VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

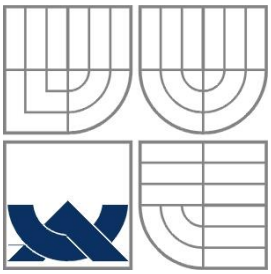
Fakulta informačních technologií

Faculty of Information Technology

BAKALÁŘSKÁ PRÁCE

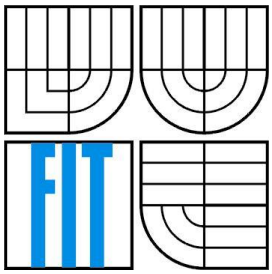
BACHELOR'S THESIS

Brno, 2016 Petr Půček



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 — PETR PŮČEK

AUTHOR

VEDOUCÍ PRÁCE — prof. Ing. ADAM HEROUT, Ph.D.

SUPERVISOR

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Půček 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. Pořiďte a anotujte vhodnou datovou sadu s různými druhy a intenzitami pohybu z více kamer.
4. Vyhledejte a vyviňte 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 srovnejte ji s alternativami.
7. Zhodnoťte dosažené výsledky a navrhněte 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, Božetěchova 2

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá rozpoznáním pohybu v obraze pořízeného stacionární kamerou. Popisuje řešení návrhu a implementace aplikace pro rozpoznávání na platformě Windows s využitím knihovny pro zpracování obrazu OpenCV. Popsané obecné principy analýzy a zpracování obrazu jsou však uplatnitelné rovněž u jiných systémů.

Abstract

This bachelor thesis deals with motion detection in video recorded on stationary camera. It describes the process of solution and implementation of motion detection app on Windows platform using OpenCV library for image processing. These principles of image analysis and processing can be used in any other systems.

Klíčová slova

Rozpoznání pohybu, detekce pohybu, odečítání pozadí, ViBe, OpenCV

Keywords

Motion detection, background subtraction, Visual background extractor, ViBe, OpenCV

Citace

Půček Petr: Detekce pohybu v obraze z kamery, bakalářská práce, Brno, FIT VUT v Brně, 2016

Detekce pohybu v obraze z kamery

Prohlášení

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

. . . . . .
Petr Půček
16. května 2016

OBSAH

1 ÚVOD

Jedna z problematik počítačového vidění je analýza pohybu, která má několik úrovní využití. První úroveň je pouhá detekce pohybu, která slouží především pro zpracování obrazu z bezpečnostních kamer, určených pro hlídání skladišť, obchodů či soukromých pozemků. Druhou úrovní je analýza 2-D pohybu, jejíž cílem je zjistit umístění objektů, směr pohybu a rychlost, kterou se pohybují. Nejčastěji se používají záznamy ze stacionárních kamer, které sledují pohybující se objekty. Třetí úrovní použití je analýza 3-D pohybu, která pracuje na stejném principu jako analýza 2-D pohybu, s tím rozdílem, že pracuje se třemi dimenzemi, tzn. že pohyb je analyzován v prostoru (u 2-D analýzy je pohyb analyzován pouze v rovině).

Práce obsahuje stručný přehled metod používaných k detekci pohybu v obraze a jejich srovnání. Podrobněji rozebírá metodu odečítání pozadí, která je použita i v praktické části. Tato metoda byla dále upravena a rozšířena, aby byla schopna rozeznat pohyb způsobený lidským faktorem a ignorovat pohyb způsobený špatnou kvalitou nahrávky nebo rychlou změnou světelných podmínek. Pro tuto část byla vytvořena sada videí a navržen jednoduchý program pro zpracování a vyhodnocení pohybu. Algoritmus pro odečítání pozadí byl dále použit k implementaci bezpečnostní demonstrační aplikace určené k monitorování interiérových prostor.

2 DETEKCE POHYBU

Detekce pohybu v dynamickém obraze je proces rozpoznání změny pozice objektů, které narušují statickou scénu. Většinou se jedná o počítačový agoritmus, který s pomocí kamery monitoruje určité prostředí a upozorňuje uživatelem obsluhovaný systém, jakmile rozpozná pohyb. Složitější systémy jsou navíc schopné rozeznat typ pohybu a na základě toho se rozhodnout, zda vyslat uživateli upozornění nebo ne.

Existují tři konvenční přístupy pro detekci pohybu [1]: rozdílová metoda, optický tok a metoda odečítání pozadí. Rozdílová metoda je velmi adaptivní k dynamickým prostředím, ale obecně vzato je slabá při extrakci všech relevantních pixelů. Optický tok může být použit k detekci nezávisle se pohybujících objektů spolu s pohybující se kamerou. Výpočet optického toku však může být výpočetně náročný a nelze ho aplikovat na videa v reálném čase bez použití patřičného vybavení [1]. Metoda odečítání pozadí poskytuje nejlepší výsledky, ale je velmi citlivá na dynamické změny způsobené změnou světla a okolními vlivy.

2.1 Problémy detekce pohybu

Detekce pohybu ve videu je elementární součástí bezpečnostních systémů určených k monitorování prostředí v reálném čase. Avšak ve venkovním prostředí, kde může nastat libovolná změna světla nebo nedůležitý (ale rušivý) pohyb v pozadí, to může být problém. Tyto rušivé elementy mohou způsobit zkreslený výstup systému, což může vést k nesprávnému nebo žádnému rozeznání pohybu.

Obr. 1: Snímek záznamu kamery monitorující silnici v lese. Příkladem špatného výstupu systému může být pohyb v lese, pohyb listů, změna světla atd.

Mějme záznam z kamery monitorující cestu v lese (Obr. 1). Předpokládejme, že automobil je v pohybu. V takovém případě by systém zaslal upozornění, že byl rozpoznán pohyb. Jakmile automobil opustí scénu, žádný pohyb nebude detekován. Toto platí pouze za velmi specifických, v některých případech i nemožných podmínek: obloha se nesmí změnit, počasí musí zůstat neměnné (nesmí fouknout vítr, nesmí začít pršet, sněžit, atd.), nesmí nastat žadný pohyb v pozadí (např. zvěř v lese). Jakmile je jedna z těchto podmínek porušena, systém chybně detekuje pohyb. Tomuto by se dalo předejít, pokud by systém detekoval pohyb na vozovce. Jedná se pouze o řešení konkrétního problému, které se nedá všeobecně aplikovat na jiná videa. Toto řešení je navíc použitelné jen v případě zatažené oblohy. Jakmile by vyšlo slunce a fouknul vítr, stíny na silnici by byly systémem detekovány jako pohyb.

Tyto komplikace mohou také nastat v interiéru a to především v noci, kdy jakákoliv změna světla může mít za následek nesprávný výstup systému. Pokud by někdo rozsvítil, byl by detekován pohyb. S největší pravděpodobností by se na dalších snímcích objevila osoba, tudíž by detekce rozsvícení nemusela být problém. Tento světelný jev ovšem může být způsoben i reflektory projíždějícího automobilu kolem domu a to už znamená nesprávný výstup systému.

Na základě nedávných výzkumů vzniklo několik metod, které se touto problematikou zabývají. V této práci je nastíněna a implementována vlastní metoda, která ignoruje rychlou změnu světelných podmínek v interiéru.

2.2 Odečítání pozadí

Statická kamera snímající scénu je typický případ bezpečnostního systému. Základním kamenem analýzy této scény je detekce objektů, které ji narušují. Snímky scény bez narušujících objektů mají nějaké obvyklé chování, které může být velmi dobře popsáno statistickým modelem [1]. Pokud známe statistický model scény, můžeme tyto objekty detekovat nalezením částí snímku, které neodpovídají modelu. Tento proces se nazývá odečítání pozadí.

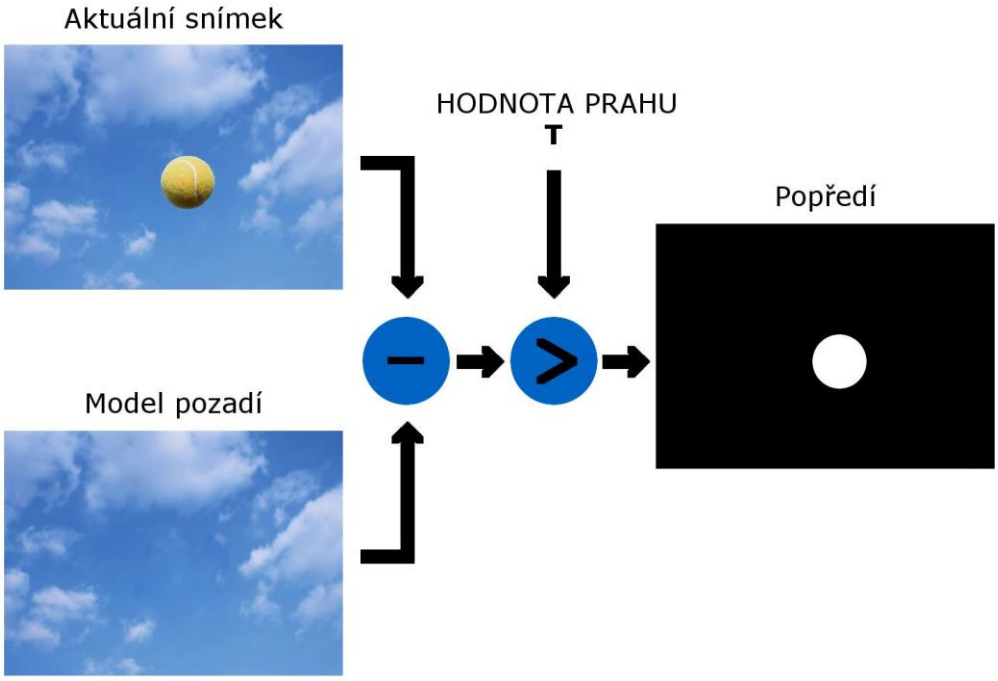
Metoda odečítání pozadí by měla být:

- Univerzální – měla by být schopna zpracovat jakékoliv video
- Jednoduchá – jednoduché techniky se lépe implementují
- Rychlá – měla by spotřebovat co nejméně výpočetního výkonu
- Přesná – měla by přesně detekovat tvary objektů
- Okamžitě použitelná – vyžadující pouze rychlou (nebo žádnou) inicializaci
- Adaptivní k dynamickým změnám scény

Všechny metody odečítání pozadí spojuje základní myšlenka, která říká, že máme statické pozadí a pohybující se objekty v popředí. Za předpokladu, že pohybující se objekt má v čase t barvu (nebo rozložení barev) lišící se od pozadí, může být tento princip popsán následujícím vzorcem:

$$F_t(s) = \begin{cases} 1 \text{ když } d\left(I_{s,t}, B_s\right) > \tau, \\ 0 \text{ jinak.} \end{cases}$$

Kde $F_t(s)$ označuje popředí pixelu s v čase t, $d(I_{s,t}, B_s)$ je vzdálenost mezi obrázkem pozadí B pixelu s a aktuálním obrázkem I pixelu s v čase t, τ je hodnota prahu. Modelování pozadí B a použití vhodné vzdálenostní metriky d jsou největší rozdíly mezi metodami odečítání pozadí.



Obr. 2: Princip modelově orientovaných metod odečítání pozadí, aktuální snímek je odečten od modelu pozadí a pixely, u kterých je výsledek tohoto rozdílu větší než hodnota prahu, jsou označeny za popředí

První přístup k detekci objektů metodou odečítání pozadí je tzv. naivní přístup. Jednoduchý model pozadí předpokládá, že hodnoty pixelů pozadí jsou s probíhajícím časem neměnné. Popředí a pozadí je tedy rozpoznáno porovnáním těchto hodnot aktuálního snímku s modelem pozadí. Rozdíly v těchto hodnotách mohou být způsobeny světelnými změnami ve scéně, šumem v obraze nebo pohybem objektů. Šum a osvětlení může být odstraněno použitím filtrů nebo detektoru zaměřujícího se na světelné změny. Odstranění pohybu v pozadí však může být velmi náročné a to hlavně pokud bereme v potaz předchozí požadavky.

Rozdíl mezi dvěma snímky je v praxi nepoužitelný, protože metody založené na této technice nejsou schopné se vypořádat se šumem nebo změnou světla. Proto byly navrženy složitější techniky. Tyto metody, pojmenované modelově orientované (model-based), vytváří model B_s pro hodnotu I_s každého pixelu. Model je použit k porovnávání předchozí a aktuální hodnoty konkrétního pixelu. Například W^4 algoritmus [2] vytváří pozadí na základě maximálních a minimálních hodnot jednotlivých pixelů a maximálního rozdílu mezi dvěma po sobě následujícími snímky. Pokročilejší modelově orientované techniky obsahují tyto body:

- Inicializace: Cílem je vytvořit validní odhad parametrů modelu již od prvních snímků videa.
- Porovnání: Hodnota aktuálního pixelu je porovnána s hodnotou modelového pixelu.
- Aktualizace modelu: Pokud hodnota pixelu odpovídá modelu, parametry a tedy i celý model je aktualizován.

V závislosti na typu modelu, technice porovnávání nebo způsobu aktualizace bylo navrženo několik kategorií modelově orientovaných metod odečítání pozadí.

V první kategorii těchto metod si model pozadí „předpoví“ hodnotu pixelu na základě nějakých statistických měření, například průměr hodnot všech pixelů. V nejjednodušším případě je model pozadí pro konkrétní pixel vypočítán jako aritmetický průměr předchozích hodnot tohoto pixelu. Pokud $B_{s,t}$ označuje hodnotu pixelu modelu na pozici s v čase t a $B_{s,t+1}$ je hodnota toho samého pixelu v čase $t+1$, vzorec pro aktualizaci modelu bude vypadat následovně:

$$B_{s,t+1} = \alpha I_{s,t} + (1-\alpha) B_{s,t}$$

kde α je aktualizační konstanta, která nabývá hodnot 0 až 1. V extrémních případech pro $\alpha = 0$ se model pozadí $B_{s,t}$ nebude aktualizovat, v opačném případě pro $\alpha = 1$ model pozadí nebude brán v potaz. Snímek složený z hodnot pixelů modelu $B_{s,t}$ představuje odhadované pozadí. Pixely spadající do popředí mohou být rozpoznány různými metrikami, například:

$$d_0 = \left|I_{s,t} - B_{s,t}\right|,$$

$$d_1 = \left|I^R_{s,t} - B^R_{s,t}\right| + \left|I^G_{s,t} - B^G_{s,t}\right| + \left|I^B_{s,t} - B^B_{s,t}\right|,$$

$$d_2 = \left(I^R_{s,t} - B^R_{s,t}\right)^2 + \left(I^G_{s,t} - B^G_{s,t}\right)^2 + \left(I^B_{s,t} - B^B_{s,t}\right)^2,$$

$$d_\infty = max\left\{\left|I^R_{s,t} - B^R_{s,t}\right|, \left|I^G_{s,t} - B^G_{s,t}\right|, \left|I^B_{s,t} - B^B_{s,t}\right|\right\},$$

kde *R*, *G*, *B* jsou jednotlivé složky barevného modelu RGB (Red, Green, Blue), d_0 je speciální metrika pouze pro obrázky v odstínech šedi.

V některých případech však tento přístup poskytuje málo obsáhlý model pozadí. Proto druhá kategorie modelově orientovaných metod odečítání pozadí má hodnoty pixelů popsané pomocí hustoty pravděpodobnosti. Model pozadí například předpokládá, že na hodnoty pixelů je aplikována Gaussova distribuce [3]. Ačkoliv je model rozsáhlejší, stále si není schopný poradit s pohyby na pozadí v dynamickém prostředí, jako je pohyb listů, problikávání monitorů, světelné odlesky atd.

Tento problém lze vyřešit použitím směsí hustot gaussovského rozložení při modelování pozadí [4]. Tento Gaussian Mixture Model (GMM) se stal velmi populární v komunitě počítačového vidění. GMM má však několik podstatných nedostatků. Předpokládá, že pozadí je častěji viditelné než popředí, což nelze aplikovat na každé video. Navíc, pokud je v pozadí rychlý nebo naopak velmi pomalý pohyb, citlivost

modelu může být špatně nastavena, což může způsobit, že daný objekt nebude detekován [5]. Nastavení validních parametrů modelu je tedy obtížné (obzvláště v prostředích, ve kterých vzniká šum).

Z těchto problémů vychází třetí kategorie modelově orientovaných metod odečítání pozadí, které se nazývají neparametrické. Vytvořené modely jsou velmi podobné modelům druhé kategorie, ale nesnaží se nastavit parametry na základě hustot pravděpodobností. Využívají jádrové odhady hustoty (Kernel Density Estimation), které počítají přímo z reálných hodnot pixelů. Díky tomuto jsou schopny rychle reagovat na události v pozadí. Tato metoda však vyžaduje velkou paměť, aby byla schopna uchovat všechny historické informace, což ji dělá téměř nepoužitelnou v praxi.

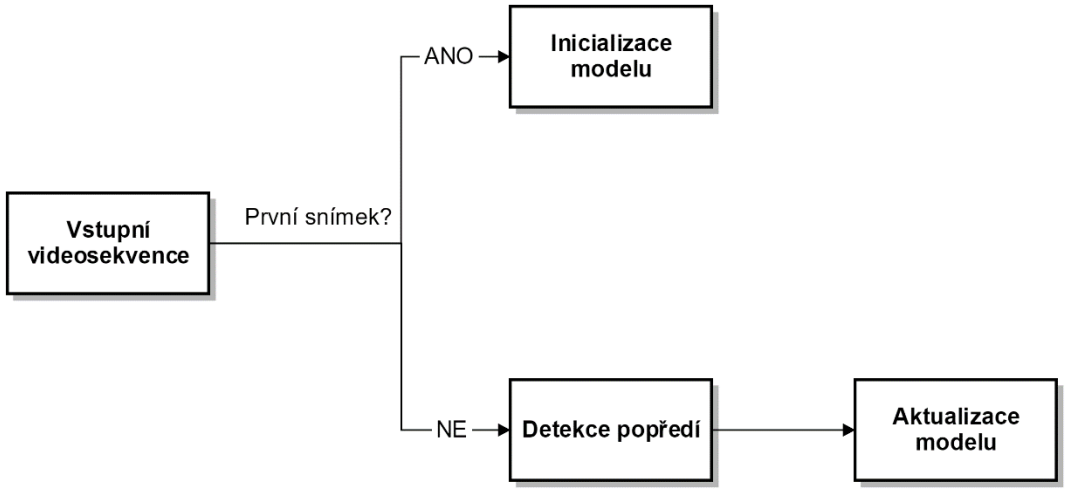
Další kategorií modelově orientovaných metod odečítání pozadí je modelování pomocí histogramu. Hlavním principem této metody je rozdělení obrázku na menší obrazové části, ze kterých jsou vytvořeny 1D nebo 3D histogramy. 1D histogramy jsou vytvořeny v případě obrázku v odstínech šedi, 3D histogramy jsou vytvořeny v případě barevného obrázku. Tento postup je aplikován na každý snímek videa pro všechny obrazové části. Poté lze jednoduše porovnat dva snímky a zjistit, zda došlo ke změně v histogramech a tudíž i k pohybu.

3 Visual Background Extractor: Metoda odečítání pozadí

Inteligentní bezpečnostní videosystémy je nově vznikající technologie zpracování obrazu a v posledních letech se velmi rychle vyvíjí. Vzniká široká škála nových aplikací, které tuto technologii využívají, například automatické monitorování dopravy, detekce osob, analýza jejich pohybu atd. Detekce pohybu znamená extrakci pohybujících se objektů, které narušují statickou scénu. Kvalitní extrakce je stěžejní krok pro inteligentní bezpečnostní videosystémy a má značný vliv při následujících operací jako je klasifikace objektů a analýza jejich pohybu.

Dosud nejpoužívanější metodou při detekci objektů je metoda odečítání pozadí [6]. Barnich a kolektiv [7] navrhli algoritmus ViBe, který efektivně sestaví model pozadí a účinně detekuje pohyb. Má velmi dobrý výkon a je schopný detekovat pohyb v reálném čase.

ViBe je vzorkově založený algoritmus, tzn. že náhodně vybírá vzorky na základě kterých sestavuje model pozadí. Hlavní části algoritmu ViBe jsou inicializace modelu, detekce popředí a aktualizace modelu. Pouze první snímek videosekvence je použit k sestavení modelu pozadí. Tento snímek je pak použit k detekci popředí a aktualizaci modelu. Blokový diagram detekce popředí algoritmu je popsán na obrázku 3.



Obr. 3: Blokový diagram detekce popředí algoritmu ViBe

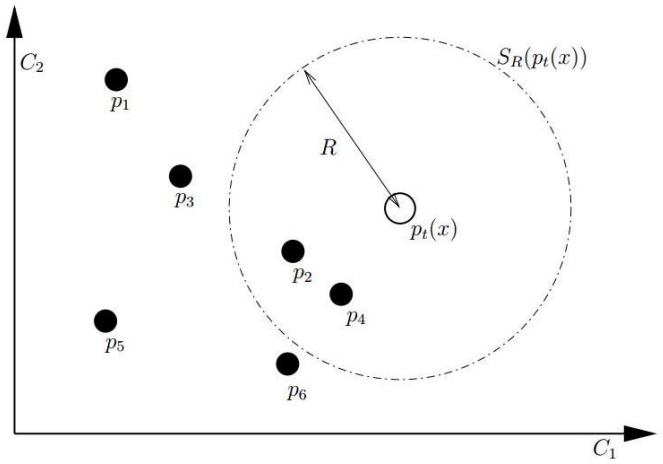
3.1 Princip

Označme si hodnotu pixelu x v čase t jako $p_t(x)$. ViBe nepočítá hustotu pravděpodobnosti jako spousta pokročilých technik zmíněných v kapitole 2, místo toho používá sadu vzorových hodnot k sestavení modelu pozadí. Abychom určili hodnotu $p_t(x)$, porovnáme ji k nejbližším hodnotám v sadě vzorků tím, že si vytvoříme oblast $S_R(p_t(x))$ s poloměrem R se středem v $p_t(x)$. Hodnota pixelu je označena za pozadí,

pokud kardinalita ♯, daná jako průnik oblasti $S_R(p_t(x))$ a sady vzorků $\{p_1,p_2,\ldots,p_n\}$, je větší než daná hodnota prahu $\sharp_{min}$ [7]. Formálně popsáno, hodnotu $\sharp_{min}$ porovnáváme s

$$\sharp\left\{S_R(p_t(x)) \cap \{p_1, p_2, \ldots, p_n\}\right\}$$

Dva parametry určující přesnost modelu pozadí jsou poloměr R a minimání kardinalita $\sharp_{min}$. Na základě experimentů bylo prokázáno, že poloměr $R=20$ a kardinalita $\sharp_{min}=2$ poskytují nejlepší výsledky. Tyto parametry není třeba měnit při aktualizaci modelu a není třeba je měnit ani pro různé části aktuálního snímku.



Obr. 4: Abychom mohli přiřadit pixel $p_t(x)$ do popředí nebo pozadí, počítáme počet vzorků v oblasti dané poloměrem R se středem v $p_t(x)$ [7]

3.2 Inicializace modelu

Jak již bylo zmíněno, ViBe použije první snímek k inicializaci modelu a začne odečítat pozadí již od druhého snímku videosekvence. Jelikož ale nemáme k dispozici žádné informace o modelu pozadí, využívá se okolních hodnot každého pixelu k nahrazení vzorků. ViBe pro každý pixel vybere N náhodných hodnot v jeho 8-okolí a tím nahradí vzorky. Sada vzorků pro první snímek může být popsáná následovně

$$M_0(x) = \{p_{t0}(y|y \in O(x))\}$$

$O(x)$ je okolí pixelu x. Výsledkem této strategie je rychlejší sestavení modelu, ale pokud jsou v prvním snímku pohybující se objekty, může dojít k vyhodnocení nesprávného pohybu, tzv. oblasti duchů (ghost regions).

Obr. 5: Příklad oblastí duchů – bílé plochy na obrázku vpravo špatně vyhodnoceny jako pohyb, jelikož tato konkrétní videosekvence obsahuje pohybující se objekty již při prvním snímku. Tyto oblasti s postupem času přechází do pozadí.

3.3 Aktualizace modelu

Aby byl model schopen vracet přesné výsledky a uměl si poradit s novými objekty ve scéně, měl by být pravidelně aktualizován. ViBe se řídí konzervativním přístupem aktualizace modelu, což znamená, že informace o popředí nejsou v modelu pozadí uloženy. Pokud je pixel označen jako pozadí, má šanci $1/\varphi$, že aktualizuje model. Vzorek, který bude nahrazen, je vybrán náhodně. Tato metoda však může způsobit, že nové objekty nikdy nebudou zahrnuty do scény a zmizí. Proto pokud byla hodnota pixelu určena k aktualizaci sady vzorků pixelu x, bude také použita k aktualizaci sady vzorků náhodně vybraného pixelu z 8-okolí pixelu x. Tímto je zajištěn určitý stupeň prostorové konzistence v celém modelu pozadí.

3.4 Výhody

Jedna z nevětších výhod algoritmu ViBe je velmi nízká výpočetní zátěž. Operace jsou omezeny pouze na odečítání, proto je schopen zvládnout zpracovávat video i v reálném čase.

Jedná se o bezparametrovou metodu odečítání pozadí, není třeba nastavovat či přizpůsobovat parametry modelu pozadí zvlášť pro každý videosoubor. Všechna videa mohou být zpracována se stejnými parametry a přitom jsou zachovány přesné výsledky.

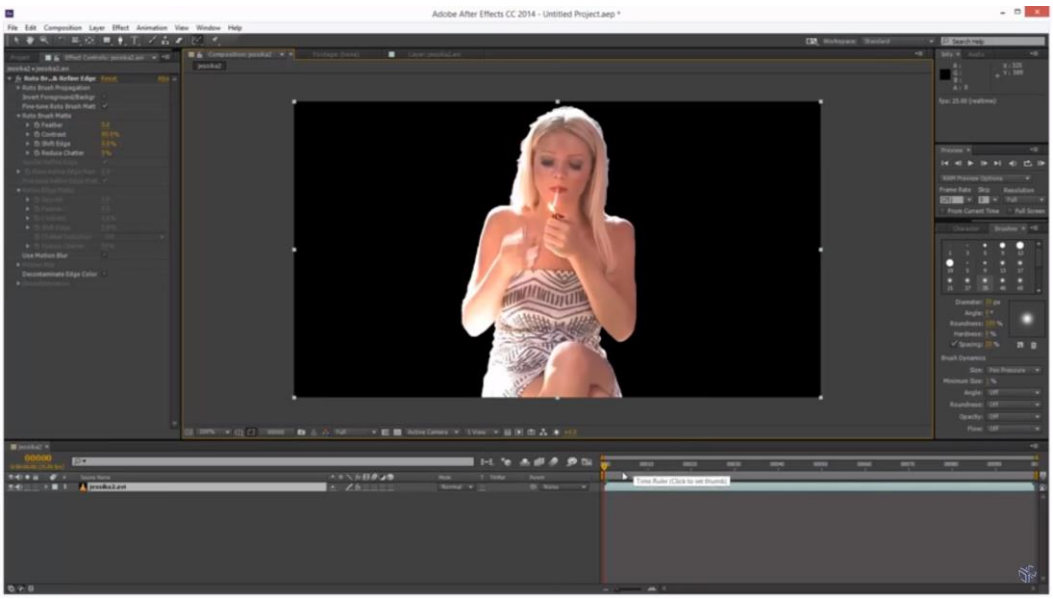
ViBe je pixelově založená metoda s okamžitou inicializací a odolností vůči šumu, což oproti ostatním moderním metodám vede k rychlejšímu zpracování obrazu a lepším výsledkům.

3.5 Využití

Tento algoritmus může být využit v široké škále aplikací, veškeré zmíněné příklady jsou však pouze potenciální možnosti využití, jelikož je algoritmus relativně novou záležitostí v poli detekci pohybu a zatím není příliš rozšířen. Ve většině případů se tedy nejedná o konkrétní využití algoritmu.

Nejrozšířenější pole působnosti algoritmů detekce pozadí jsou bezpečnostní systémy. Rozpoznání pohybu je nejdůležitější částí bezpečnostních systému, ale použití ve venkovním prostředí může způsobit problémy. ViBe si však s těmito problémy dokáže poradit a spolu s nízkou výpočetní náročností je vhodný i pro systémy s mnoha bezpečnostními kamerami (např. nákupní centra nebo letiště).

Jedna z dalších možností využití je zpracování videa. Každý počítač může využít algoritmu k detekci pohybu a poté aplikovat různé efekty (například aplikovat masku na pohybující se objekt a odstranit tak jeho pozadí).



Obr. 6: Detekce popředí v Adobe After Effects CC 2014, jedná se pouze o příklad, jak by mohl být algoritmus ViBe využit u programů pro střih a úpravu videí. Manuální vytvoření masky popředí je zdlouhavý a náročný proces, který se musí dělat pro každý snímek videa. S použitím algoritmu ViBe by se dala tato práce automatizovat a ulehčit tak práci uživatelovi.

Výrobci digitálních kamer či fotoapárátů by také mohli využít algoritmu ViBe a to jak pro profesionální tak i amatérské učely. Algoritmus by mohl být naimplementován do operačního systému daného zařízení a využit například při zaostřování pohybujících se objektů.

Tohoto by se také dalo využit ve filmovém průmyslu. Pokud by byl do digitálních kamer naimplementován algoritmus ViBe, který by automaticky zaostřoval popředí, mohl by ulehčit spoustu práce s následnými úpravami ve studiu.

Obr. 7: Konkrétní použití algoritmu ViBe u fotoaparatu značky Canon. Algoritmus v reálném čase automaticky detekuje popředí (které je v tomto případě označeno zelenou barvou).

Ve videoherním průmyslu můžeme také nalézt praktické využití detekce pohybu. Za zmínku stojí například kamera Kinect pro Xbox, za kterou stojí Microsoft. Celá tato technologie je postavená na rozpoznání hlasu a pohybu.

Obr. 8: Využití detekce pohybu ve hře Dance Central 3 pro Xbox 360. Kamera Kinect snímá uživatelův pohyb a porovnává ho s pohybem modelů ve hře. Čím větší shoda, tím víc bodů hráč získá.

Další využití algoritmu ViBe může být v dopravě a to například monitorování dopravních komunikací. Konkrétním příkladem může být monitorování křižovatky za

účelem zjištění, zda by nebylo vhodnější ji nahradit kruhovým objezdem. Algoritmus by mohl detekovat pohybující se automobily a výsledek zakreslovat do grafu v průběhu času. Z tohoto grafu by se poté dalo zjistit, jak moc je křižovatka vytížena.

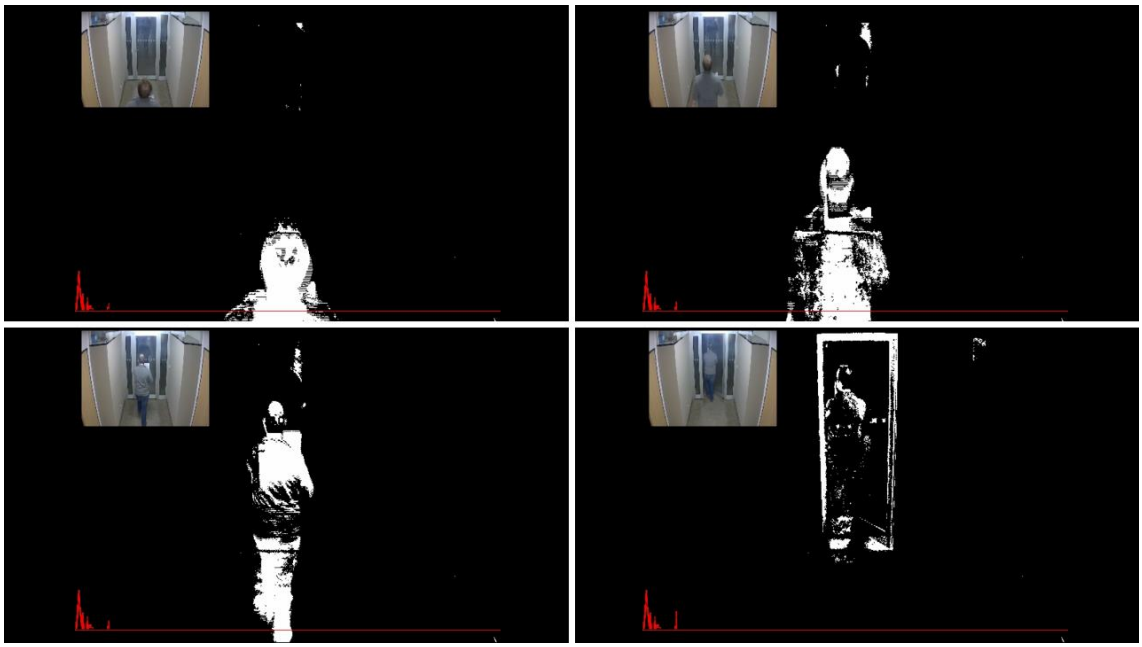
Obr. 9: Snímek záznamu ze statické kamery monitorující křižovatku vhodný pro aplikaci algritmu ViBe. Namísto použití člověka, který by počítal, kolik za den projede automobilů, by se dalo využít záznamu ze statické kamery, aplikovat algoritmus pro detekci pohybu, výsledek vykreslovat do grafu a posléze zhodnotit, zda by nebylo lepší nahradit tuto křižovatku kruhovým objezdem.

4 NAVRŽENÉ ŘEŠENÍ A IMPLEMENTACE DETEKCE POHYBU

Hlavní cíl programu je zpracování vstupního videa (Obrázek 10) na výstupní video (Obrázek 11). K tomuto účelu byla nastudována teorie z předchozích kapitol, použita sada programovacích nástrojů a knihoven popsaná v této kapitole a pořízena a stažena sada různě zajimavých videí z hlediska detekce pohybu. To vše bylo následně otestováno a vyhodnoceno. Na závěr byl tento upravený algoritmus detekce pohybu implementován do demonstrační aplikace.

Obr. 10: Snímky vstupního videa

Při zpracování videosouboru se vychází z předpokladu, že je záznam pořízen stacionární kamerou. Jakýkoliv pohyb či malé zachvění kamery může mít za následek zkreslené výsledky systému. Video nesmí být nijak sestříháno a jeho části nesmí být vizuálně upravovány (pouze video jako celek může být upraveno). Jakýkoliv střih znamená změnu pixelů oproti referenčnímu modelu pozadí, což je vyhodnoceno jako pohyb. Jakákoliv náhlá změna barev videa (aplikace vizuálního efektu) je opět nesprávně vyhodnoceno jako pohyb.

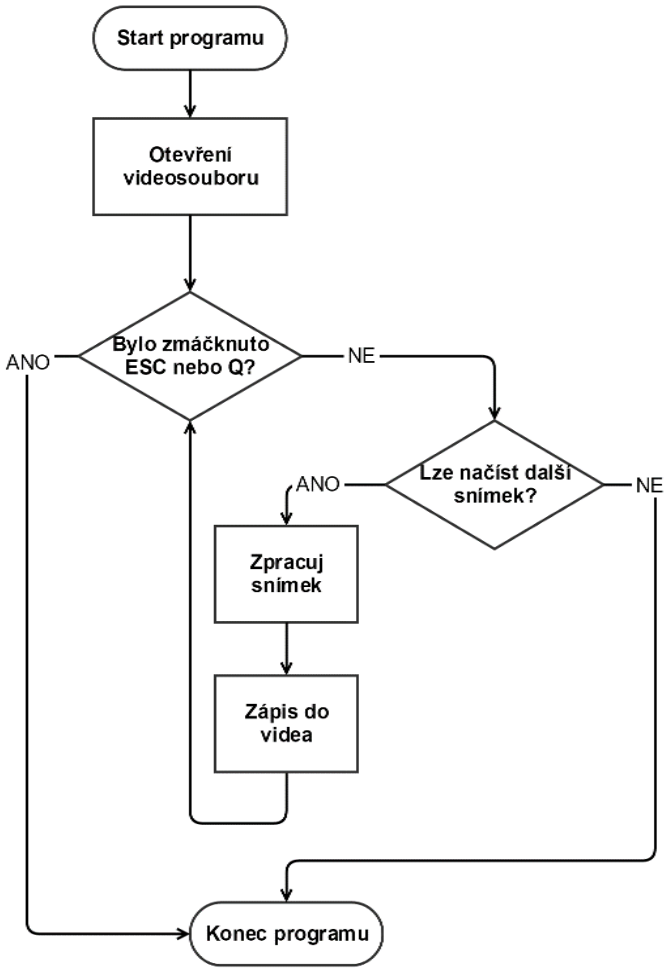


Obr. 11: Snímky výstupního videa

Pro zpracování obrazu byla po zvážení zvolena knihovna OpenCV ve verzi 3.0. Jedná se o knihovnu dostupnou jak pro komerční tak i akademické účely. Má rozhranní pro jazyky C++, C, Python a Java. Podporuje běh na operačních systémech Windows, Linux, Mac OS, iOS a Android. Pro implementaci byl původně zvolen jazyk C++ a integrované vývojářské prostředí Microsoft Visual Studio 2015 Community (VS 2015). Verze OpenCV 3.0 však neměla knihovny určené pro VS 2015, proto byly použity knihovny pro Microsoft Visual Studio 2013 Community (VS 2013), které způsobovaly chyby při překladu projektu a načítání videosouborů. Po odstranění chyb při překladu se vyskytly nové chyby při ladění projektu. Jelikož ladění projektu je jedna z nejdůležitějších částí implementace a Visual Studio má velmi kvalitní prostředí pro ladění, bylo zvoleno VS 2013 jako vývojářské prostředí.

K pořízení záznamů byla použita integrovaná webkamera notebooku nahrávající videa v rozlišení 640x480 pixelů, dále digitální fotoaparát značky Olymp nahrávající videa v rozlišení 1920x1080 pixelů a výzkumné kamery fakulty informačních technologií VUT v Brně nahrávající videa v rozlišení 1920x1080 pixelů. Všechny tyto kamery pořídily videa interiéru. Pro testování bylo také staženo několik videí venkovního prostředí. Videa byla zpracována na třech počítačích s různými hardwarovými specifikacemi.

Struktura výsledného programu je rozdělena do 3 částí, které byly postupně implementovány a rozšiřovány. Mezi tyto části patří načítání vstupního videa, zpracování každého snímku a následné upravení výstupu programu.



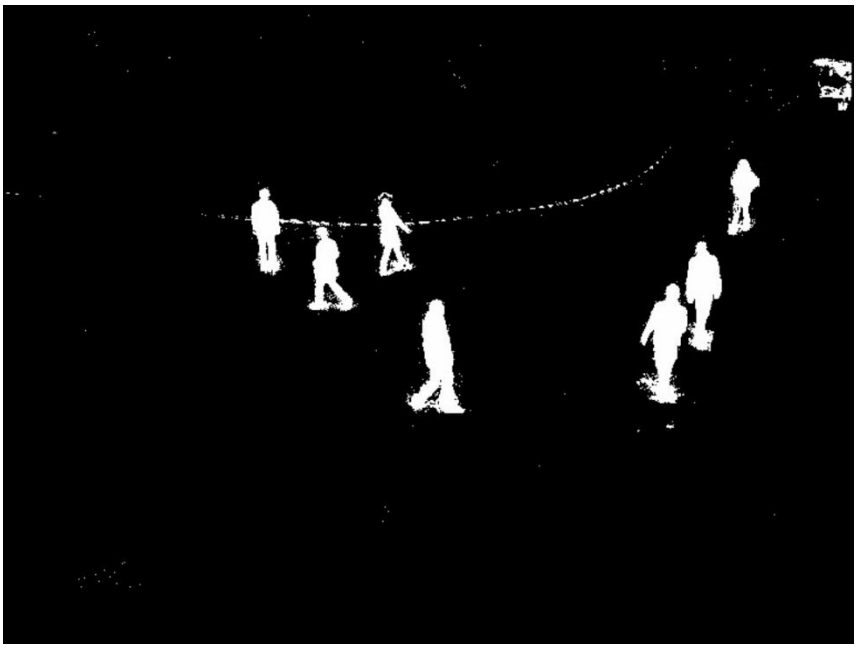
Obr. 12: Vývojový diagram programu pro detekci pohybu

Pro načtení videa do programového prostředí byla využita třída VideoCapture, která je součástí knihovny OpenCV. Výhodou této třídy je také možnost načíst video jako posloupnost obrázků nebo ji lze využít jako vstupní rozhraní pro zobrazení živého přenosu z připojeného zařízení (například webkamery). Tato třída dokáže načítat širokou škálu formátů videí, konkrétně testovanými byly formáty MP4, AVI, MPG a WMV.

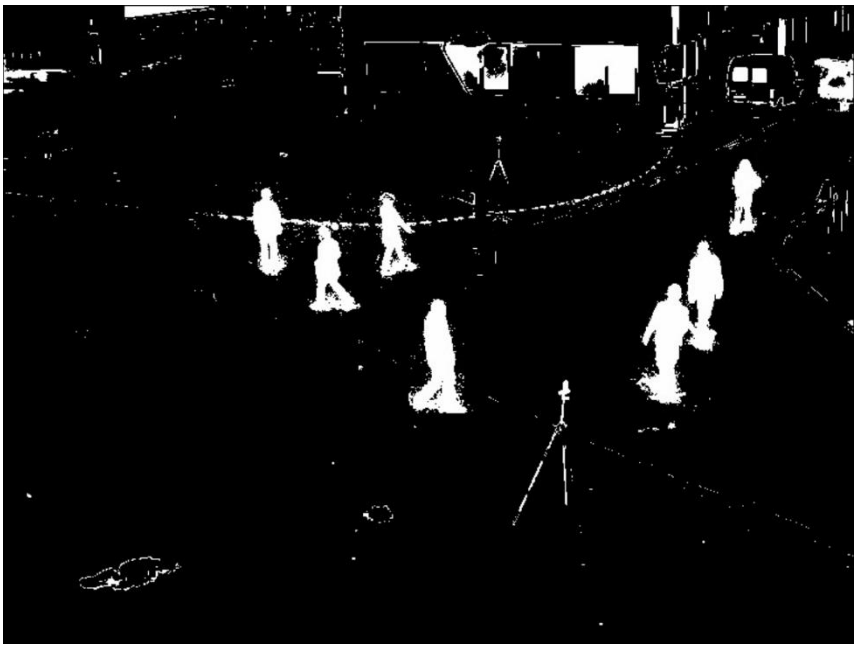
Po otevření videa je okamžitě načten první snímek, který slouží k inicializaci modelu ViBe, jak již bylo zmíněno v předchozí kapitole. Poté se přechází na hlavní smyčku programu – zpracování snímku videa.

Nedůležitější částí hlavní smyčky je detekce popředí aktuálního snímku. Algoritmus ViBe pracuje s několika proměnnými, které ovlivňují jeho výstup: celočíselná hodnota R, která slouží k porovnávání pixelů a rozhodování, zda se jedná o popředí či pozadí, a proměnná noMin, která udává po kolika snímcích přechází daný pixel do pozadí (pokud se jeho hodnota nemění). Jak moc tyto proměnné ovlivňují výstup lze vidět na obrázcích 14,15,16 a 17.

Obr. 13: Snímek před zpracováním algoritmem ViBe

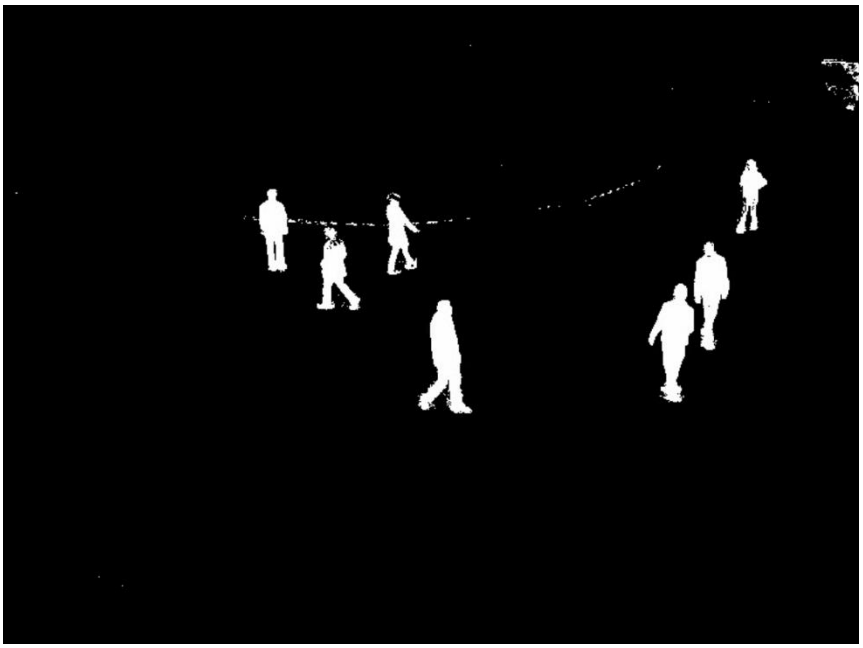


Obr. 14: Výstup algoritmu ViBe s proměnnými *R=10*, *noMin=1*

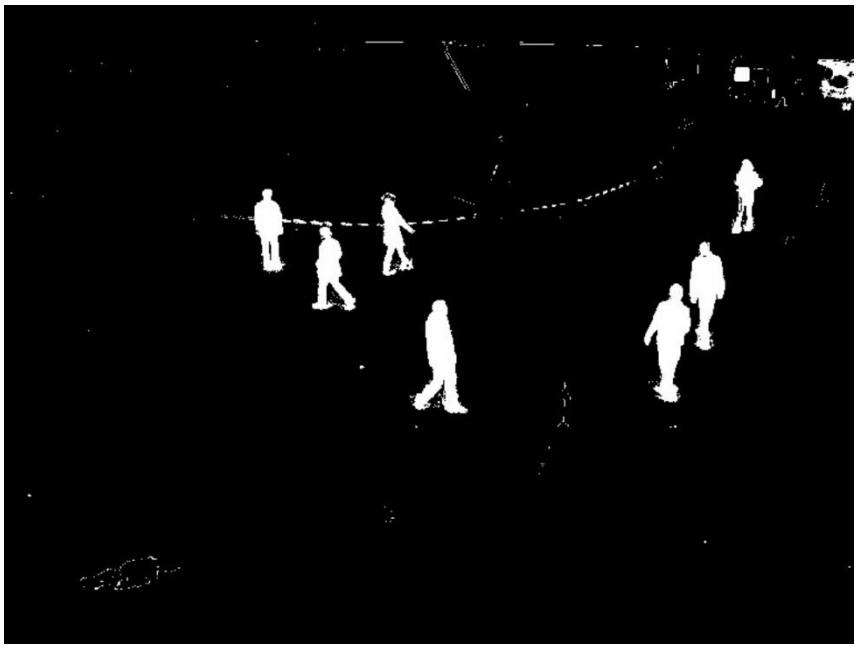


Obr. 15: Výstup algoritmu ViBe s proměnnými *R=10, noMin=2*

Při použití proměnných R=10, noMin=2 vrací program téměř nepoužitelný výstup. V tomto případě by program detekoval pohyb po celou dobu videa. Pixely porovnávají své okolí s malým poloměrem (10), což znamená, že zpracování každého snímku trvá déle a jednotlivé neměnné pixely přechází do pozadí až po dvou snímcích. Některé pixely jsou tedy porovnávany vícekrát a to může být výpočetně náročné.



Obr. 16: Výstup algoritmu ViBe s proměnnými *R=20, noMin=1*

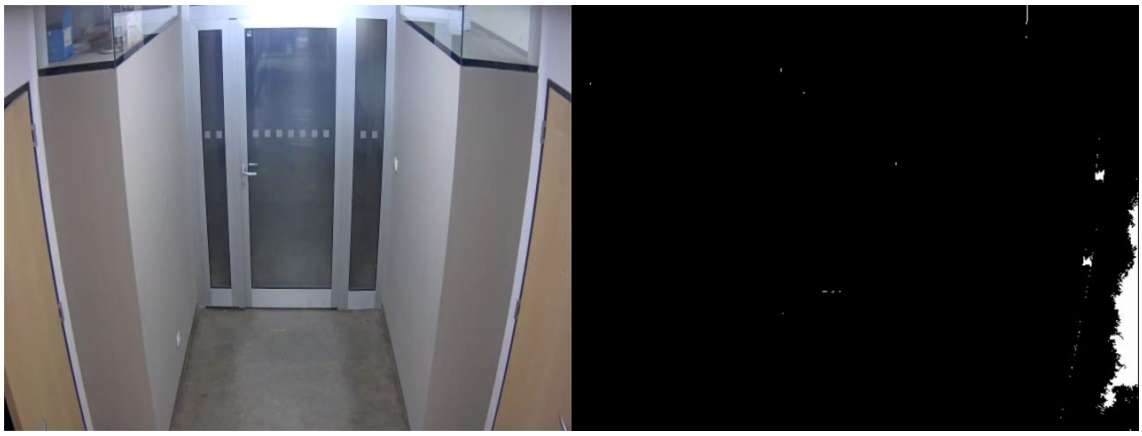


Obr. 17: Výstup algoritmu ViBe s proměnnými *R=20, noMin=2*

Na základě předchozích obrázků lze vidět, že ViBe vrací nejlepší výsledky při použití hodnoty 1 u proměnné *noMin*. Toto ovšem platí pouze v případě, že objekty mění svoji polohu při každé změně snímku. Pokud by se objekty pohybovaly pomaleji, začaly by postupně přecházet do pozadí a pokud by jejich rychlost klesla pod určitou hranici, pohyb by nebyl rozpoznán, což může mít vážné následky (například krádež v obchodě).

V případě implementace programu byly zvoleny parametry *R=20, noMin=1*. Jelikož program ignoruje rychlou změnu světelných podmínek, je nezbytné aby model na tuto změnu reagoval rychle. Pokud by bylo noMin větší než 1, algoritmu by trvalo déle než by se této změně přizpůsobil a pokud by mezi tím něco způsobilo pohyb, nebyl by detekován (viz. Obr. 18).

Obr. 18: Výstup algoritmu ViBe 5 vteřin po bliknutí světla (*noMin=2*)

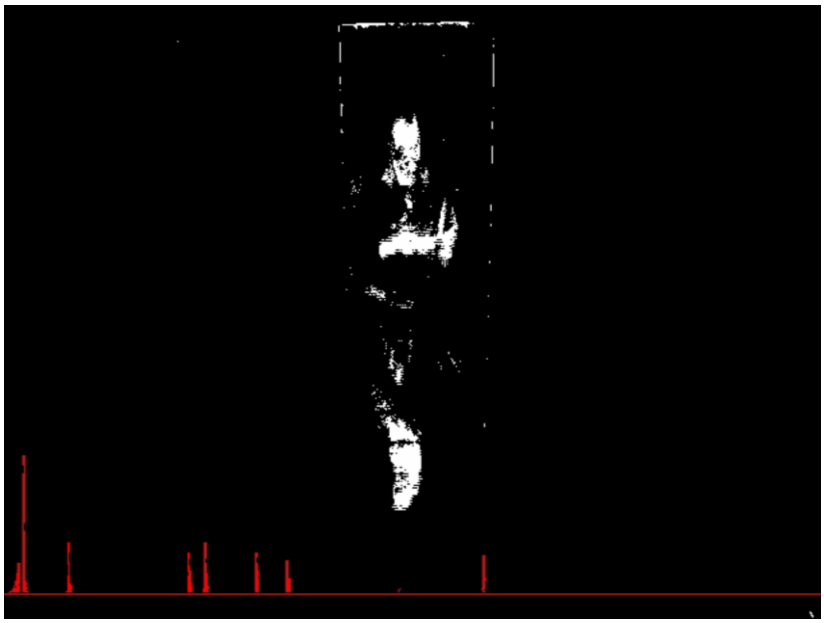


Obr. 19: Výstup algoritmu ViBe 5 vteřin po bliknutí světla (*noMin=1*)

Jakmile je snímek zpracován, je třeba odstranit šum a drobný pohyb způsobený například chvěním nebo odlesky. Toho je docíleno programově (vizuální výstup zůstává stejný) detekcí okrajů objektů a vypočtení kontur. Pokud je plocha kontur větší než určitá hranice, jedná se o pohyb.

Tato hranice určuje citlivost detekce pohybu, čím menší hodnota, tím vetší citlivost. Nelze určit univerzální hranici, která by zajistila správnou detekci pohybu ve všech případech. Mějme například záznam z kamery snímající vzdálené objekty. Pokud bychom nastavili příliš vysokou hranici, pohyb by nebyl detekován. Proto je tato hranice vypočtena z rozlišení videa a stanovena na optimální hodnotu, což je dvacetina výšky obrazu videa.

Aby se dalo ověřit, zda-li tato metoda funguje správně, je třeba to nějakým způsobem uživateli sdělit. Jako nejvhodnější možnost bylo vybráno vykreslení grafu přímo do výstupního videa. Jakmile je identifikován pohyb, začne se vykreslovat graf, jehož velikost je přímo úměrná ploše pohybujících se objektů.



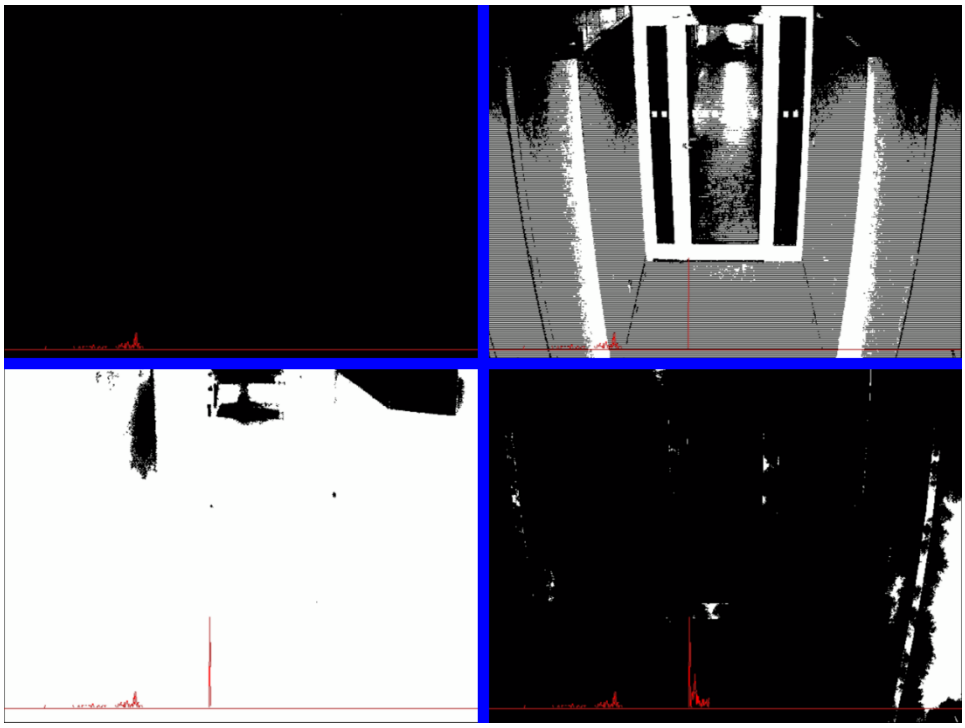
Obr. 20: Vykreslování grafu do výstupního videa

Z obrázku 20 lze jednoduše vyčíst, kdy konkrétně nastal pohyb a to pomocí osy X, která znázorňuje časovou osu videa. Stačí tedy video posunout do konkrétního bodu křivky grafu a ověřit, zda se opravdu jedná o pohyb. V tomto konkrétním grafu je maximum několikanásobně vetší než průměrné hodnoty, což znamená, že plocha bílých pixelů musí pokrývat většinu obrazu. Při přesunu do tohoto momentu se dá zjistit, že se jedná o rychlou změnu světelných podmínek (někdo rozsvítil), viz. Obr. 21.

Obr. 21: Posloupnost snímků při změně světla

Metoda odečítání pozadí je velmi citlivá na změnu světelných podmínek. Jelikož se jedná o metodu, která porovnává pixely aktuálního snímku s modelem pozadí, jakákoliv sebemenší změna pixelu je vyhodnocena jako pohyb. Jedním z výsledků této práce je návrh a implementace vlastní metody, která takovou změnu ignoruje.

Při změně světla v interiéru vycházíme z předpokladu, že se změní celá scéna (každý pixel bude mít jinou hodnotu). Výstup programu v určitém momentu tedy bude kompletně bílý (celá plocha obrazu videa je detekována jako pohyb). Na základě tohoto předpokladu lze navrhnout a naimplementovat jednoduchou metodu, která bude tento nesprávně vyhodnocený pohyb ignorovat.



Obr. 22: Výstup programu při změně světla

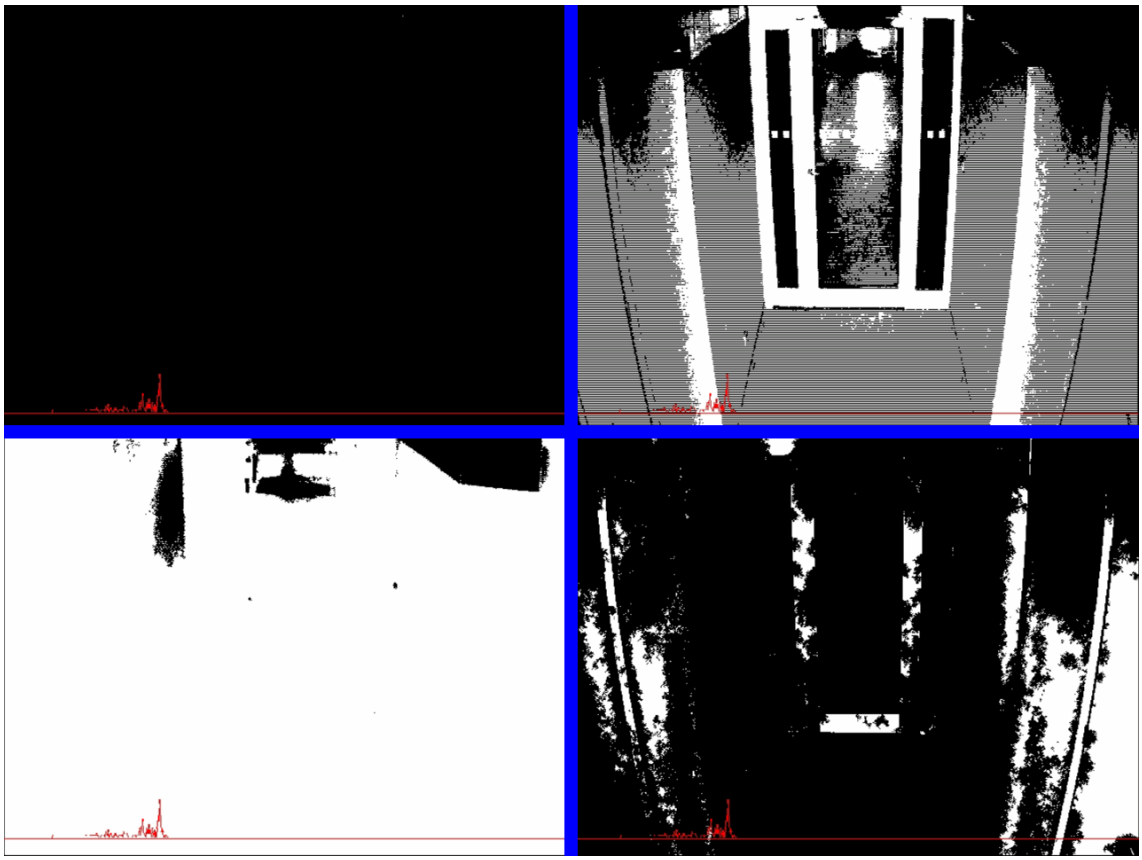
Vycházíme ze tří případů, které mohou nastat:

1. Byl detekován pohyb a poté došlo ke změně světla
2. Došlo ke změně světla a poté byl detekován pohyb
3. Změnilo se světlo, ale nebyl detekován žádný pohyb

Nejvhodnější řešení tohoto problému je detekce jakýchkoliv světelných podmínek. Tímto se dá vyhnout složitým programovým strukturám pro zjišťování pohybu před nebo po rozsvícení, které ani nemusí být přesně vyhodnoceno (nelze určit časový úsek, kdy již změna světla není součástí následného pohybu).

Do hlavní smyčky programu byla přidána podmínka a několik proměnných, které si ukládají plochu pohybujících se objektů aktuálního a předchozího snímku. Pokud je rozdíl těchto dvou ploch větší než určitá hranice (vypočtená z rozlišení videa), jedná se o světelnou změnu. Jakmile je světlo detekováno, program určitou dobu (100 snímků) přestane detekovat pohyb, aby se ViBe mohl přizpůsobit změně.

Tento přístup má však nevýhodu. Pokud nastane změna světla, za kterou bude následovat pohyb, je zde šance, že pohyb nebude detekován. Tento pohyb však musí být velmi rychlý, jelikož doba regenerace programu je pouhých 100 snímků.



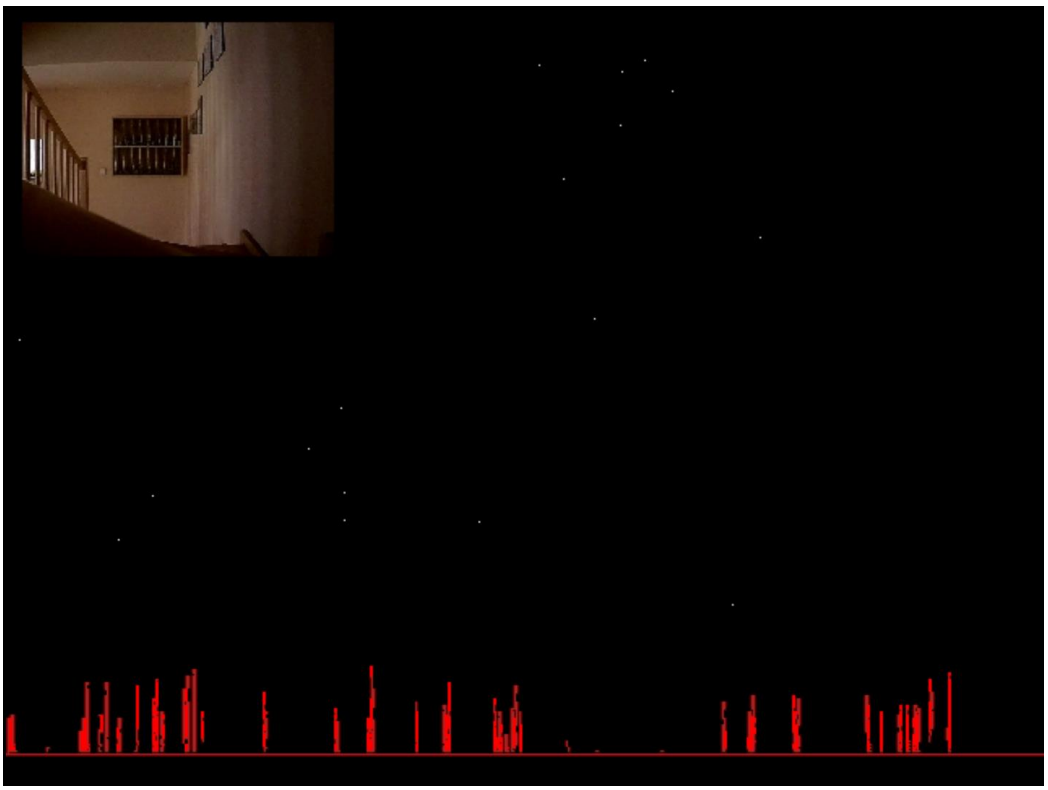
Obr. 23: Ignorování změny světla

Pro ověření, zda pohyb skutečně nastal, je třeba nahlédnout do původního videa a najít konkrétní okamžik, což je neefektivní a časově náročné. Proto je do výstupu přidáno okno zobrazující originální snímek, které usnadňuje kontrolu. To vše se následně zapisuje do výstupního videosouboru zadaného jako parametr při spuštění.

Obr. 24: Finální výstup programu

O zápis do výstupního videa se stará třída VideoWriter, která je také součástí OpenCV. Výstupní formát je ovšem ovlivněn sadou video kodeků nainstalovaných na zařízení, ze kterého je program spouštěn. Pro výstup byl vybrán kodek DivX, který nepodporuje všechny formáty, ale zachovává dobrý poměr kvalita/velikost. Ověřené formáty výstupního videa jsou AVI a WMV.

Při prvotních testech byl však zjištěn jeden velký nedostatek. Při světelných změnách, které neovlivňují celou scénu, ale pouze její část, program vrací výsledky s velkou chybovostí. Především se jedná o videa z interiéru za špatných světelných podmínek (viz. Obr. 25).

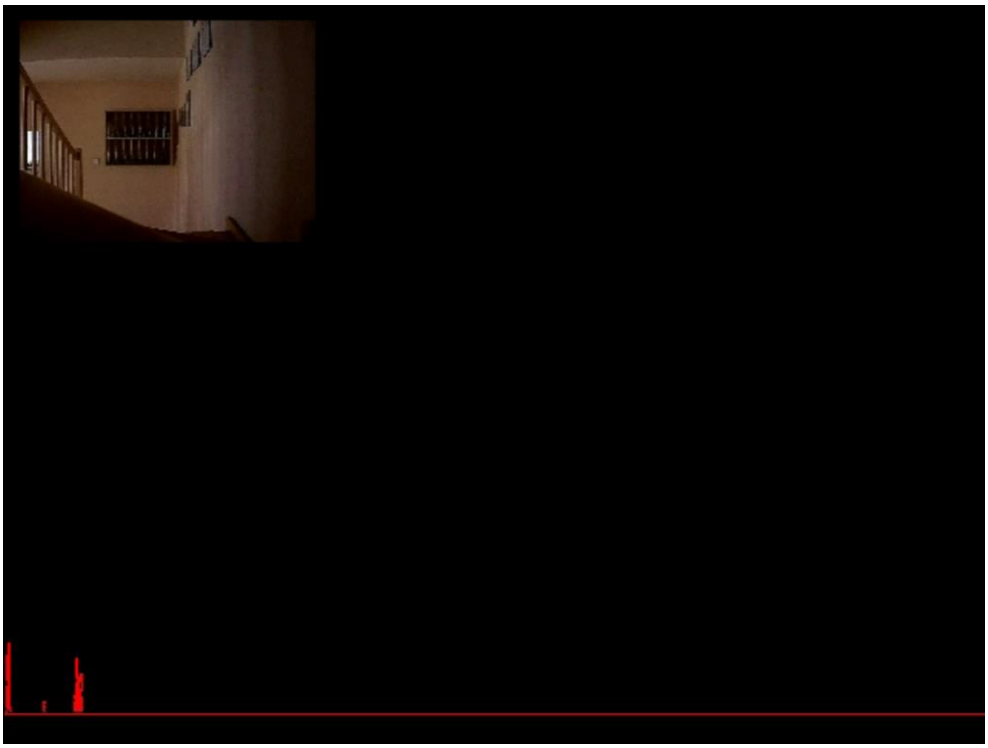


Obr. 25: Chybná detekce pohybu – pouze tři výchylky grafu jsou opravdový pohyb

Ve videu, jehož snímek můžeme vidět výše, došlo k 26 detekcím pohybu (každá výchylka grafu znázorňuje pohyb), z toho opravdový pohyb nastal pouze třikrát. Chybovost je tedy 88%, což je naprosto nepřípustný výsledek. Všechny chybné detekce byly způsobeny šumem a slabou změnou světelných podmínek.

Jako první způsob řešení tohoto problému bylo snížení citlivosti. Snížení citlivosti by tento nedostatek odstranilo, ale mohlo by způsobit, že některé vzdálenější objekty by nemusely být detekovány, což by mělo vážnější následky.

Aby se tomuto problému zabránilo a přitom byl program pořád schopen efektivně detekovat pohyb, byla aplikována funkce knihovny OpenCV pro redukci šumu, konkrétně eroze. Tato funkce zajistila efektivní řešení aniž by bylo třeba měnit parametry programu (Obr. 26).



Obr. 26: Výstup programu po aplikování eroze pro redukci šumu – chybný pohyb kompletně eliminován

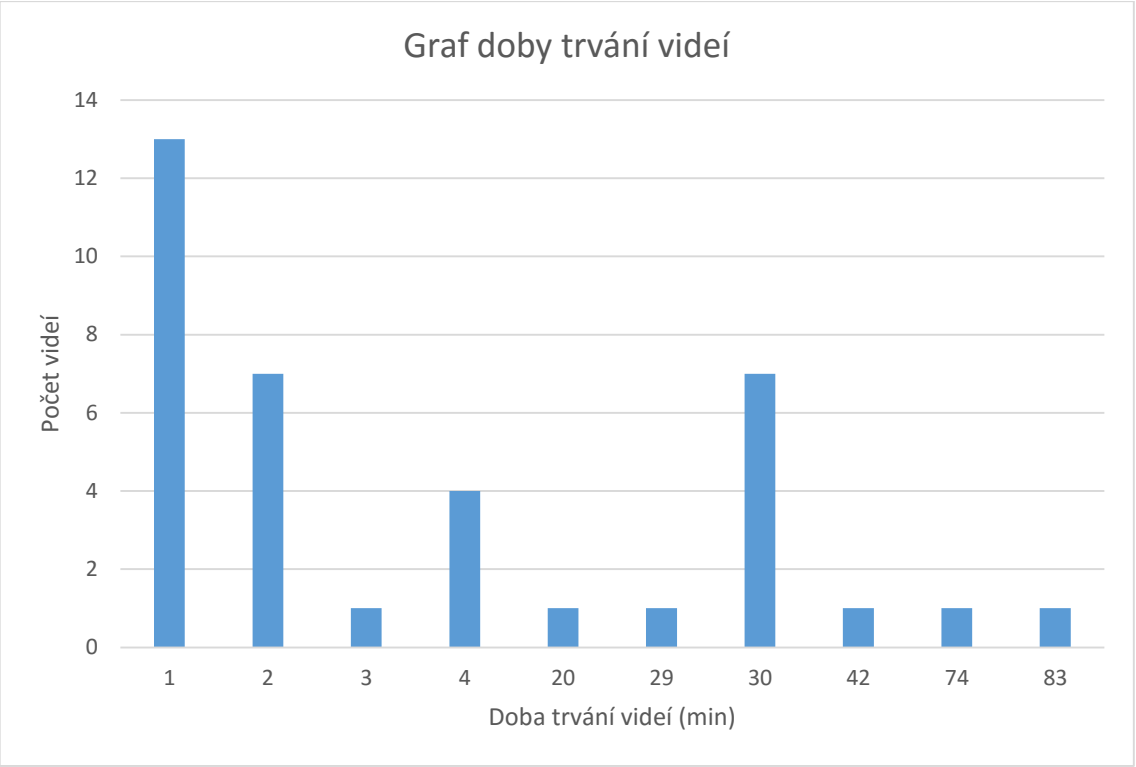
4.1 Datová sada pro testování

K závěrečnému testování bylo pořízeno 37 videí s celkovou dobou trvání 8 hodin 21 minut. Doba trvání jednotlivých videí je v rozmezí od 31s do 1h 23min. Všechna videa byla natočena statickou kamerou.

25 videí bylo staženo z webové stránky www.changedetection.net, což je videodatabáze určená k testování algoritmů detekce pohybu. Videa byla stažena z Datasetu 2012 a Datasetu 2014. Tato videa byla stažena jako posloupnost snímků ve formátu JPG. S využitím programu pro editaci videa byla tato posloupnost snímků převedena do formátu MP4.

5 videí bylo pořízeno digitálním fotoaparátem značky Olympus z domácího prostředí. Zbylých 7 videí bylo získáno za účelem testování z videoarchivu Fakulty informačních technologií VUT v Brně.

Videa jsou rozdělena do dvou kategorií: videa z venkovního prostředí a videa z interiéru a to v poměru 21:16. Dále byla tato videa rozdělena do několika kategorií na základě složitosti pohybu, který se v daných videích odehrává. Konkrétně se jedná o kategorie: základní, dynamické pozadí, stíny, špatné počasí a noční videa.



Obr. 27: Graf doby trvání jednotlivých videí

4.2 Výsledky

Závěrečné testování bylo rozděleno na dvě části: testování výkonu programu a testování přesnosti programu.

Test výkonu byl proveden na třech různých počítačích (dva notebooky, jeden stolní počítač). Pro tyto účely byla použita tři různá videa. Specifikace počítačů a videí viz. tabulky níže.

	Stolní počítač	Notebook 1	Notebook 2
Procesor	AMD Phenom II X4 965	Intel Core i3-	Intel Core i7-4510U
Počet jader procesoru	4	2	2
Frekvence jader	3,4 GHz	2,3 GHz	2 GHz
RAM	8 GB	4 GB	16 GB
Operační systém	Windows 7 64bit	Windows 7 64bit	Windows 10 64bit

Obr. 28: Tabulka specifikací počítačů, na kterých byl program testován

Z hlediska výkonů počítačů se dalo předpokládat, že nejlepší výsledky bude mít notebook č. 2, na druhém místě bude stolní počítač a poslední pozici obsadí notebook č. 1. Výsledky testu však přinesly překvapující výsledky.

	Video 1	Video 2	Video 3
Formát	MPG	MPG	MP4
Rozlišení (px)	640x480	640x480	1920x1080
Délka	1min 2s	30min 2s	1h
FPS	25	25	30
Velikost	39,1 MB	898 MB	354 MB

Obr. 29: Tabulka specifikací videí, která byla použita k testování

V této práci jsou použity třídy OpenCV, které zpracovávají obraz ze strany procesoru, proto v tabulce specifikací daných počítačů není třeba uvádět výkon grafické karty.

	Video 1	Video 2	Video 3
Stolní počítač	1min 1s	27min	5h 35min
Notebook 1	1min	28min	3h 34min
Notebook 2	44s	18min	2h 13min

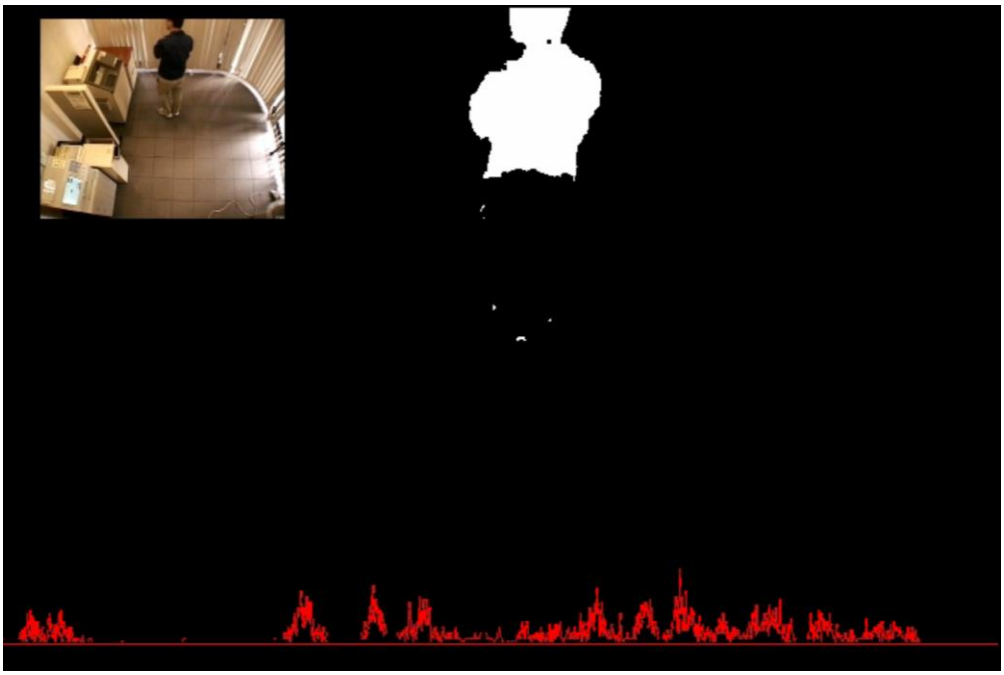
Obr. 30: Doba zpracování jednotlivých videí na počítačích určených k testování

Na všech počítačích byly všechny procesy na pozadí omezeny na minimum, běžel pouze průzkumník souborů a systémové procesy. U videa č. 1 a č. 2 program přinesl očekáváné výsledky. Videa byla zpracována v reálném čase, dokonce v menším předstihu. Video 3 však přineslo překvapivé výsledky. Vzhledem k rozlišení videa a počtu snímků za sekundu se dalo předpokládat, že program bude zpracovávat video o něco déle. Nepředpokládalo se však, že Notebook 1, který má značně nižší výkon než stolní počítač, zpracuje video rychleji a to téměř o dvě hodiny. Stolní počítač má sice číselně větší výkon než Notebook 1, ale má starší hardware a menší rychlost zápisu na disk.

Na základě výsledků těchto testů bylo stanoveno rozlišení videí u všech následujících testů a to na 640x480 pixelů. Jedná se kompromis mezi stále ještě dobrou kvalitou zobrazení a rychlostí zpracování videa. Všechny tyto následující testy byly provedeny na Notebooku 2.

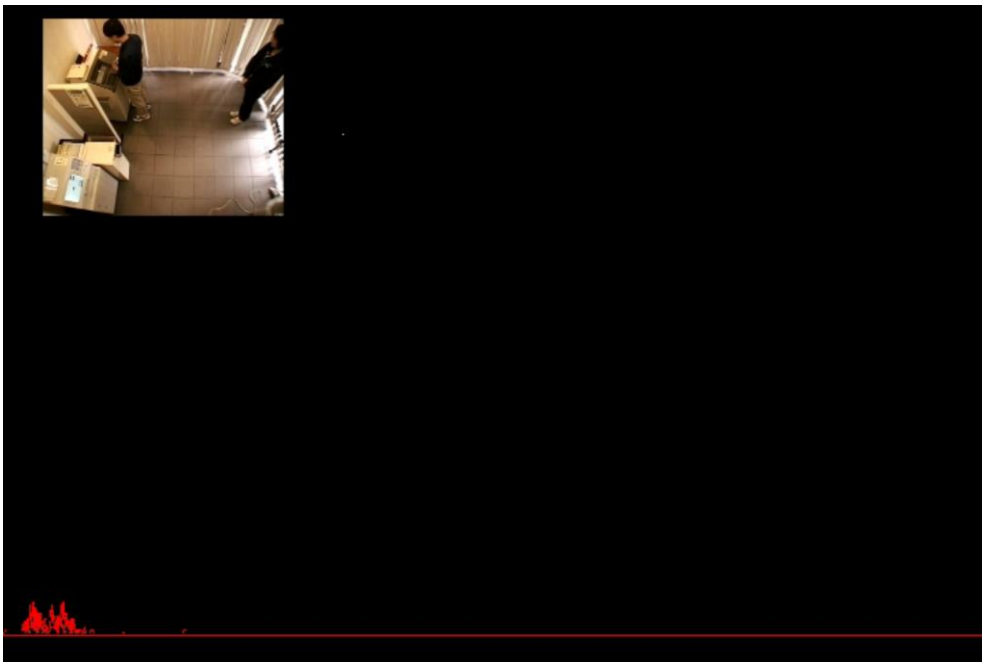
K testování přesnosti bylo pořízeno 37 videí. Program načítá na vstupu dva parametry: vstupní video a výstupní video, které je vytvořen během činnosti programu. Zadávat všechny parametry ručně pro každé video by bylo velice zdlouhavé a neefektivní, proto byl vytvořen .bat soubor, který je součástí datové sady a slouží k načítání všech videí. Tento soubor vytvoří složku pro výstupní videa a spouští program pro detekci pohybu, který načítá všechna MP4, AVI, MPG a WMV videa ve složce, která je předána proměnné `FOLDER`. Uvnitř této složky vytvoří nový adresář Output, do kterého bude program ukládat výstupní videa ve formátu AVI.

Zpracování všech videí zabralo necelé čtyři hodiny a přineslo různorodé výsledky. V konečném výsledku, program byl nejvíce přesný u videí z interiéru, kde byl téměř bezchybný.



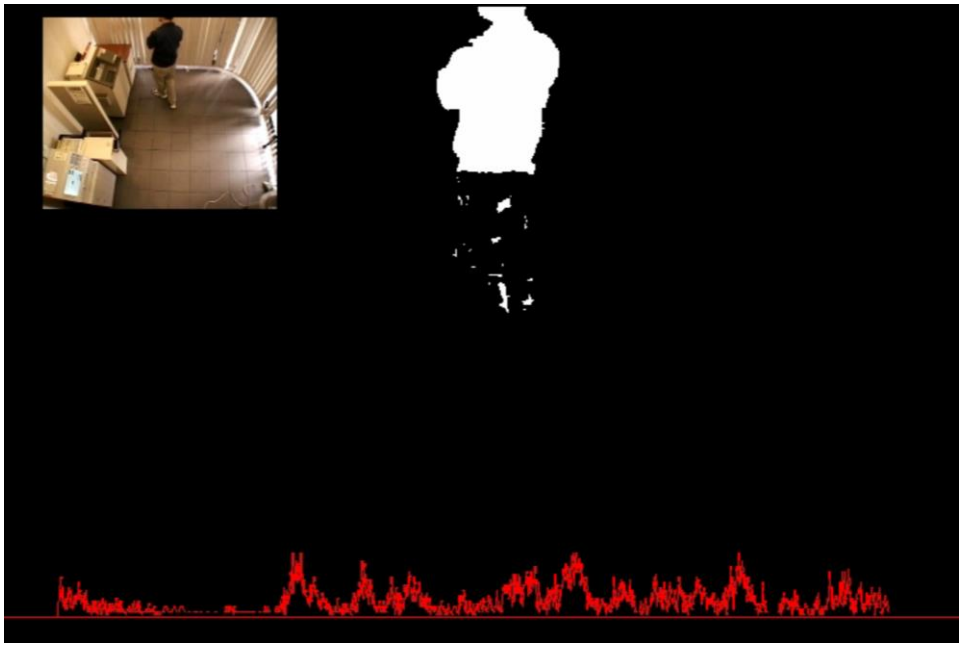
Obr. 31: Výstup programu u videa z interiéru, výsledný graf ukazuje, že téměř celou dobu probíhal pohyb

Na základě grafu u výstupního videa na obrázku č. 31 lze vidět, že pohyb probíhal téměř po celou dobu trvání videa. Pokud bychom se zaměřili na prázdná místa grafu, zjistili bychom, že pohyb sice nenastal, ale ve scéně se stále daná osoba vyskytuje. Toto je způsobeno proměnnou *noMin*, která je nastavena na hodnotu 1, což znamená, že objekty, které se přestanou pohybovat, se velmi rychle propadají do pozadí (Obr. 32). To však není nijak závažný problém, objekt bude pouze znovu detekován, pokud změní svoji pozici.



Obr. 32: Výstup programu u videa z interiéru - jakmile se osoby přestanou pohybovat, propadají se do pozadí

Tomuto se dá částečně zabránit a to zvýšením proměnné noMin. Jak lze vidět na obrázku 33, tato změna nepřináší příliš velké rozdíly. Zvětšily se maximální hodnoty v grafu, ale program reaguje na změny pohybu pomaleji a nelze jej již použít pro detekování světelných změn.



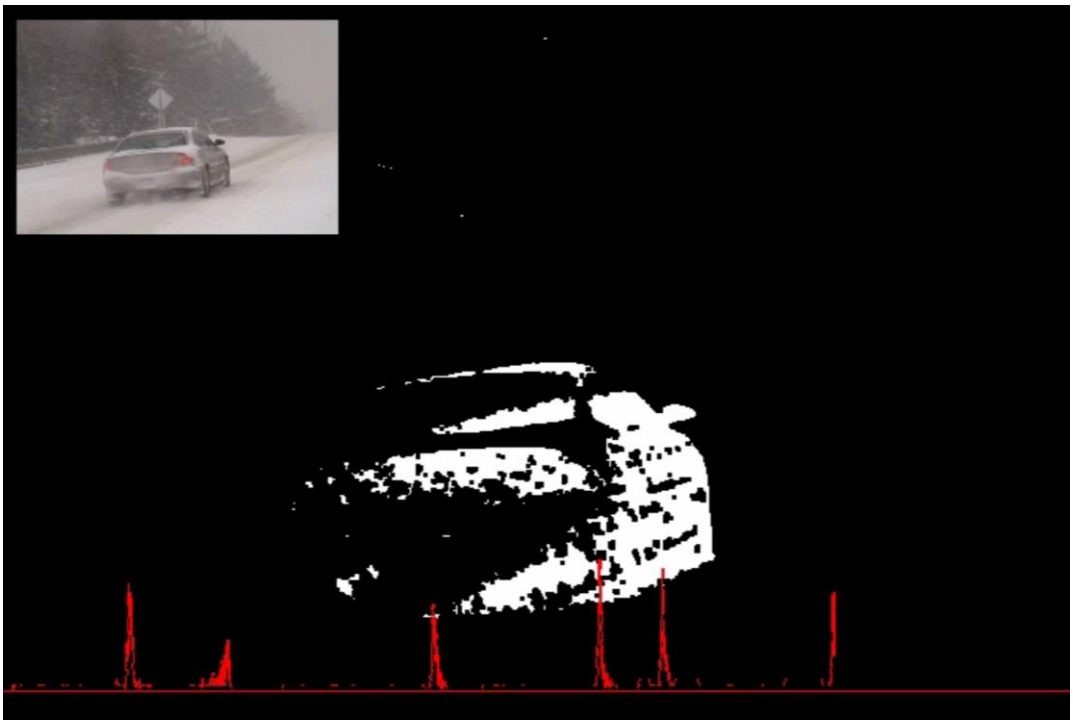
Obr. 33: Výstup programu u videa z interiéru – použití programu s proměnnou noMin=2

Detekce pohybu v interiéru je relativně jednoduchou záležitostí. Program vracel velice přesné výsledky (výstupní videa jsou součástí přílohy). Občas nastala situace, že redukce šumu způsobila ještě rychlejší přechod objektu do pozadí, ale neovlivnilo to přesnost programu, jelikož objekt byl předtím správně detekován a při dalším pohybu byl detekován znovu.

Praktická část této práce je především zaměřena na spolehlivou, efektivní a přesnou detekci pohybu v interiéru. Algoritmus ViBe je ale také velmi efektivní i při detekci pohybu ve venkovním prostředím s dynamickým pozadím. Proto byla také provedena série testů na videích z venkovního prostředí, která dokazuje, že i přes některé nedostatky je tento algoritmus velice perspektivní metodou detekce pohybu.

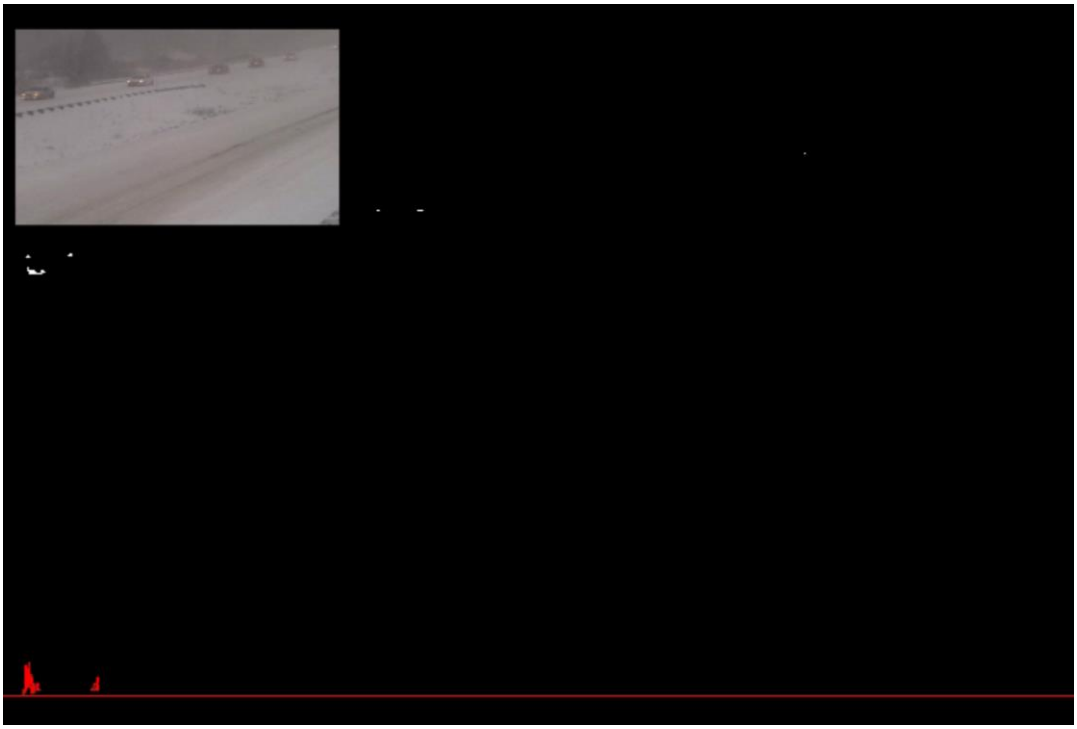
Všeobecným problémem detekce pohybu je dynamické prostředí. Kombinace citlivosti detekce pohybu a redukce šumu programu tento problém téměř odstraňuje. Tento přístup má ovšem i své nevýhody.

V kategorii videí obsahující špatné počasí program vracel velice přesné výsledky při detekci blízkých objektů (Obr. 34), avšak téměř nebyl schopen detekovat vzdálenější objekty (Obr. 35). Husté sněžení či déšť výrazně ovlivňuje viditelnost objektů ve scéně, proto vzdálenější objekty se velmi rychle stávají součástí pozadí a proto jejich plocha někdy ani nepřesáhne hranici citlivosti a nejsou detekovány.



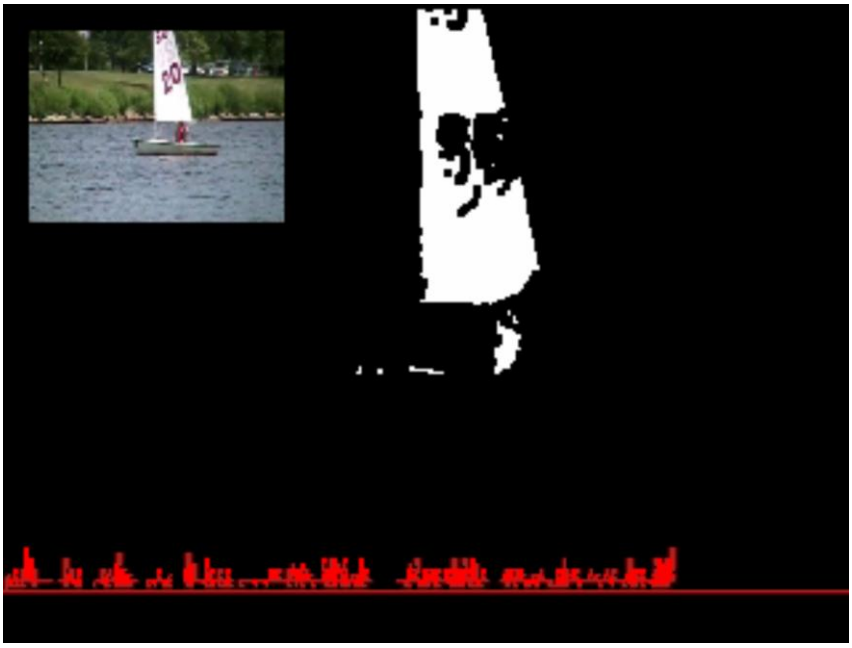
Obr. 34: Výstup programu při detekci blízkých objektů za špatného počasí

Objekty, které se na scéně objeví velmi rychle a zaberou velkou plochu, jsou detekovány velmi přesně, ale také se začnou postupně stávat součástím pozadí a ke konci jejich existence ve scéně také nejsou detekovány.



Obr. 35: Výstup programu při detekci vzdálených objektů za špatného počasí

Testy programu v kategorii videí s dynamickým pozadím přinesly velmi různorodé výsledky. V některých případech program vracel velmi přesné výsledky a v některých případech byl výstup velmi zkreslený.



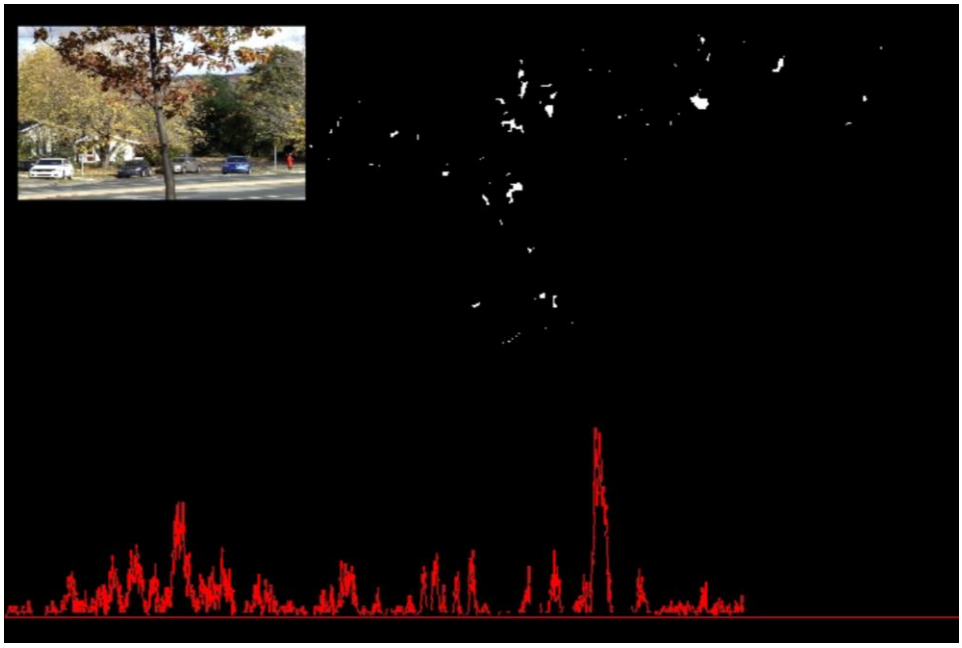
Obr. 36: Výstup programu v kategorii videí s dynamickým prostředím

Z obrázku č. 36 lze vyčíst, že si program dokáže velmi dobře poradit s vodní hladinou a to především díky odstranění šumu. Na základě křivky grafu však nelze odhadnout, zda je pohyb způsoben lodí v popředí nebo pohybem automobilů v pozadí. Každopádně se jedná o pohyb, který je přesně detekován.

Velmi dobře také dokáže odfiltrovat nežádoucí pohyb na popředí a přesně detekovat pohyb automobilů na pozadí (viz. Obr. 37).

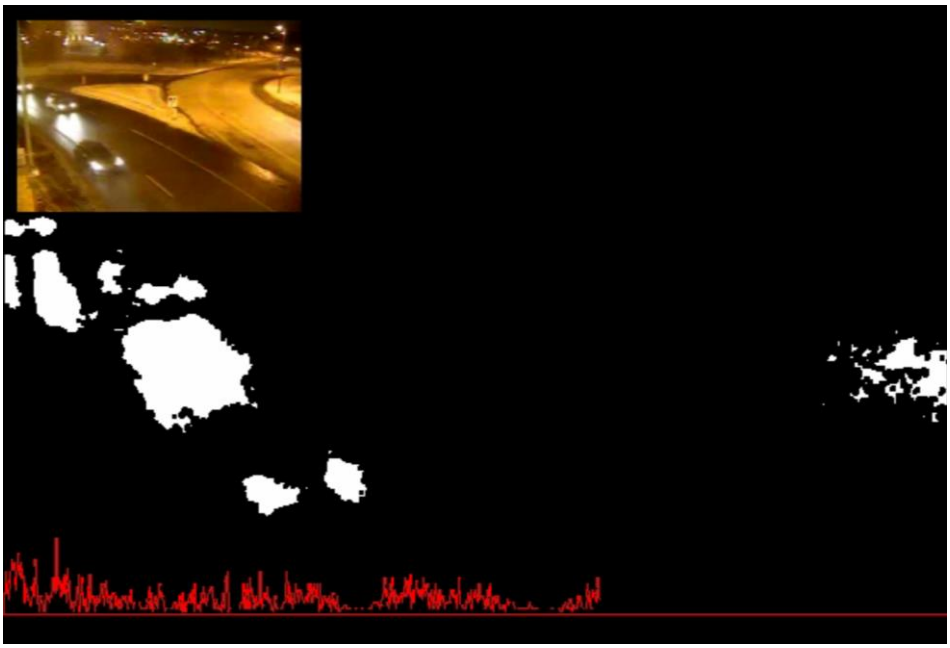
Obr. 37: Odstranění pohybu způsobeného fontánou a detekce automobilu v pozadí

Jelikož byla citlivost detekce pohybu nastavena na optimální hodnotu použitelnou pro všechna videa, extrémní případy mohou způsobit velmi nepřesný výstup programu (Obr. 38). Pokud je plocha listů příliš velká, je třeba změnit parametry redukce šumu na větší hodnoty, což by pak způsobilo odstranění téměř veškerého pohybu a nic by nemuselo být detekováno. To by se dalo vyřešit nastavením vyšší citlivosti, což je řešení pouze tohoto konkrétního případu, nebo použitím vhodnější metody pro detekci pohybu.



Obr. 38: Chybná detekce pohybu listů a tím pádem zkreslený výstup programu

Poslední testovanou kategorií byla noční videa. Podobně jako u videí za špatného počasí, má špatná viditelnost velký vliv na výstup programu. Jelikož však reflektory automobilů tuto špatnou viditelnost znatelně narušují, rozpoznání pohybu je velmi přesné a efektivní. Problém ovšem může nastat, pokud světla automobilů oslní celou scénu. Tuto situaci program vyhodnotí jako rychlou změnu světelných podmínek a nějakou dobu přestane detekovat pohyb. Tomu se dá zabránit vypnutím této možnosti, ale jelikož byl program navržen pro detekci pohybu v interiéru, nebylo třeba tuto možnost vypínat pro účely testování.



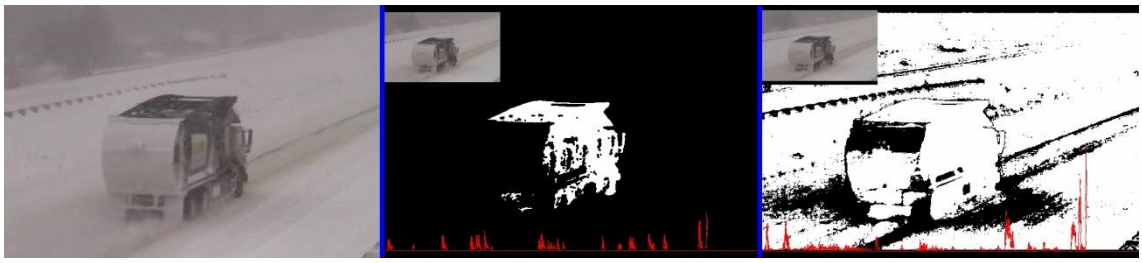
Obr. 39: Výstup programu při zpracování nočního videa

Na úplný závěr byly všechny testy přesnosti provedeny ještě jednou a to s využitím metody odečítání pozadí, která je součástí knihovny OpenCV, a následně porovnány s výsledky testů této práce. Konkrétně byla využita metoda využívající směsi gaussovských distribucí (dále jen SGD). Tato metoda vybírá vhodný počet gaussovských distribucí pro každý pixel (ostatní metody používají *K* počet distribucí). Velmi dobře se přizpůsobuje měnící se scéně, která je způsobena světelnými změnami atd. Tato metoda nebyla nijak dále programově upravována, pouze byla nastavena citlivost detekce pohybu na stejnou hodnotu jako u algoritmu ViBe.

SDG byla schopna velmi přesně detekovat pohybující se objekty (i ty vzdálenější) a pokud pohyb přestal, objekty se nepropadaly do pozadí tak rychle jako u upraveného algoritmu ViBe, který byl naimplementován v praktické části této práce. SDG byla ovšem při detekci velmi citlivá a velmi často chybně detekovala pohyb, který ani nenastal.

Obr. 40: Detekce vzdálených objektů za špatného počasí - vlevo originalní snímek, uprostřed výstup ViBe, vpravo výstup SDG

Při detekci vzdálených objektů za špatného počasí vrací SDG velmi přesné výsledky narozdíl od upraveného algoritmu ViBe. Pokud se ovšem na scéně objeví větší objekt, výstup SDG je velmi nepřesný (viz. Obr. 41).

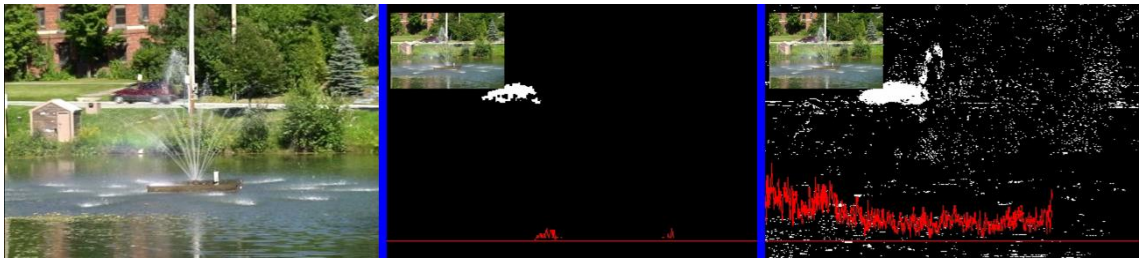


Obr. 41: Detekce blízkých objektů za špatného počasí – vlevo originalní snímek, uprostřed výstup ViBe, vpravo výstup SDG

V ostatních případech poskytovala upravená verze ViBe algoritmu lepší výsledky. Ačkoliv algoritmus nedokáže velmi přesně detekovat tvar pohybujících se objektů, je schopen velmi dobře rozlišit popředí od pozadí a tím pádem efektivně rozpoznat pohyb.

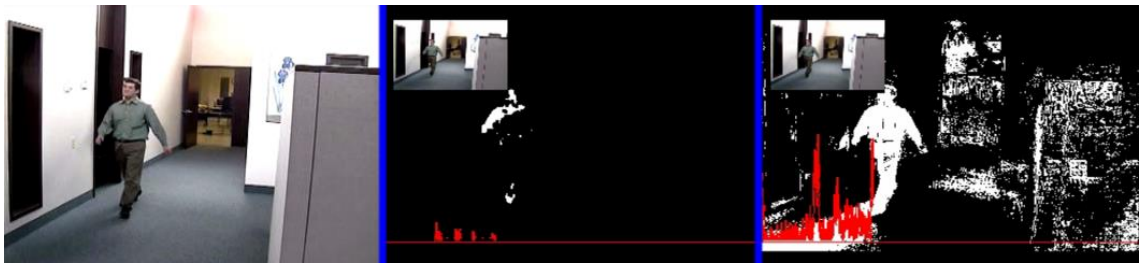
Obr. 42: Detekce objektů s dynamickým pozadím – vlevo originalní snímek, uprostřed výstup ViBe, vpravo výstup SDG

Z Obr. 42 je zřejmé, že SDG velmi přesně detekovala pohybující se kánoe, ale po celý zbytek videa nesprávně rozpoznávala vodní hladinu jako pohyb. Upravený ViBe nedokázal rozeznat přesný tvar pohybující se kanoe, ale dokázal odstranit pohyb způsobený vodní hladinou a rozpoznal pohyb přesně tam, kde nastal.



Obr. 43: Detekce objektů s nežádoucím pohybem na popředí – vlevo originalní snímek, uprostřed výstup ViBe, vpravo výstup SDG

U videí z interiéru má lepší výsledky algoritmus ViBe, který přesně detekuje pohyb a ignoruje rychlou změnu světelných podmínek. SDG má problémy s jakoukoliv sebemenší změnou scény a tím pádem poskytuje nepřesné výsledky.



Obr. 44: Detekce objektů v interiéru – vlevo originalní snímek, uprostřed výstup ViBe, vpravo výstup SDG

4.3 Návrh a implementace demonstrační aplikace

Cílem demonstrační aplikace je aplikovat algoritmus detekce pohybu pro praktické využití. Jedno z hlavních odvětví využití metody odečítání pozadí je hlídání pozemků pomocí kamerových systémů. Takový bezpečnostní systém si ovšem každý nemůže dovolit a většinou je vyžadován neustálý dohled nad tímto systémem. Proto bylo navrženo levné a jednoduché řešení pro hlídání interiéru a to pomocí připojené kamery nebo integrované webkamery v počítači nebo notebooku.

Hlavní předpoklad pro správný běh aplikace je připojení počítače k síti. Aplikace pomocí kamery monitoruje okolní prostředí a při rozpoznání pohybu okamžitě informuje uživatele. Největší výhoda je tedy okamžité upozornění uživatele, který se může v daný moment nacházet kdekoliv (avšak musí být připojen k síti). Z toho vyplývá, že aplikaci je nejvhodnější použít například při trávení dovolené daleko od domova.

Bezpečnost a hlídání pozemků ovšem není jediné využití. Lze ji využít i jako notifikační aplikaci. Uživatel tak může mít přehled o své domácnosti nebo o čase příchodů ostatních členů domácnosti.

Při návrhu aplikace byl kladen důraz na tři hlavní požadavky:

1. Jednoduchost
2. Srozumitelnost
3. Minimální uživatelská interakce

Nejlepší způsob jak splnit všechny tři požadavky je vytvořit aplikaci bez uživatelského rozhranní, kde jediná interakce s uživatelem je spuštění programu. Interakce s uživatelem je v tomto případě nesmyslná, nevíme jestli uživatel bude přítomen u počítače v době běhu aplikace a tvorba aplikace, která by ovládala systém na dálku by bylo časově i programově náročné. Jako uživatelské rozhranní byla tedy zvolena konzolová aplikace.

Dalším bodem programu je zpracování obrazu, které již bylo naimplementováno v praktické části této práce.

Nejdůležitější částí je upozornění uživatele při detekci pohybu. Uživatel může v daný moment používat jakékoliv zařízení připojené k síti. Pokud by však upozornění přišlo na zařízení, ke kterému by neměl přístup, nemusel by mít dostatek času na reakci.

Nejjednodušší a nejuniverzálnější řešení je použití emailové komunikace. Dnes má již téměř každé zařízení, které se dá připojit k síti, předinstalovaného poštovního klienta a pokud ne, tak do schránky lze také přistoupit pomocí internetového prohlížeče.

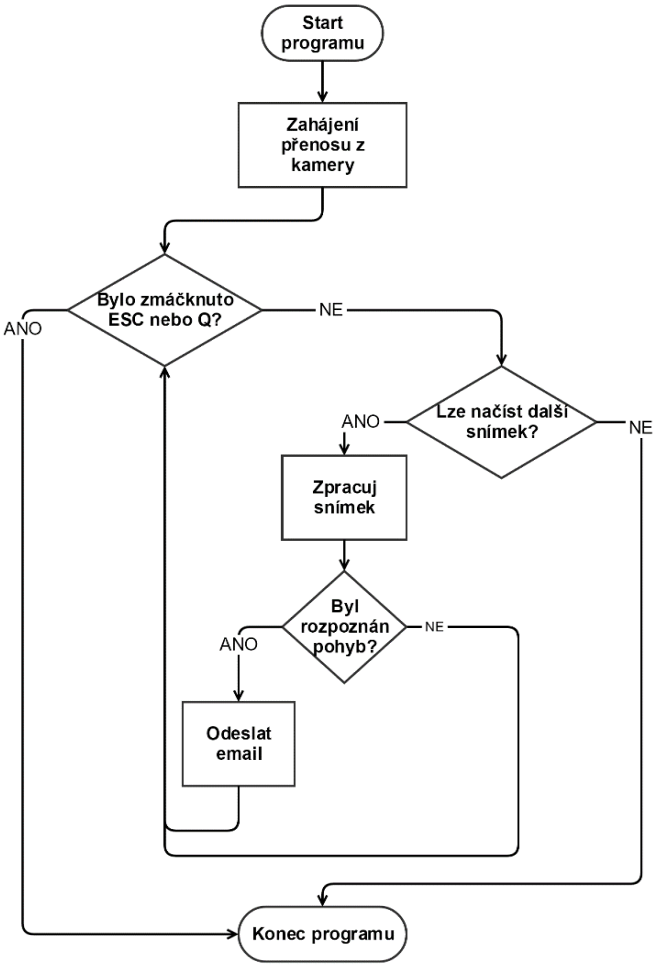
Odeslání emailu je velmi jednoduché, je zdarma, zprávu lze odeslat více lidem, druhá strana zprávu obdrží téměř okamžitě a lze k ní také připojit soubory nebo obrázky. Odeslání elektronické pošty je programově jednoduchá operace a proto byla zvolena jako řešení problému upozornění uživatele.

Zaslání pouze textové zprávy prostřednictvím elektronické komunikace je ovšem nedostačující. Ve většině případů je třeba zjistit, co problém způsobilo nebo ověřit, zda-li se nejedná o „falešný poplach“.

Existují dvě možnosti jak k danému problému přistupovat:

1. Odeslat v příloze video, na kterém je detekován pohyb
2. Odeslat v příloze několik snímků, na kterých je detekován pohyb

Elektronický klient umožňuje zasílání videí v příloze zprávy, ale velikost takového videa může narůst až do řádu stovek MB, což může ve výsledku zpomalit nebo kompletně zrušit odesílání zprávy. Tento přístup je také náročnější na výpočetní výkon počítače a může velmi rychle zahltit vnitřní paměť počítače. Naopak uložení jen několika snímků je rychlé a nenáročné na paměť (snímky mají velikost v řádu stovek KB). Problém tohoto přístupu je správně zvolit počet snímků, aby bylo odeslání zprávy rychlé a přesto se dalo rozeznat příčinu pohybu.

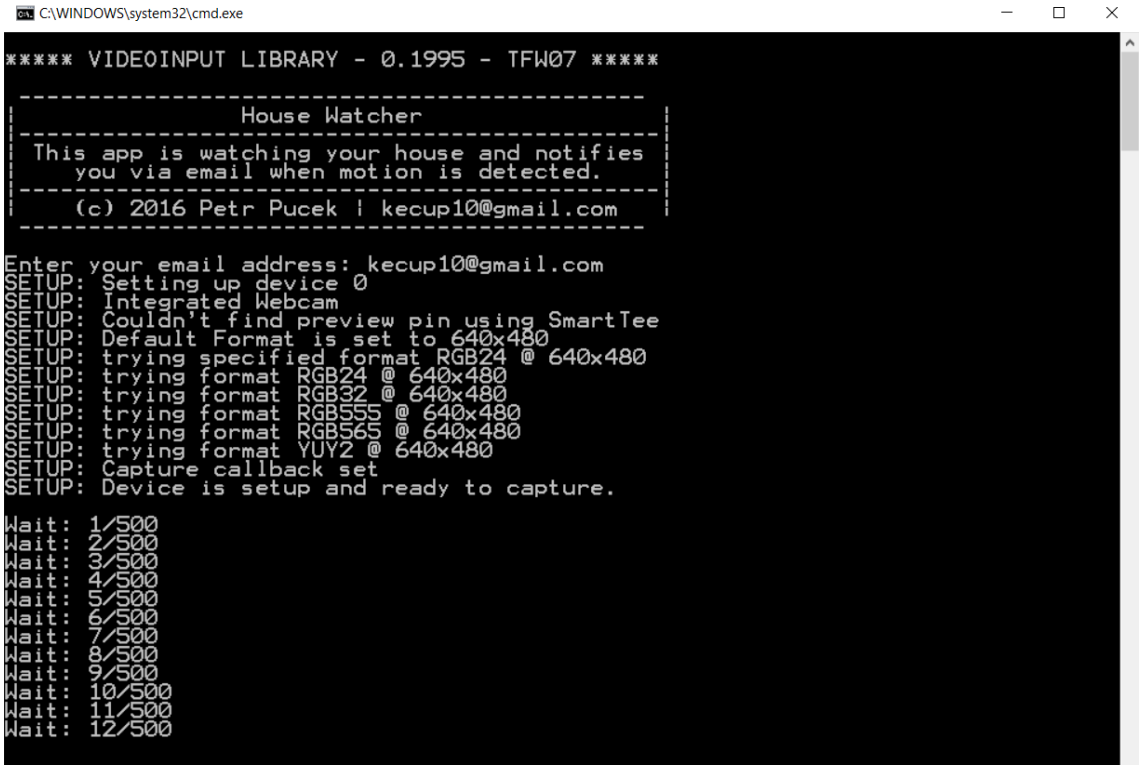


Obr. 45: Vývojový diagram demonstrační aplikace

Struktura aplikace vychází z programu jehož návrh je popsán v této kapitole. Obsahuje 3 hlavní části: inicializace proměnných a zahájení živého přenosu, zpracování snímku a odesílání emailu. K implementaci byly použity všechny nástroje popsané v této kapitole.

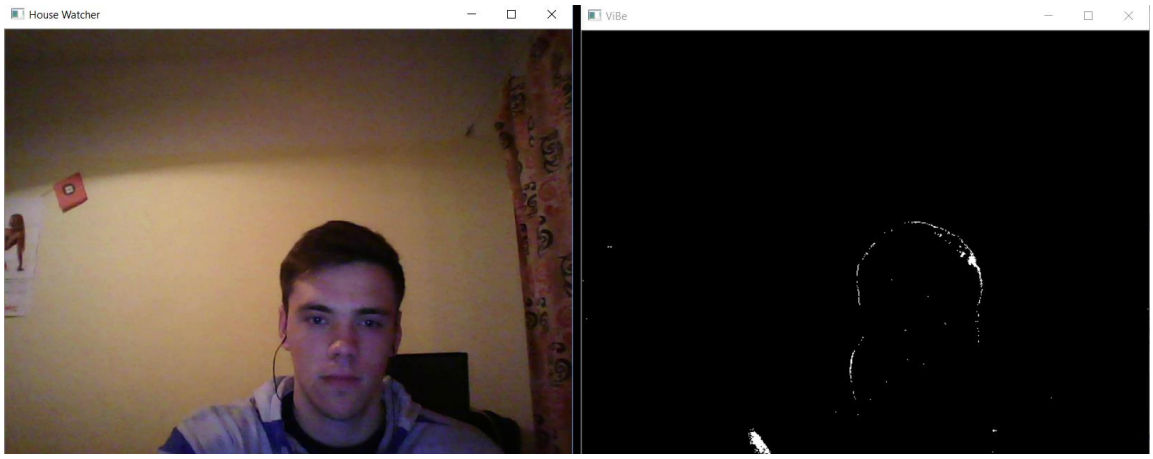
Obr. 46: Výstup aplikace okamžitě po spuštění – zadání emailové adresy je jediná interakce s uživatelem

Jakmile uživatel potvrdí zadání korektní emailové adresy, aplikace aktivuje připojenou kameru a přejde do režimu čekání. Tímto dává uživateli prostor k opuštění monitorovaného prostoru aniž by byl systémem detekován a následně emailem upozorněn. Čekací doba je nastavena na 500 snímků, což je přibližně 20s.



Obr. 47: Aktivace připojené kamery a přechod aplikace do režimu čekání

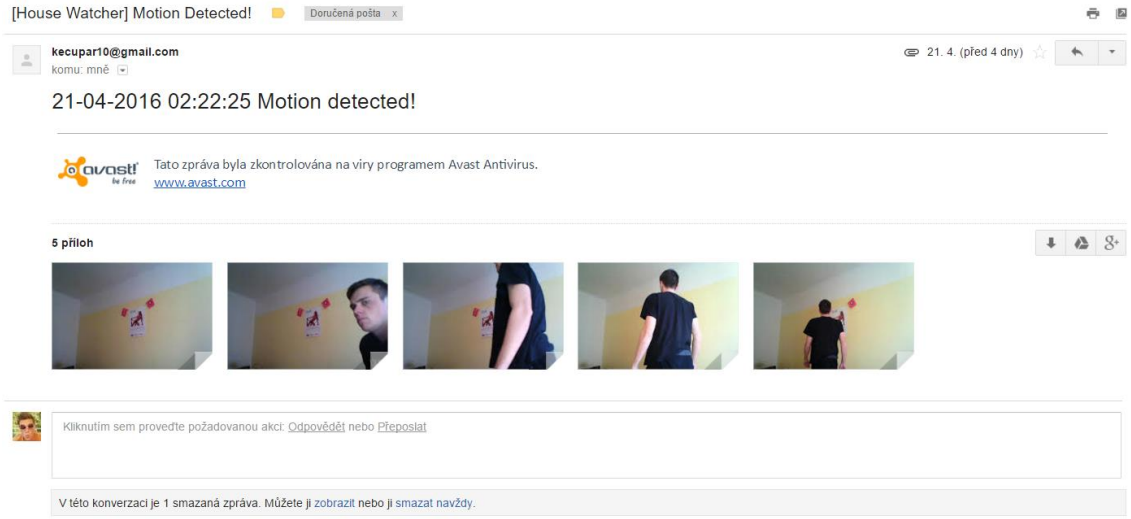
Po vypršení čekací doby aplikace přechází do režimu aktivního monitorování. Okamžitě jsou zobrazena 2 okna: výstup webkamery a výstup algoritmu ViBe, který je volán se stejnými parametry jako v praktické části (*R=20, noMin=1*).



Obr. 48: Výstup aplikace v režimu aktivního monitorování

Jakmile je detekován pohyb (plocha kontur pohybujících se objektů je větší než dvacetina výšky obrazu), aplikace si uloží aktuální čas a přepne se z režimu

monitorování do režimu zápisu. Režim zápisu ignoruje veškerý pohyb, pouze si ukládá do pole každý desátý snímek z kamery. První snímek v poli je jediný snímek před detekcí pohybu, další čtyři snímky obsahují pohyb. Všechny tyto snímky jsou také uloženy fyzicky do vnitřní paměti a to do složky images, která je vytvořena v adresáři, kde se nachází aplikace.



Obr. 49: Formát emailového upozornění

Pokud je všech pět snímků uloženo, aplikace přechází k odesílání emailu. O odesílání se stará komponenta EASendMail postavená na SMTP protokolu, která je ke stažení na www.emailarchitect.net. Pro účely demonstrační aplikace byla stažena zkušební verze. Aplikace předá emailovou adresu adresáta, zprávu obsahující časové razítko detekce pohybu a cestu k uloženým snímkům funkci, která se stará o odesílání emailu. V této funkci jsou nastaveny parametry elektronické komunikace jako je nastavení adresy a hesla odesílatele, nastavení adresy adresáta, nastavení předmětu zprávy, obsahu zpávy, přidání příloh ke zprávě, nastavení SMTP serveru a port. Jakmile jsou všechny tyto parametry nastaveny, aplikace daný email odešle do schránky adresáta a opět přechází do režimu aktivního monitorování.

Do demonstrační aplikace nebyla implementována technika ignorování rychlé změny světelných podmínek. Cílem aplikace je uživatele upozornit, zda byl rozpoznán pohyb v interiéru. Pokud uživatel nechá dům prázdný delší dobu a někdo uvnitř rozsvítí, měl by být okamžitě informován, jelikož se může jednat o podezření o vloupání. Zda-li je toto podezření pravdivé, se dozví v dalších emailových zprávách, ve kterých už pravděpodobně bude pachatel zachycen. Jestliže tuto změnu bude mít na svědomí počasí, což je málo pravděpodobné v interiéru, další zpráva již nepřijde.

5 Závěr

Práce podrobně rozebírá jednu z metod detekce pohybu a to odečítání pozadí. Na úvod byla nastudována jedna z problematik počítačového vidění a to analýza a detekce pohybu. Výzkum byl zaměřen na metodu odečítání pozadí, konkrétně byla vybrána relativně nová technika, která se nazývá Visual Background Extractor. Tato technika je díky nízké výpočetní náročnosti velmi rychlá a dokáže zpracovávat video i rychleji než v reálném čase.

Tato technika byla implementována do programu, který pomocí kamery připojené k počítači pořizoval videa při každém rozpoznání pohybu. Tento algoritmus byl dále upraven a přizpůsoben pro detekci pohybu v interiéru, ve kterém poskytuje velmi kvalitní výsledky. Byl rozšířen o možnost detekce a následné ignorace rychlých světelných podmínek jako je například rozsvícení světla, což umožňuje jeho použití i za špatných světelných podmínek. Dále byl jeho výstup upraven, aby byl uživatel schopen vidět všechen pohyb a to pomocí vykreslení pohybu do grafu a zakreslení originálního snímku do rohu výstupního videa pro jednodušší kontrolu správnosti výstupu.

Pro účely testování byla pořízena datová sada s různými druhy a intenzitami pohybu. Bylo využito dvou datasetů z internetové databáze určené pro algoritmy detekce pohybu, byla získána sada videí z výzkumných kamer Fakulty informačních technologií VUT v Brně a poslední sada videí byla pořízena z domácího prostředí digitálním fotoaparátem.

Testování přineslo zajímavé výsledky. V interiéru byl program velmi přesný, téměř bezchybný. Jediný nedostatek je rychlé propadávání objektů do pozadí, což však úzce souvisí s detekcí rychlé změny světelných podmínek. Jelikož byl program vyvíjen pro interiér, výsledky testů ve venkovních videích přinesly překvapivé výsledky. S parametry nastavenými pro interiér si dokázal poradit i s dynamickým prostředím, ale také v některých případech poskytl nepoužitelný výstup.

Dále byla tato technika porovnána s její alternativou a to metodou využívající směsi gaussovských distribucí. Tato alternativa byla přesnější v detekci hran objektů, avšak byla velmi citlivá na jakoukoliv sebemenší změnu scény. Upravený ViBe byl přesnější v detekci pohybu i přesto že nedokázal přesně detekovat plochu pohybujicích se objektů.

Tento částečně upravený algoritmus byl dále naimplementován do demonstrační aplikace, která slouží jako jednoduchý bezpečnostní systém pro hlídání soukromých prostor bez nutnosti jakékoliv uživatelské interakce. Aplikace monitoruje prostory přes připojené kamerové zařízení a komunikuje s uživatelem přes síť a to pomocí elektronické pošty. Uživatel je tedy okamžitě informován o jakémkoliv pohybu.

Další vývoj by mohl být zaměřen na inteligentní rozpoznání dobrých či špatných světelných podmínek a na základě této informace automaticky měnit citlivost detekce pohybu či aplikovat redukci šumu.

Jelikož jsou dnes chytré telefony každodenní součástí našich životů, další krok by mohl směrovat k vývoji mobilní aplikace, která by nahradila upozorňování uživatele přes elektronickou poštu.

Literatura

[1] Collins, R. T., Lipton, A. J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt1, P., Wixson, L. *A System for Video Surveillance and Monitoring*. The Robotics Institute. Carnegie Mellon University, Pittsburgh PA 1 The Sarnoff Corporation, Princeton, NJ. 2000

[2] Haritaoglu, I., Harwood, D., Davis, L. S. *W^4: Real-time surveillance of people and their activities*. IEEE Transactions on Pattern Analysis and Machine Inteligence. 22(8): 809-830. 2000

[3] Wren, C., Azarbayejani, A., Darrell, T., Pentland, A. *Pfinder: Real-time tracking of the human body*. IEEE Transactions on Pattern Analysis and Machine Inteligence. 19(7): 780-785. 1997

[4] Stauffer, C., Grimson, E. *Adaptive background models for real-time tracking.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 246-252. 6 1999

[5] Elgammal, A., Duraiswami, R., Harwood, D., Davis, L. S. *Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance.* Proceedings of the IEEE. Vol. 90, No. 7, 1151-1163. July 2002

[6] Li, L., Luo, R. *Context-controlled adaptive background subtraction.* Proceedings of IEEE Int Workshop on Performance Evaluation of Tracking and Surveillance. 31-38. 2006

[7] Barnich, O., Droogenbroeck, M. V. *Vibe: a powerful random technique to estimate the background in video sequences*. Proceedings of ICASSP 2009, Taipei: IEEE Computer Society Press. 945-948. 2009