně v programu. Z této práce by se dalo vycházet v případě snahy o rozlišení typu pohybu. Vhodné rozlišení člověka od zvířete nebo automobilu by se dalo specifikovat jako interval velikostí plochy pixelů pro jednotlivé kategorie.

Při testování jsem používal záznamy z uzavřených prostor vytvořených jak za denního světla tak i za nočního osvětlení. Datovou sadu by bylo vhodné rozšířit o záznamy z venkovních prostorů.

Literatura

- [1] Bradski, G. R.; Kaehler, A.: *Learning OpenCV*. O'Reilly, 2008, ISBN 978-0-596-51613-0.
- [2] Canny, J.: *A Computational Approach to Edge Detection*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986: s. 679 – 698.
- [3] Cutler, R.; Davis, L. S.: *Robust real-time periodic motion detection, analysis, and applications*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000: s. 781 – 796.
- [4] Haralick, R. M.; Sternberg, S. R.; Zhuang, X.: *Image Analysis Using Mathematical Morphology*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987: s. 532 – 550.
- [5] Lee, D.-S.: *Effective Gaussian mixture learning for video background subtraction*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005: s. 827 – 832.
- [6] Lipton, A. J.; Fujiyoshi, H.; Patil, R. S.: *Moving target classification and tracking from real-time video*. *Proceedings Fourth IEEE Workshop on Applications of Computer Vision (WACV)*, 1998: s. 8 – 14.
- [7] McFarlane, N.; Schofield, C.: *Segmentation and tracking of piglets in images*. *Machine Vision and Applications*, 1995: s. 187–193.
- [8] Šonka, M.; Hlaváč, V.; Boyle, R.: *Image Processing, Understanding, and Machine Vision*. Thomson Learning, 2008, ISBN 978-0-495-08252-1.
- [9] Piccardi, M.: *Background subtraction techniques: a review*. *IEEE International Conference on Systems, Man, and Cybernetics*, 2004: s. 3099 – 3104.
- [10] Stauffer, C.; Grimson, W. E. L.: *Adaptive background mixture models for real-time tracking*. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999: s. 246–252.
- [11] Suzuki, S.; Abe, K.: *Topological Structural Analysis of Digitized Binary Images by Border Following*. *Computer Vision, Graphics, and Image Processing*, 1985: s. 32–46.
- [12] Zivkovic, Z.: *Improved adaptive Gaussian mixture model for background subtraction*. *International Conference on Pattern Recognition (ICPR)*, 2004: s. 28 – 31.

Příloha A

Obsah CD

- src - zdrojové kódy programu
- data - datové sady pro testování
- text - pdf textové zprávy a zdrojové kódy zprávy
- video
- program - spustitelné soubory
- README - návod pro spuštění aplikace
- plakát