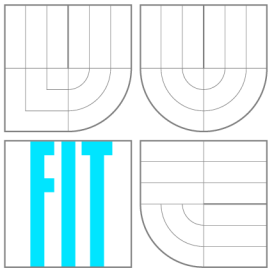


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# SLEDOVÁNÍ HLÍDANÉHO PROSTORU A DETEKCE NARUŠENÍ BEZPEČNOSTI KAMEROVÝM SYSTÉMEM

PERIMETER MONITORING AND INTRUSION DETECTION BASED ON CAMERA SURVEILLANCE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ GOLDMANN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. FILIP ORSÁG, Ph.D.

BRNO 2013

## Abstrakt

Tato práce obsahuje popis základního systému používaných pro sledování hlídaného prostoru. Ve stěžejní části práce jsem představil metody počítačového vidění vhodné pro detekci a klasifikaci objektů. Dále jsem na základě metody odečítání pozadí realizoval algoritmus pro detekci lidí, který využívá pro popis objektů histogram orientovaných gradientů a pro klasifikaci SVM klasifikátor. V poslední části práce se zabývám porovnáním deskriptoru založeného na histogramu orientovaných gradientů se SIFT deskriptory a vyhodnocením preciznosti detekčního algoritmu.

## Abstract

This bachelor thesis contains a description of the basic system for perimeter monitoring. The main part of the thesis introduces the methods of computer vision suitable for detection and classification of objects. Furthermore, I devised an algorithm based on background subtraction which uses a Histogram of Oriented Gradients for description of objects and an SVM classifier for their classification. The final part of the thesis consists of a comparison of the descriptor based on the Histogram of Oriented Gradients and the SIFT descriptor and an evaluation of precision of the detection algorithm.

## Klíčová slova

detekce lidí, extrakce příznaků, Histogram gradientů, Algoritmus podpurných vektorů, Ada-Boost, SIFT, SURF, PIR

## Keywords

human detection, feature extraction, Histogram of Gradient, Support Vector Machine, Ada-Boost, SIFT, SURF, PIR

## Citace

Tomáš Goldmann: Sledování hlídaného prostoru a detekce narušení bezpečnosti kamerovým systémem, bakalářská práce, Brno, FIT VUT v Brně, 2013

# Sledování hlídaného prostoru a detekce narušení bezpečnosti kamerovým systémem

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Filipa Orsága Ph.D..

.....  
Tomáš Goldmann  
15. května 2013

## Poděkování

Děkuji panu Ing. Filipu Orságovi Ph.D. za kvalitní odborné konzultace a zároveň i všem dalším, kteří mě během práce jakkoliv podpořili.

© Tomáš Goldmann, 2013.

*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

<b>1 Úvod</b>	<b>1</b>
<b>2 Analýza existujících řešení</b>	<b>3</b>
2.1 Detekce pohybu pomocí PIR čidel	3
2.2 Duální senzory	4
2.3 Infračervené závory	5
2.4 Analýza dat z termovize	5
2.5 Detekce objektů analýzou obrazu	6
<b>3 Počítačové vidění</b>	<b>8</b>
3.1 Získávání obrazových dat	9
3.1.1 Kamera	9
3.2 Segmentace objektů	9
3.2.1 Metoda odčítání pozadí	10
3.2.2 Segmentace pomocí Graph-cut	11
3.2.3 Získávání oblastí jednotlivých objektů	12
3.2.4 Sliding Window	12
3.3 Extrakce charakteristických rysů	13
3.3.1 Local Binary Pattern (LBP)	13
3.3.2 Scale-invariant feature transform (SIFT)	14
3.3.3 Speeded Up Robust Features (SURF)	15
3.3.4 Histogram of Oriented Gradients (HOG)	16
3.3.5 Porovnání deskriptorů	18
3.4 Klasifikace a rozpoznávání objektů	18
3.4.1 Algoritmu AdaBoost	19
3.4.2 Algoritmus SVM (Support Vector Machine)	20
3.5 Detektory	24
3.5.1 Viola-Jones detektor	24
3.5.2 Detektor s SVM algoritmem	25
3.5.3 Shrnutí	25
<b>4 Implementace algoritmu</b>	<b>26</b>
4.1 Segmentace objektů	26
4.2 Klasifikace objektů	27
4.3 Deskriptory založené na SIFT	28
4.4 GUI	29
4.4.1 Prostředí	29
4.4.2 Získávání obrazu prostřednictvím vlákna	29



4.5	Shrnutí . . . . .	30
<b>5</b>	<b>Testování</b>	<b>31</b>
5.1	Detekce na videu s výskytem osob . . . . .	32
5.2	Detekce na videu bez výskytu osoby . . . . .	34
5.3	Souhrn . . . . .	34
<b>6</b>	<b>Závěr</b>	<b>35</b>

# Kapitola 1

## Úvod

Čas od času se vyskytne potřeba sledovat hlídáný prostor, může se například jednat o prodejnu v nákupním středisku nebo i o oblast v areálu armádního prostoru. Škála typů objektů popř. oblastí, kde vzniká potřeba provádět sledování nebo jen hlídání, je široká a pestrá. Cílem sledování je získat přehled o tom, co se v takovém prostoru odehrává. Obvykle se vyhodnocuje, zda byl sledovaný prostor narušen, v pozitivním případě může být žádoucí provést identifikaci narušitele.

Tato práce se zabývá analýzou existujících řešení pro hlídání sledovaného prostoru, dále pak návrhem, implementací a testováním algoritmů pro detekci narušitele sledovaného prostoru pomocí analýzy videodat získaných ze stacionární kamery. Řešení je realizováno prostřednictvím metod z oboru počítačového vidění. Pomocí široké škály technik lze analyzovat obrazová data a získávat z nich informace o tom, jaké objekty se pravděpodobně ve scéně nacházejí. Algoritmus navržený v této práci umožňuje ve video záznamu detekovat objekty, které poté pomocí binárního klasifikátoru klasifikuje a vyhodnocuje, zda je klasifikovaným objektem člověk nebo ne. Jednou z nezbytných vlastností algoritmu je možnost detekce narušitele v reálném čase. Mezi další důležité vlastnosti patří invariantost vůči změnám snímané oblasti. Výsledek této práce je určen pro zabezpečovací a sledovací systémy, které využívají pro získávání obrazových dat stacionární CCTV kameru. Dále jsou v této práci porovnány tři metody popisu objektů pro jejich klasifikaci.

Do sledovacích systémů můžeme zahrnout trackovací systémy, kamerové systémy nebo dokonce i nejrůznější systémy pro sledování síťové komunikace (internetu). Kamerové sledovací systémy (označované jako CCTV) slouží k vizuálnímu sledování prostoru. V prvopočátku se tyto systémy používaly nejvíce v privátním sektoru. V posledních dvou desetiletích dochází k velké expanzi těchto kamer do veřejného sektoru a to především z důvodů eliminace kriminality a zvyšování veřejné bezpečnosti. Na trhu je možné nalézt několik provedení těchto kamer např. kamera ve tvaru pouličního osvětlení. Výstup z těchto kamer by měl sloužit jako případný důkazní materiál pro dokázání trestného činu. S nástupem výkonnějších hardwarových prostředků dochází k vývoji algoritmů, které jsou schopné detekovat osoby nebo obličeje osob ve sledovací oblasti kamery. Na základě tohoto pokroku došlo k rozšíření využití kamerových systémů v zabezpečovací technice, kde se ve většině případů používá kombinace se senzorem nebo jinou technologií. Například existují kamery s PIR senzorem pro vizuální potvrzení poplachu. Zpracování takto získaných dat a jejich uchování, se musí řídit dle platných právních norem, které se v jednotlivých státech mohou od sebe radikálně lišit. Rozšiřování sledovacích systémů, především pak těch, ve kterých dochází k identifikaci osob, vyvolává kontroverze z důvodu narušování soukromí a možnosti zneužití takto získaných dat.

Práce je rozdělená do tří větších tematických celků, tvořených jednou nebo i více kapitolami. První z nich popisuje existující používané technologie pro detekci a sledování pohybu (senzory, infrakamery, . . .) v zabezpečovací technice, jedná se především o technologie určené pro detekci pohybu objektu narušujícího hlídaný prostor. Další celek se zabývá principy počítačového vidění, metodám detekce a klasifikace objektů, především pak klasifikačních a extrakčních algoritmů. Poslední část této práce popisuje implementovaný program a vyhodnocuje, na základě provedeného testování, vlastnosti algoritmu.

## Kapitola 2

# Analýza existujících řešení

Cílem této kapitoly je seznámit se s řešeními, která jsou dostupná na trhu pro detekci narušení sledovaného prostoru. Některé uvedené řešení umožňují určit, zda případným narušitelem je osoba. Algoritmus pro sledování hlídaného prostoru s možností detekce výskytu člověka je zároveň předmětem této práce, jedná se o oblast, která se využívá především v oblasti zabezpečovací techniky nebo průmyslu. Většina uvedených řešení je určena jenom pro vyhodnocení, zda byl hlídaný prostor narušen nebo ne. Rozpoznávání typu narušitele není v dnešní době příliš rozšířeno. Za jeden z důvodů minoritního rozšíření se dá považovat to, že se jedná o relativně novou oblast. První prakticky použitelný detektor pro detekci objektů v reálném čase Viola-Jones byl publikován v roce 2002. Do té doby byly algoritmy časově velmi náročné s tím, že výkon dostupného hardwaru byl ve srovnání s výkonem v dnešní době značně malý.

### 2.1 Detekce pohybu pomocí PIR čidel

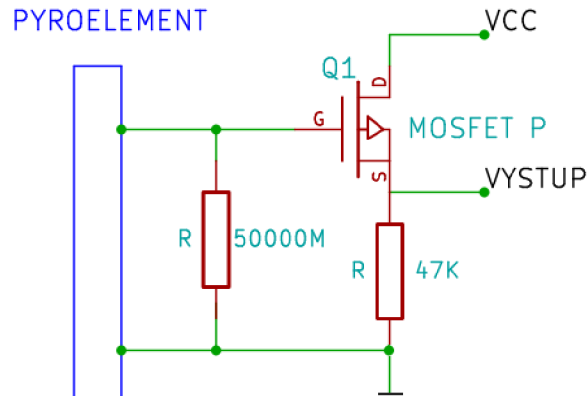
Jedná se o aktivní detektor pohybu pracující na principu pyroelektrického jevu. Na základě tohoto jevu se při vniknutí nevyžádané osoby do sledovaného prostoru, na pyroelektrickém elementu uvnitř senzoru generuje elektrický náboj, přímo závislý na vyzařovaném infračerveném záření prostoru a narušitele. Osoba ve sledované poli senzoru vyzařuje určité infračervené záření, které tepelně ovlivňuje jednu část senzoru, přičemž druhá část senzoru přijímá teplotu okolí. Na základě pyroelektrického jevu vznikne na čidle časově proměnný náboj. Tento náboj je přiveden na rezistor s vysokým odporem (z důvodů, že náboj má jednotky pA), následně je elektrické napětí na rezistoru zkoumáno pomocí napěťového sledovače. Schéma jednoduchého napěťového sledovače s MOSFET tranzistorem je znázorněno na obrázku 2.1.

Jednotlivé typy PIR senzoru se liší především sledovací oblastí a sledovacím úhlem, který může být až 360 stupňů (dosaženo kombinací více PIR senzorů). Maximální dosah PIR čidel bývá kolem 60 metrů.

Nevýhodou těchto senzorů je především zhoršující se možnost detekce s narůstající vzdáleností narušitele od senzoru. Mezi další nevýhody patří ovlivnitelnost teplotou okolí nebo falešný poplach při proudění teplého vzduchu. První verze těchto senzorů pozitivně reagovaly i na přímé sluneční světlo, nynější generace používá filtry, které vliv přímého slunečního světla nebo jiného stacionárního zdroje infračerveného záření eliminují [4]. V praxi se tento senzor používá jako spínač pro zabezpečovací systémy např. u vchodových dveří nebo jako schodišťový spínač.

Rozdělení PIR senzorů dle typů sledovaných oblastí [10]:

- Typ "vějíř"- Horizontální snímání oblasti v širokém krátkém výseku
- Typ "záclona"- Vertikální snímání oblasti v širokém krátkém výseku
- Typ "dlouhý dosah"- Horizontální snímání oblasti v úzkém dlouhém výseku



Obrázek 2.1: Napěťový sledovač PIR senzoru

### Komerční řešení

Na základě počtu firem zabývajících se vývojem a výrobou těchto senzorů, lze tvrdit, že se jedná o poměrně rozšířený typ bezpečnostních senzorů. Podle způsobu připojení můžeme moduly s PIR senzorem rozdělit na drátové a bezdrátové. Bezdrátové jsou zejména vhodné do objektů, kde doposud nebyl instalován žádný bezpečnostní systém. Výrobou PIR senzoru se zabývá například tuzemská firma EVOLE, která se zabývá výrobou bezdrátových provedení těchto snímačů. Další tuzemskou firmou je firma Elko, zabývající se mimo jiné vývojem a výrobou modulů pro inteligentní domy, v jejímž produktovém sortimentu lze rovněž nalézt několik provedení PIR senzorů. Většina nabízených produktů dokáže nereagovat na malé živočichy (myši, potkany, atd.). Firma Jablotron vyrábí PIR senzory s kamerou pro vizuální potvrzení poplachu. V případě, že PIR senzor detekuje pohyb, kamera pořídí sekvenci snímku, ty jsou následně uloženy do interní paměti detektoru a přenášeny do ústředny. Kamera je vybavena bleskem a infračerveným přisvícením pro focení v noci.

## 2.2 Duální senzory

Jelikož se PIR senzory nehodí na aplikace, kde je větší pravděpodobnost vzniku falešných poplachů, vznikly tzv. duální senzory. Tyto senzory pracují se dvěma senzory s rozdílnými typy detekce. Ve většině případů se jedná o pyroelektrický senzor (viz 2.1) a mikrovlnný senzor popř. ultrazvukový senzor, který pracuje na stejném principu. Mikrovlnný senzor se skládá z přijímače a vysílače. Vysílač vyšle rádiovou vlnu o frekvenci  $f_{tx}$ , která se odrazí od objektu a přijímač ji následně přijme. Po té se vyhodnotí, jak se frekvence změnila (rozdíl  $f_{tx}$  a  $f_{rx}$ ). Princip je založený na základě Dopplerova jevu, popisující změnu frekvence odražené rádiové vlny od pohybujícího se tělesa.

$$f_{rx} = f_{tx} \left(1 + \frac{2v_0}{c}\right) \quad (2.1)$$

kde :  $c$  – rychlost světla,  $v_0$  – rychlost pohybujícího se tělesa

Naměřená frekvence přijímačem [23]

Aby bylo možné tyto senzory používat, musí být střežený prostor uzavřen. Pokud je vyžadováno použití více senzorů v jedné místnosti, musí se při instalaci provést synchronizace, aby bylo vyloučeno vzájemné ovlivňování.

### Komerční řešení

Výrobou duálních senzorů se zabývá polská firma Satel. Produkt APMD-150 obsahuje dva pyroelementy a jeden mikrovlnný senzor, detektor je velice citlivý a zároveň imunní vůči falešným poplachům. Detektor má také prostředky pro kompenzaci vlivu tepla, čímž se zvýší pracovní teplotní rozsah. Z českých firem tyto senzory vyrábí světoznámá firma Jablotron.

## 2.3 Infračervené závory

Infračervené závory neboli světelné závory detekují přerušení světelného paprsku. Jedna světelná závora tudíž není schopná pokrýt široký rozsah, jak výše zmíněné PIR nebo duální senzory. V případě použití světelných závor pracujících s více paprsky ve venkovním prostředí, musí být rozsah a tvar paprsku zvolen tak, aby v případě přeletu ptactva nebo padajícího listí nedošlo k falešnému poplachu. Existují dva druhy infračervených závor, jeden z nich má oddělený přijímač a vysílač. Druhý z typů obsahuje přijímač a vysílač v jednom zařízení, přičemž paprsky vysílá proti odrazové ploše. Maximální vzdálenost mezi vysílačem a přijímačem bývá několik desítek metrů.

### Komerční řešení

Infračervené závory jsou stejně jako PIR senzory velice dobře dostupné na tuzemském i zahraničním trhu. Jedním z výrobců je firma Optex, která vybírá celou škálu infrazávor pro rozsahy od 20 m až 340 m. Firma své infrazávory postavila na dvou pulsních infračervených paprscích.

## 2.4 Analýza dat z termovize

Touto metodou se pomalu dostáváme do oblasti analýzy obrazu, ve které určujeme, kdo vnikl do sledovaného prostoru. V případě této metody se určuje, zda jde o potencionálně živý objekt nebo neživý, rozhodování probíhá na základě obrazové analýzy snímku zachycujícího tepelné záření. Pravděpodobnost správné detekce živého objektu je relativně vysoká, jelikož teplotní kontrast živého objektu vůči prostředí je daleko větší než barevný kontrast ze standardních RGB kamer. Z každého povrchu tělesa se uvolňuje tzv. infračervená radiace, kterou je možné zachytit, aniž by byl senzor v přímém kontaktu s tělesem. Obvyklá délka vlny, kterou snímač zachycuje je  $2 - 5 \mu\text{m}$ . Tímto způsobem lze zjistit, jakou teplotu má těleso.

### **Komerční řešení**

Kamery pracující na principu zachytávání tepelného záření vyrábí americká firma FLIR. Jeden z vyráběných typů kamer je SR-124, dokáže zachytit radiozitu objektu až do rozsahu 205 m. V porovnání s předchozími možnostmi je řešení s použitím termovize značně nákladné a používá se především pro sledování perimetru na rozsáhlých územích (nejčastěji kolem armádních oblastí).

## **2.5 Detekce objektů analýzou obrazu**

Nyní se dostáváme k metodě, kterou řeší i tato práce. Cílem této metody je pokusit se vyhodnocovat objekty stejným způsobem, jakým to provádí lidský mozek, tzn. vizuálním zhodnocením. V dnešní době se tato technologie postupně rozšiřuje, na vývoji se podílejí například firmy IBM a Microsoft. Využití této detekce nachází uplatnění v armádě, zabezpečovací technice a automobilovém průmyslu. Potenciál na rozšíření této technologie pro běžné komerční využití je u posledně zmíněného oboru, jelikož do některých typů aut se pomalu začínají instalovat systémy pro detekci chodce před vozidlem. Toto vyhodnocování probíhá právě na základě analýzy obrazu. Za nevýhodu této metody lze považovat potřebu výkoného hardwaru, protože na každý snímek videosekvence musí být aplikovaný algoritmus (mnohdy několik algoritmů). Jedno z řešení může být použití FPGA obvodu pro provádění výpočetně náročných částí společně s procesorem, který vykonává hlavní program. Při použití počítače platformy (x86, x64, atd.), jsou možnosti pro provádění výpočtu daleko větší. Do budoucna se dá předpokládat, že se tato metoda rozšíří na prostorovou detekci objektu pomocí počítačového vidění. Například použitím několika jádrového procesoru nebo akcelerací některých výpočtu přes GPU (Graphic Processor Unit). Tato práce obsahuje popis řešení na počítači architektury (x86, x64) s výpočty prováděnými standardně přes CPU.

S detekcí pomocí analýzy obrazu se možná nevědomky už někteří obyvatelé setkali, poněvadž se tato metoda používá pro automatickou identifikaci obličejů z dat některých veřejných bezpečnostních kamer (v ČR se identifikace zatím neprovádí).

V komerčním sektoru se zatím analýza obrazu používá relativně málo, plně automatizované vyhodnocování zda se jedná o osobu nebo jiný objekt narušující prostor se pravděpodobně neprovádí. Existují však systémy s enkodéry umožňující získávat informace pro operátory, kteří sledují prostřednictvím bezpečnostních kamer dané prostory o tom, že došlo k narušení prostoru nespécifikovaným objektem. Vyhodnocování o jakého narušitele se jedná, provádí operátor.

### **Komerční řešení**

Systém pro sledování narušení hlídaného prostoru analýzou obrazu prodává firma logicBRICKS, která vyrábí kity logiPDET <sup>1</sup> založená na FPGA čipu od firmy Xilinx. Tento kit obsahuje i systém pro detekci výjezdu auta z jízdního pruhu. Produktů a informací ohledně bezpečnostních systémů s využitím detekce pomocí analýzy obrazu je v síti internet relativně malý počet.

Pro online identifikaci osob z obrazu získaného pomocí CCTV existuje systém secu-eFACE [14] od firmy VÍTKOVICE IT SOLUTIONS a.s.. Celé řešení je postavené na několika biometrických algoritmech, data z kamer jsou zpracovávána na počítačích architektury x86 popř. x64. On-line analýza snímaného obrazu realizuje rozpoznání osoby v obraze, kde zpracovaná data ve formě výřezu (tokenu) obličejů jsou porovnávána metodou 1:N. Následně

<sup>1</sup><http://www.logicbricks.com/Products/logiPDET.aspx>

akce systému (např. přenos alarmu konkrétnímu bezpečnostnímu dohledu) jsou zcela parametrizovatelné podle konkrétního případu užití. Firma pro tento rozpoznávací systém dále poskytuje otevřená API rozhraní a SDK.

Komerční řešení, které se alespoň trochu blíží tématu této práce, vyrábí firma BOSCH. Dnes se již jedná o systém ve své čtvrté verzi s názvem Bosch Intelligent Video Analysis (IVA), jedná se o bezpečnostní asistent pro venkovní i vnitřní použití umožňující detekci pohybujících se objektů s potlačením falešných poplachů způsobenými rušivými zdroji v obraze. Celé řešení je postavené tak, že po zaznamenání obrazu je provedena analýza obrazu z níž se získají tzv. metadata. Proces analýzy obrazu obsahuje algoritmy pro detekci zmizení statického objektu z prostoru, vniknutí/opuštění objektu z prostoru, detekci jakou rychlostí se objekt pohybuje, jakým směrem a zda se nemění jeho velikost popř. o kolik se změnila a mnoho dalších podobně informativních algoritmů. Systém dále umožňuje provádět trackování se záznamem trajektorie pohybujícího se objektu a detekci hlavy. Metadata se jednak uchovávají a zároveň slouží operátorovi, jako informace na jaký prostor se má aktuálně zaměřit.

Konfigurace celého systému probíhá prostřednictvím filtrů, jejichž nastavením se určuje, jaké dynamické objekty bude systém brát jako relevantní a jaké bude ignorovat. Mezi vlastností pro konfiguraci patří nastavení minimální velikosti objektu, velikost, rychlost a směr.

Prostřednictvím systému VCA (Video Content Analysis) dokáže rozhodnout, jaké video bude nahráno. Jedná se o tzv. poplachová videa a jenom ty při použití tohoto systému budou vybrána pro záznam nebo streaming. O tom jaké video se označí jako poplachové neboli kritické, rozhodují uživatelem definované podmínky. Při vyhodnocování systém zjistí, zda metadata splňují zadané kritéria a video označí nebo neoznačí jako poplachové.

Cílem celého systému je ulehčit práci operátorovi, zefektivnit ji a zamezit přehlédnutí kritických situací. Firma BOSCH uvádí, že operátor během 20 minut přehledne 90 % aktivit ve sledované oblasti. S pomocí tohoto systému lze tuto nepříliš příznivou statistiku vylepšit.



## Kapitola 3

# Počítačové vidění

Obecně můžeme o počítačovém vidění tvrdit, že se snažíme napodobit lidské vidění prostřednictvím technických prostředků. Když se podíváme podrobněji, co počítačové vidění vlastně je, zjistíme, že se jedná o získávání, zpracování, analýzu a porozumění obrazovým datům. Existuje několik zdrojů, ze kterých lze získávat obrazová data, nejčastěji se používají data získaná z kamer. V oboru lékařství se lze setkat s obrazovými daty získanými z magnetické rezonance (MRI) nebo počítačové tomografie (CT). Z těchto dat můžeme získat symbolickou (pro podporu rozhodování) nebo číselnou informaci. Nabytá data bývají obvykle zpracována pomocí znalostních systémů a technik umělé inteligence. Mezi hlavní úkoly počítačového vidění patří rozpoznávání, nejčastěji se řeší problém rozpoznání objektů, identifikace a detekce. Dalším úkolem může být například detekce pohybu ve scéně, konkrétně pak tracking objektu. Obor počítačového vidění prochází v poslední době neustálým rozvojem, k tomuto jevu přispěla dostupnost dostatečného výpočetního výkonu a dostupnost mnoha knihoven např. knihovna OpenCV, která obsahuje několik stovek optimalizovaných algoritmů. V praxi se dále používají matematické programovací jazyky, jakým je například komerční MATLAB.

Postup provádění počítačového vidění může být u různých aplikací odlišný, nejčastěji je ovšem tvořen těmito částmi:

- Získání obrazových dat
- Předzpracování obrazových dat - Cílem této části je odstranění šumu a vyfiltrování obrazu, tak aby byl co nejlépe připraven pro další zpracování.
- Detekce/segmentace objektů - Segmentování obrazových dat na data reprezentující jednotlivé objekty.
- Extrakce charakteristických rysů - Slouží pro vytvoření struktury, která v dalších krocích reprezentuje objekt.
- Zpracování dat - Data se zpracovávají podle aplikace, v případě, že chceme provést klasifikaci objektů, zjistíme, jak je objekt podobný se svým vzorem, nejčastěji pomocí klasifikátoru natrénovaného na množině dat.
- Provedení rozhodnutí - Tato část se používá v případě, že je potřeba provádět rozhodnutí např. zda objekt odpovídá nebo neodpovídá vyhledávanému objektu. Provádí se na získaných datech z rozpoznávacího algoritmu.

## 3.1 Získávání obrazových dat

Pro získání dat se nejčastěji používají obrazové senzory, které zaznamenávají pro jednotlivé obrazové body (pixely) hodnoty jasu. Některé typy senzorů ovšem dokážou místo jasu zachytit jinou fyzikální veličinu (propustnost, hloubku, elektromagnetické vlnění atd.).

### 3.1.1 Kamera

Mezi hlavní typy kamer patří vidicon, CCD a CMOS kamera [17]. Nejstarší typy kamer využívali k zachycování obrazu takzvanou vidicon trubici, fungující na principu zachycení dopadajícího světla na fotocitlivou vrstvu, které je následně „naskenováno“ pomocí elektronového paprsku. Tato analogová technologie je v dnešní době nahrazená CCD a CMOS digitálními senzory. Technologi CCD a CMOS používají pro zachycení jednotlivých bodů pole senzorů, které je možné rozlišit podle toho, jestli jsou postavené na pasivní technologii nebo aktivní (nejpoužívanější). První uvedenou technologií (CCD) vynalezli pánové W. Boyle a George E. Smith v roce 1969. Samotný snímač funguje na základě fotoefektu, jevu, při kterém foton nárazem do atomu dokáže dostat elektron do vyšší energetické hladiny. Tyto elektrony jsou pak přitahovány k elektrodě, která je ve spodní části čipu, přičemž v polovodiči vznikají tzv. díry. Za těchto okolností vznikne elektrický náboj o velikosti závislé na množství dopadnutého světla. Konstrukce čipu je tvořena sérií posuvných registrů, kde elektrické náboje z jedné snímací řady se postupně pomocí synchronizačních signálů přesouvají směrem k zesilovači.

CMOS senzory se liší od CCD senzorů technologií sběru elektrického náboje, většina dnes vyráběných aktivních čipů má u každého obrazového bodu tranzistor pro zesílení elektrického signálu. Vzniklý elektrický signál je odveden do paměťové matice čipu. Hlavní výhodou CMOS technologie je možnost využití podobné technologie jako pro výrobu mikroprocesoru, čímž se snižuje výrobní cena senzoru. Protože v části jednotlivých obrazových bodů je umístěn zesilovací tranzistor, je plocha pro dopadající světlo zmenšena, proto mají tyto senzory horší světelnou citlivost. V poslední době se použitím nejrůznějších technologií daří tento nedostatek eliminovat, v budoucnu pravděpodobně CMOS z větší části CCD nahradí. V praxi jsou tyto čipy daleko více komplikovanější, např. obsahují u jednotlivých obrazových bodů miniaturní čočky nebo mají ve struktuře přidáné další elektrody nebo dokonce další aktivní elektronické součástky.

Pro bezpečnost kamery se využívají CMOS i CCD senzory s rozlišením obvykle do 2 MPix. Výstup kamer je obvykle ukládán pomocí videorekordéru do datového úložiště, na trhu lze však najít i kamery, které ukládají videozáznam na paměťovou kartu umístěnou přímo na kameře. Jestliže je zapotřebí sledovat, zda do sledovaného prostoru nevniknul narušitel, musí se video zpracovávat v reálném čase.

## 3.2 Segmentace objektů

Segmentace obrazu tvoří první krok pro detekci objektu ve videu nebo obrazu. Cílem je nalézt tzv. oblast zájmu, ve které se nachází objekt, se kterým chceme dále pracovat (Region Of Interest - ROI). V případě této práce je cílem nalézt objekt, který narušil hlídaný prostor a rozpoznat, zda se jedná o člověka.

### 3.2.1 Metoda odčítání pozadí

Odčítání pozadí je jeden z procesů počítačového vidění pro segmentaci objektů umístěných v popředí scény. Základní metoda vychází z toho, že máme zachycené pozadí scény, od kterého odčítáme aktuálně získaný snímek. Odčítají se vždy jednotlivé pixely a do nového snímku se ukládá rozdíl, který je větší než prahová hodnota. Prahování musí být nastaveno tak, aby byly správně zachyceny objekty, nikoliv však minoritní změny ve scéně. Další možností je využití algoritmu, který na základě vstupních dat nastavuje hodnotu prahování automaticky (automatické prahování). Výsledkem je pozitivní maska vymezující jednotlivé objekty v obraze. Aby bylo možné tuto metodu prakticky použít je zapotřebí nějakým způsobem reagovat na změny intenzity světla ve scéně, pohyby kamery (byť jen vibrace) a změnu geometrie pozadí např. přeparkování aut.

$$|frame_i - frame_{i-1}| > Th \quad (3.1)$$

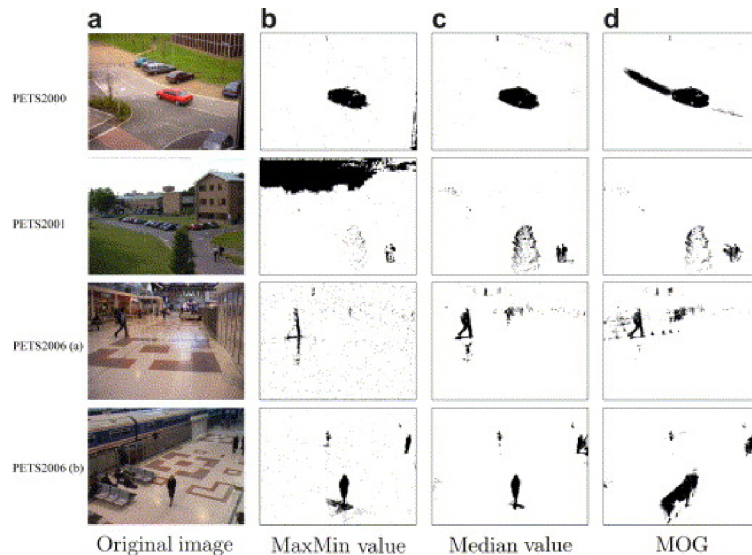
Obecný vzorec základní metody odčítání pozadí

Jednou z takových pokročilých metod je metoda využívající tzv. medián filtr. Tento filtr pracuje s více obrazy pozadí, které jsou uloženy v paměti. Pro každý pixel se provede výpočet mediánu z pixelů, který leží na stejném místě ve snímcích ve vyrovnávací paměti. Výhodou tohoto filtru je, že odstraňuje šum z pozadí. Jako nevýhodu lze považovat větší paměťovou náročnost, která je přímo úměrná počtu snímků. Mezi další metody patří metoda s využitím Kalmanova filtru (využívá Gaussova šumu) nebo metoda kombinující směs Gausiánů (Mixture of Gaussians - MoG).

$$B(x, y, t) = median\{I(x, y, t - i)\} \quad (3.2)$$

$$|I(x, y, t) - median\{I(x, y, t - i)\}| > Th \text{ kde } i \in \{0, n - 1\} \quad (3.3)$$

Metoda využívající tzv. Medián filtr [24]



Obrázek 3.1: Na tomto obrázku je vizuální porovnání metod odčítání pozadí [11].

---

**Algoritmus 1:** Algoritmus pro odčítání pozadí

---

Získání dat pozadí

Detekce popředí

Validace dat

**if** *nastal čas obnovy pozadí a nebyl detekován žádný objekt* **then**

| vrať se na bod č.1

**end**Vrať masku

---

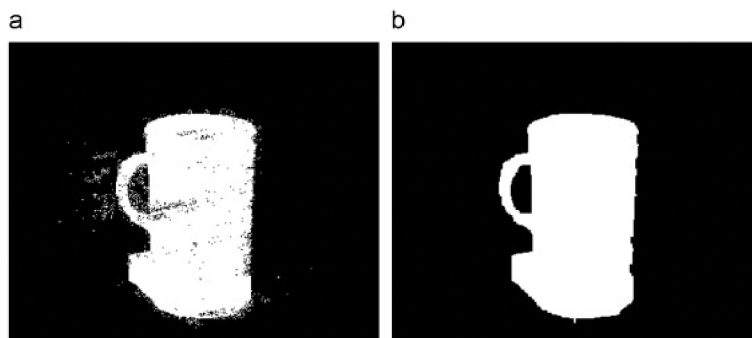
Jelikož je často v oblasti masky, která vymezuje objekt na popředí, několik míst ve kterých je oblast nespojitá, je možné tyto defekty odstranit pomocí morfologického filtru dilatace/eroze, tak aby vznikla spojitá oblast vymezující objekt. Filtr eroze ořezává okraje masky, zatímco filtr dilatace rozšiřuje. Použití nebo nepoužití tohoto filtru závisí jakým způsobem je nastavený algoritmus pro označování spojitých oblastí.

$$A \oplus B = \bigcup_{b \in B} A_b \quad (3.4)$$

Morfologická operace - dilatace

$$A \ominus B = \{z \in E | B_z \subseteq A\} \quad (3.5)$$

Morfologická operace - eroze



Obrázek 3.2: a) před aplikací morfologického filtru, b) po aplikaci morfologického filtru.

**Vlastnosti metody**

Hlavní výhodou této metody je rychlost zpracování a jednoduchost na implementaci. Nevýhodou může být horší kvalita získané masky, kterou ovlivňuje nastavená prahovací hodnota. Tato metoda je i přes tyto nedostatky velice vhodná pro zpracování videa v reálném čase. Především je tato metoda rozšířená pro detekci pohybujících se objektů na videu pořízeného statickou kamerou.

**3.2.2 Segmentace pomocí Graph-cut**

Tato metoda umožňuje rozdělit obraz do dvou segmentů pozadí a objekt. Princip spočívá v tom, že každý pixel obrazu značí jeden uzel grafu, který je k sousedním pixelům připojen hranami. Každá hrana má svou cenu, která odpovídá rozdílu kapacit intenzity uzlů. Zdrojový

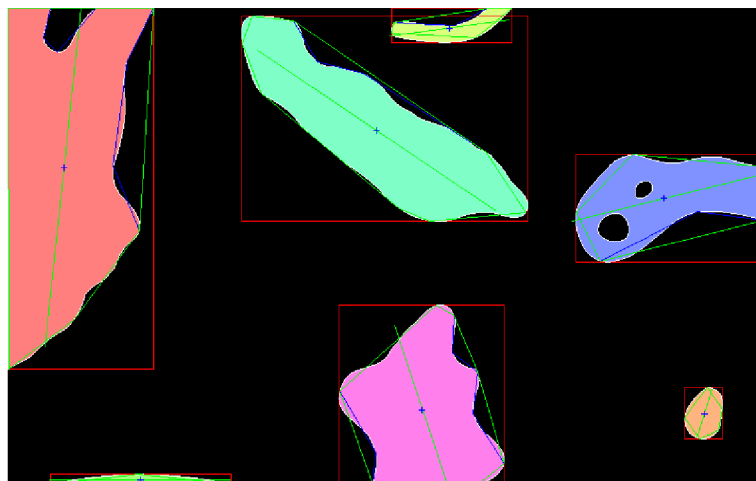
uzel je spojený s pixely objektů na popředí. Na tento graf se aplikuje metoda maximálního toku/minimálního řezu. Výsledkem je množina uzlů definující objekt nebo pozadí, přičemž musí být nasycená kapacita minimálního řezu.

### Vlastnosti metody

Kvalita výsledné masky je přímo závislá na intenzitě přechodu jasů jednotlivých částí objektů na popředí, v případě těles, které mají hodně odlišný jas oproti pozadí a minimum přechodu jasů v oblasti objektu je získaná maska velice kvalitní. Nevýhodou tohoto algoritmu je výpočetní náročnost.

### 3.2.3 Získávání oblastí jednotlivých objektů

Získaná maska výše uvedenou metodou, vymezuje jednotlivé objekty na popředí. Aby bylo možné s jednotlivými objekty pracovat, je zapotřebí popsat oblasti, kde se jednotlivé objekty nacházejí a vymezit je (popř. je možné provést vyznačení kontur). Toto řeší algoritmus pro označování spojitých oblastí vycházející z matematické oblasti teorie grafů. Jedna z variant algoritmu je implementována v knihovně OpenCV. V programu se aplikuje pomocí metody `cvContour`, ke které je dostupná metoda `cvBoundingRect` pro vymezení oblasti objektu (vrací strukturu popisující obdélník). Další možností je použití doplňkové knihovny k OpenCV `cvBlob` [15], hlavní její předností je optimalizovanější algoritmus a širší výběr možností.



Obrázek 3.3: Vymezování oblastí jednotlivých objektů (červeně), vykreslení kontur (modře), vykreslení obálky objektu (konkávní, konvexní) (zeleně) [15]

### 3.2.4 Sliding Window

V případě, že se nedá získat pozadí scény a jednotlivé objekty nelze ani pomocí jiných metod rozlišit, používá se tato metoda. Na rozdíl od předchozích metod se nesnaží o separaci objektů a vyznačení jejich oblastí. Idea metody je taková, že se podle vstupních parametrů vytvoří patřičně velké okno, které se postupně posouvá a vyhodnocuje se, zda se v získaném podokně nachází vyhledávaný objekt, pokud se pomocí klasifikátoru určí, že se jedná o rozpoznávaný objekt, vrátí se toto okno jako hranice objektu. V některých implementa-

cích se po každém průchodu okno zmenší (omezeno parametrem minimálního okna). Tuto metodu využívají detektory Schneiderman-Kanade, Viola Jones a mnoho dalších.

### Vlastnosti metody

Obrovskou výhodou této metody je její univerzálnost, protože funguje se statickými i dynamickými snímky. Tato výhoda je ovšem vykoupena tím, že se pro každé okénko musí vypočítat příznaky a provést klasifikace. Z tohoto faktu vyplývá i vyšší pravděpodobnost nežádané pozitivní klasifikace.

## 3.3 Extrakce charakteristických rysů

Lidský mozek dokáže vnímat spoustu faktorů, mezi ně například patří barva, tvar, struktura povrchu, hloubkový vjem apod... Ovšem lze těžko určit jakým způsobem mozek s těmito informacemi pracuje a nakládá, i když moderní technologie jako pozitronová emisní tomografie nebo impedanční tomografie můžou při průzkum této oblasti pomoci [17]. Z tohoto důvodu existují pouze psychologické modely popisující, jak lidský mozek funguje. Jednou dobře známou funkcí lidského vizuálního systému je to, že využívá hrany a hranice objektů.

I přesto, že by existovala studie detailně popisující, jakým způsobem lidský mozek zpracovává popř. rozpoznává objekty, pravděpodobně by z důvodu extrémní výpočetní náročnosti nebylo možné tento systém výpočetně realizovat. Proto se prostřednictvím algoritmů z obrazových dat získávají jednoduchá data, která slouží pro popis objektu.

Proces probíhá tak, že se s pomocí některého algoritmu pro extrakci z obrazových dat získají určité prvky popisující objekt. Ve většině případů se jedná o body nebo konkrétní prvky (hrany, rohy, oblasti objektu, ...). Pro získávání hran, rohů nebo oblastí se používají nízko úrovněvé algoritmy zpracování obrazu. Cílem tohoto procesu je získat deskriptor<sup>1</sup> obrazových dat.

### 3.3.1 Local Binary Pattern (LBP)

Local Binary Pattern (LBP) [19] operátor je obrazový operátor pro získání deskriptoru obrazových dat. Z dat získaných po aplikaci tohoto operátoru se obvykle vytváří histogram popisující obrazová data. První verze tohoto operátoru byla popsána v roce 1996 [12]. V dnešní době existuje již několik modifikací tohoto operátoru lišících se např. ohodnocovací funkcí.

Základní idea originální verze tohoto operátoru je založena na popisu obrazových bodů prahováním sousedních pixelů proti aktuálně zpracovávanému pixelu, výsledkem je binární vzor reprezentovaný binárním číslem. V originální verzi se pro tento účel používal blok  $3 \times 3$  pixely, přičemž okrajové pixely byly prahovány vůči hodnotě prostředního pixelu. Hodnota získaného vzoru je u této metody rovna kombinaci osmi binárních čísel, proto je možné dosáhnout  $2^8$  kombinací. Tento algoritmus předpokládá, že obrázek se kterým se pracuje je převeden do odstínů šedi. Na základě každého lokálního binárního vzoru u pixelu, je možné pixel označit (podle typu v obrázku), jako potencionální plochu, hranu, roh nebo konec čáry.

V roce 2001 publikovali Timo Ojala, Matti Pietikäinen, Topi Mäenpää [18] rozšířený LBP operátor o možnost použití okolí odlišných rozměrů. Ve své verzi používali radiální okolí s pomocí bilineární interpolace neceločíselných souřadnic, přičemž mohli používat libovolný

---

<sup>1</sup>struktura popisující data



rádus a počet sousedních pixelů. Pro popis jednotlivých variant vznikla (P, R) notace, kde P znamená počet pixelů v okolí a R znamená vzdálenost od středového pixelu (rádus). Pro tento rozšířený LBP operátor platí tato ohodnocovací funkce:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (3.6)$$

$$s(x) = \begin{cases} 1 & \text{pokud } x \geq 0 \\ 0 & \text{jinak} \end{cases} \quad (3.7)$$

Pro praktické použití je zapotřebí rozdělit obrázek na buňky, každou buňku bude ve výsledném deskriptoru reprezentovat jeden histogram. Stejně nebo podobné rozdělení bude použito i u ostatních deskriptorů.

### Vlastnosti

Mezi základní vlastnosti LBP patří při použití jistých úprav rotační invariantnost, efektivní diskriminace rozdílných typů textur, odolnost vůči změnám osvětlení. Další důležitou vlastností je možnost rychlého výpočtu, která umožňuje provádět analýzu obrazu v reálném čase. Používá se například pro rozpoznávání obličejů, otisků prstů nebo pro rozlišování struktury povrchu.

### 3.3.2 Scale-invariant feature transform (SIFT)

Jedná se o algoritmus pro extrakci obrazových deskriptorů klíčových bodů [22]. Mezi základní vlastnosti patří invariantnost vůči změně měřítka nebo nasvícení. Cílem algoritmu je nalézt stabilní body na obrázku, jako jsou například rohy nebo skvrny a vypočítat jejich deskriptor. Ten je tvořen histogramy gradientů v okolí jednotlivých bodů.

Algoritmus je možné rozdělit do pěti částí:

- Vytvoření scale space
- LoG aproximace
- Nalezení klíčových bodů
- Přiřazení orientace klíčovým bodům
- Extrakce deskriptorů

Jak už bylo výše uvedeno, obrazová data obsahují vysokofrekvenční složky se kterými nechceme dále pracovat. Cestou je postupně aplikovat filtr Gaussovského rozostření, tím dojde k postupnému odstraňování detailů z obrazu, pokud se každý mezikrok zachová, vznikne sekvence snímků. Tato metoda vychází z tzv. teorie Scale-space umožňující reprezentovat signál ve více měřítcích, s rostoucím scale jsou v obraze potlačovány vysokofrekvenční složky. Odfiltrováním vysokofrekvenčních složek můžeme efektivně pracovat s nízkofrekvenčními složkami. V případě extraktoru SIFT se vytváří Scale-space pyramida, každá sekvence přitom tvoří jednu sekvenci snímků s postupným aplikovaným gaussovským filtrem. Další sekvence vznikne zmenšením snímku na polovinu a opětovnou aplikaci filtru.

V dalším kroku se vezmou z každé oktávy dva sousedící obrazy a provede se jejich rozdíl, jelikož byl na snímky aplikovaný Gaussovský filtr, získá se rozdíl dvou Gausiánů (DoG <sup>2</sup>). Tímto způsobem lze získat ekvivalentní výsledek, jaký by vznikl při použití LoG <sup>3</sup> filtru.

<sup>2</sup>Difference of Gaussians

<sup>3</sup>Laplacian of Gaussian

Přímá aplikace tohoto filtru se nepoužívá z důvodu výpočetní náročnosti druhých derivací, proto se výpočet provádí prostřednictvím rozdílu Gaussiánu.

Klíčové body jsou takové, které tvoří lokální maxima nebo minima v DoG snímku. Musí se však vyloučit body, které leží na hranách nebo mají nízký kontrast. Aby bylo možné dosáhnout rotační invariance jednotlivých klíčových bodů, musí se určit jejich orientace. Idea je taková, že pro každý klíčový bod je vytvořen histogram gradientů (každý sloupec reprezentuje  $10^\circ$  z  $360^\circ$ ) v jeho okolí. Z histogramu se vybere sloupec s nejvyšší hodnotou a podle něho se přiřadí orientace. Ze sloupců majících 80 % z nejvyššího sloupce se vytvoří nový klíčový bod s orientací původního sloupce.

Posledním krokem je vytvořit deskriptor klíčových bodů. Nejčastěji je používaný deskriptor histogramů gradientů v okolí klíčového bodu. Okolí bodů se rozdělí na  $16 \times 16$  bodů, přičemž klíčový bod bude mít polohu uprostřed, tato síť se dále rozdělí velikost na  $4 \times 4$ . V každé takto vzniklé buňce se vytvoří histogram gradientů o osmi sloupcích. Výsledný vektor popisující jeden klíčový bod tudíž bude mít  $4 \times 4 \times 8 = 128$  rozměrů. Pro zvýšení robustnosti popisovače se provádí váhování jednotlivých gradientů, aby ty co jsou vzdálenější od klíčového bodu méně přispívaly do histogramu a naopak, ty co jsou blíže, měly větší váhu.

### Vlastnosti

Algoritmus se především používá pro vyhledávání objektů v obraze, rozpoznávání gest v obličeji nebo vytváření fotografických panoramat snímků. Mezi základní nevýhody algoritmu patří výpočetní náročnost. SIFT algoritmus umožňuje vytvořit rotačně invariantní deskriptor, který je nezávislý na osvětlení.

### 3.3.3 Speeded Up Robust Features (SURF)

Jedná se o poměrně nový algoritmus (poprvé publikován v roce 2006 [2]) pro získávání klíčových bodů, částečně je inspirován SIFT algoritmem, ale na rozdíl od něj se při výpočtu nevytváří pyramida škálovaných obrazů s aplikovaným gaussovským filtrem. Scale-space se vytváří pomocí změny rozměru Gaussovského jádra, protože zvětšení Gaussovského filtru vyjadřující jednu z derivací je ekvivalentní výpočtu derivace na úměrně zmenšeném obraze. Gaussův filtr se aplikuje na integrální obraz, doba tohoto výpočtu je nezávislá na rozměrech Gaussovského jádra. Tento přístup ve výpočtu Scale-space vedl ke značnému zrychlení. Pro popis klíčových bodů ve výsledném deskriptoru se používá součet odezev Haarova filtru v okolí bodu.

K detekci klíčových bodů se využívá výpočet determinantu Hessovou maticí, která je tvořena pomocí druhých parciálních derivací skalární funkce. Pro praktický výpočet jsou hodnoty funkce  $f(x, y)$  nahrazeny hodnotami jasů obrazových bodů a ze druhé parciální derivace se dosadí hodnota pixelu vzniklá konvolucí původního obrazu s vhodným Gaussovským jádrem. Po získání matice s Hessovými hodnotami je možné provést lokalizaci klíčových bodů. Pomocí prahování se nejdříve odstraní nejméně významné body, po té se naleznou lokální maxima a určí se pozice maxim do subpixelové přesnosti. Pro konstrukci deskriptoru je okolí klíčových bodů rozděleno na 16 čtverců, přičemž každý čtverec je popsán vektorem o čtyřech položkách. Výsledný vektor pro jeden klíčový bod má 64 rozměrů.



$$H(x, \sigma) = \begin{pmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{pmatrix} \quad (3.8)$$

Hessova matice

### Vlastnosti

Algoritmus pro výpočet tohoto deskriptoru má oproti SIFT algoritmu výhodu především v rychlosti, protože konvoluce Gaussovského jádra probíhá s integrálním obrazem. Dále je invariantní vůči změnám rozměru a je možné nalézt implementace s rotační invariancí (původní verze U-SURF není rotačně invariantní).

### 3.3.4 Histogram of Oriented Gradients (HOG)

Základní idea deskriptoru histogramu orientovaných gradientů je založená na vektorovém popisu objektu pomocí distribuce směru hran do sloupců histogramu. Jednotlivé vektory popisují orientaci hran objektu nebo rozložení intenzity gradientů, kombinace jejich histogramů tvoří samotný deskriptor. Tento deskriptor představili pánové Navneed Dalal a Bill Triggs v roce 2005 [6], kteří vyzkoušeli několik typů tohoto popisovače s různými metodami výpočtu gradientů a normalizace.

Samotné provádění detekce začíná tím, že se vypočítají pomocí níže uvedených matic aplikovaných na obrazová data derivace ve směru x a y, tato operace probíhá s obrázkem nebo snímkem převedeného do odstínů šedi. Následně se pro každý pixel provede výpočet gradientů. Dalal a Trigs vyzkoušeli výpočet gradientů provést pomocí aplikace Sobelovy (velikost  $3 \times 3$ ) nebo diagonální masky, experimentálně ovšem zjistili, že prostřednictvím předchozího výpočtu je dosaženo lepších výsledků.

$$D_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (3.9)$$

$$D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (3.10)$$

Matice pro výpočet derivací ve směru x a y

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$$D_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.12)$$

Sobelův filtr  $3 \times 3$

Orientace gradientů se určí ze vztahu  $\theta = \arctan \frac{I_y}{I_x}$ , kde  $I_x = ID_x$  a  $I_y = ID_y$  [1], I znamená obraz a  $D_{x,y}$  reprezentuje konvoluční matici.

V rámci experimentování s tímto algoritmem bylo zjištěno, že při detekci osob, je dosaženo lepších výsledků, když se před aplikací derivační matice, aplikoval na obraz Gaussův filtr. Vychází to především z toho, že Gaussův filtr rozostření dokáže zamaskovat některé méně podstatné detaily při ponechání detailů podstatnějších.

Dalším krokem pro získání deskriptoru je rozdělení matice gradientů (velikosti ekvivalentní rozměru snímku) do buněk. Velikosti gradientů obsažených v jednotlivých buňkách se rozdělí do několika tříd (nejčastěji do 9), z kterých se následně vytvoří histogram pro každou buňku. Pro barevné RGB obrázky je zapotřebí vypočítat HOG pro každý kanál, pro každou buňku se ze tří kanálů vybere největší hodnota histogramu a ta se zvolí pro popis. V tomto případě bude výsledný deskriptor obsahovat histogramy zkombinované ze tří kanálů.

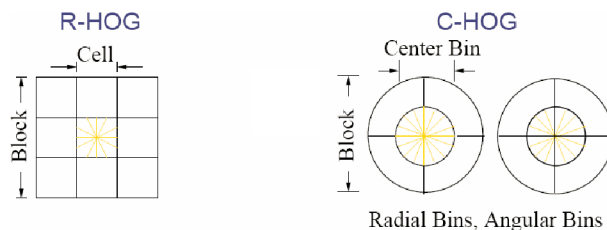
Protože jednotlivé části obrazu mají většinou rozdílné hodnoty jasu, je zapotřebí provést normalizaci jednotlivých buněk ve všech blocích, která se provede rozvinutím překrývajících se bloků nad buňkami. Existuje několik způsobů, jakými lze rozdělit obraz do bloků, nejčastěji se používá R-HOG deskriptor (podobný SFIT deskriptoru). Při použití tohoto deskriptoru mají jednotlivé bloky tvar obdélníku (odtud taky název R(ECTANGEL)-HOG). Dalším méně používaným typem je deskriptor typu C-HOG jehož bloky reprezentují kružnici. Zvolený typ bloků musí zároveň odpovídat typu použitých buněk. Pánové Dalal and Triggs zkoumali čtyři odlišné normalizační metody L1-norm, L2-norm, L1-sqrt a L2-hys. Výsledkem experimentování [6] bylo zjištěno, že použití posledních tří metod přináší podobný výkon, zatímco první metoda je o něco méně výkonná. Při vynechání normalizace bylo dosaženo výrazného zhoršení výkonu detektoru. Výsledný deskriptor tvoří vektor histogramu orientovaných gradientů jednotlivých buněk.

$$L1 - norm : f = \frac{v}{\|v\|_2 + e} \quad (3.13)$$

$$L1 - sqrt : f = \sqrt{\frac{v}{\|v\|_1 + e}} \quad (3.14)$$

$$L2 - norm : f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (3.15)$$

Metody pro normalizaci [6]



Obrázek 3.4: Zobrazení jednoho bloku a buněk [7].

Histogram orientovaných gradientů nelze použít v případě, že je objekt natočený, protože nastává změna tvaru okna a charakteristických rysů.

Knihovna OpenCV [3] (verze 2.4) obsahuje dvě sady podobných metod pro práci s HOG deskriptorem, liší se v tom, že výpočty jedné sady metod jsou akcelerovány přes GPU podporující technologii NVIDIA CUDA. V akcelerovaných metodách (prostřednictvím GPU) je možné nalézt implementovaný detektor chodců vhodný pro použití v reálném čase.

## Vlastnosti

Deskriptor využívající histogram gradientů dokáže dobře popsat tvar objektu prostřednictvím hran nebo struktury gradientů. Po provedení normalizace je invariantní vůči změně osvětlení a kontrastu. Dále je tento deskriptor částečně invariantní vůči změně geometrie, hodnota HOG se nezmění po otočení buněk, do kterých jsou obrazové data rozdělena. Hlavní nevýhodou je výpočetní náročnost deskriptoru. Velikosti gradientů můžou být dále ovlivněny šumem v nízkointenzivních oblastech popř. nelineární gama korekci v obraze.

Na základě provedených experimentů, lze tvrdit, že HOG deskriptor v kombinaci s klasifikačním SVM algoritmem podává lepší výsledky při detekci osob než detektor s Haar příznaky [6]. Na základě tohoto porovnání byl pro detektor implementovaný v této práci vybrán klasifikační algoritmus SVM společně s HOG deskriptorem.

### 3.3.5 Porovnání deskriptorů

Výše zmíněné extrakční algoritmy pro získávání deskriptoru existují v několika modifikacích, přičemž jednotlivé modifikace můžou mít odlišné vlastnosti. Následující porovnání bude vycházet z původních publikací, v jiném případě bude explicitně uvedeno o jakou verzi deskriptoru se jedná. Porovnávání rychlosti vyčíslování příznaků je problematické a jakýkoliv konkrétní údaj by byl značně subjektivní. Proto údaj rychlosti je znázorněn jako relativní porovnání k jinému popř. jiným deskriptorům. U některých deskriptorů se tento údaj nepodařilo získat.

	LBP	SURF	SIFT	HOG
Popisuje	texturu	body	body	hrany
Invariantní vůči změnám	rotace* jasu	rotace velikosti pohledu	rotace velikosti pohledu	velikosti jasu** geometrie***
Rychlost		>SIFT	<SIFT	<LBP
Vhodné pro	rozpoznávání obličejů	sledování	sledování	rozpoznávání

\* - novější verze LBP je rotačně invariantní

\*\* - v případě provedení normalizace

\*\*\* - kromě rotace

Z výše uvedených deskriptorů jsou si nejvíce podobné deskriptory vypočítané na základě algoritmu SURT a SIFT, oba popisují klíčové body snímku. SURF algoritmus dokáže toto vyčíslování provádět daleko rychleji. Používá se především pro sledování objektů, mapování prostoru nebo i pro rozpoznávání.

Popisovač založený na histogramech orientovaných gradientů se především vyznačuje možností popsat tvar objektů, z tohoto důvodu je vhodný pro klasifikaci objektů primárně se vyznačujícími hranami. Hlavní nevýhodou ovšem je výpočetní náročnost.

## 3.4 Klasifikace a rozpoznávání objektů

Klasifikace slouží pro zařazení objektů do příslušných tříd. V případě tzv. klasifikace s učitelem, při níž je klasifikátor natrénován na množině znalosti (trénovací množině) jsou objekty zařazeny s určitou pravděpodobností do tříd prostřednictvím klasifikátoru. V případě, že neexistuje znalostní množina (trénovací) jedná se o tzv. učení bez učitele, při kterém se shlukují data se společnými vlastnostmi a jednotlivé shluky reprezentují třídy. Klasifikované

objekty je zapotřebí popsat, buď to pomocí příznaků nebo pomocí struktur. V předchozí kapitole bylo uvedeno několik algoritmů jak získat deskriptory pro popis konkrétních příznaků. Pro klasifikaci existuje několik metod např. neuronové sítě (inspirované biologickým modelem), boosting algoritmy (skládání slabých klasifikátorů), rozhodovací stromy (ID3) nebo algoritmus použitý v této práci SVM.

Klasifikaci lze rozdělit do dvou fází, první fáze je tzv. učící a druhá vybavovací. Trénovací množinu tvoří ohodnocené příklady, v případě binární klasifikace jsou příklady rozdělené do dvou tříd na pozitivní a negativní. Množiny trénovacích příkladů jsou pro některé typy objektů dostupné na internetu, například množinu příkladů reprezentujících chodce lze nalézt na stránkách MIT <sup>4</sup> nebo INRIA <sup>5</sup>. Velikost trénovací množiny musí být zvolená tak, aby klasifikátor měl dostatečně vysokou přesnost a nebyl tzv. přeučení. V případě přeučení se stává klasifikátor málo generalizovaný <sup>6</sup> a zhoršují se jeho klasifikační vlastnosti.

### 3.4.1 Algoritmu AdaBoost

Algoritmus AdaBoost (Adaptive Boosting) vychází z obecné metody strojového učení Boosting. Metoda Boosting je specifická především tím, že se snaží o dosažení vyšší přesnosti učícího algoritmu a to pomocí kombinace několika slabých klasifikátorů (se špatnou detekcí) do jednoho silného. Trénovací množinou jsou dva typy objektů (vzorů) a to negativní a pozitivní. Negativní vzor popisuje prostředí, čili obrazové vzory, které nejsou objektem. Pozitivní vzory tvoří opak negativních, tudíž se jedná o obrazové vzory detekovaného objektu např. o postavy v různých pozicích. Ze vstupních obrazových filtrů se vytvoří příznaky. Od základního Boost algoritmu se AdaBoost dále liší tím, že špatně zaklasifikované body mají v další iteraci větší váhu  $D_i$ , tímto se zajistí, že algoritmus bude postupně konvergovat. Z tohoto také vyplývá, že jednotlivé trénovací body mají v každé iteraci odlišnou váhu. Chyba algoritmu je daná součtem vah špatně kvalifikovaných vzorků, trénovací chyba se s rostoucím počtem vzorků redukuje exponenciálně.

Tento algoritmus byl publikován v roce 1995 Yoavem Freundem a Robertem Schapirem [8]. V roce 1997 tento algoritmus upravili (zobecnili) Robert Schapire a Yoram Singer. V dnešní době, je možné v různých publikacích najít nespočet variant tohoto algoritmu, základ ale zůstal stejný.

Cílem následujícího algoritmu je získat hodnotu vah pro jednotlivé slabé klasifikátory, které vzniknou z prvků trénovací množiny  $\alpha_t$ .

---

<sup>4</sup><http://cbcl.mit.edu/cbcl/software-datasets/PedestrianData.html>

<sup>5</sup><http://pascal.inrialpes.fr/data/human/>

<sup>6</sup>zobecněný

---

**Algoritmus 2:** Alogritmus AdaBoost [9]

---

**Vstup:** Trénovací množina správně ohodnocených bodů

$x_i, y_i, x_i \in R^D, y_i \in \{+1, -1\}, i = 1, \dots, m$

**Výstup:** Finální klasifikátor  $H_{final}(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

Inicializuj váhy trénovacích příkladů:  $D_1(i) = \frac{1}{m}$

**for**  $t = 1, \dots, T$  **do**

Najdi slabý klasifikátor s minimální chybou distribuce  $D_t$   $h_t : X \rightarrow \{-1, +1\}$

Spočti váženou chybu:

$$\epsilon_t = \sum_{i=1}^m D_t(i) I(y_i \neq h_t(x_i))$$

Spočti váhu modelu  $\alpha_t$ :

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right) > 0$$

Uprav váhy trénovacích bodů:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \exp(-\alpha_t y_i h_t(x_i))$$

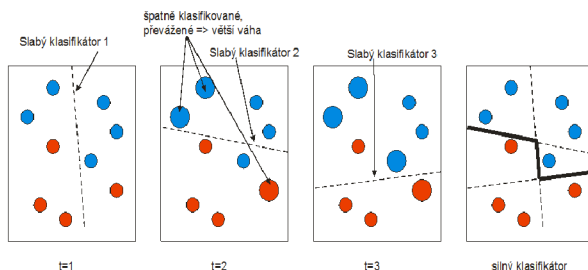
kde  $Z_t$  je normalizační konstanta

**end**

Ukončení, výstup

---

Knihovna OpenCV obsahuje několik typů AdaBoost algoritmu například diskretní nebo reálný AdaBoost. Trénování se spouští metodou `train` z třídy `CvBoost`.



Obrázek 3.5: Znázornění poskládaného výsledného klasifikátoru [13].

### 3.4.2 Algoritmus SVM (Support Vector Machine)

Algoritmus SVM [21] neboli algoritmus podpůrných vektorů je metoda strojového učení využívající podpůrné vektory a patříci do kategorie tzv. jádrových algoritmů (v tomto je popsána varianta s lineárním jádrem). Úlohou SVM algoritmu je vyhledávání optimální nadroviny, reprezentované lineární funkcí, která rozděluje trénovací data. Jednoduše lze říct, že lineární funkcí oddělíme do dvou dimenzí objekt reprezentovaný složitější funkcí. Optimální nadrovina je taková, která má maximalizovanou vzdálenost nejbližších prvků z obou tříd. Podpůrné vektory ležící v plusové nebo minusové rovině, pouze tyto mají vliv na hranici mezi třídami, ve výsledku se jedná o ty vstupní vektory, které mají nulový parametr  $\alpha$ . Výhodou tohoto algoritmu je, že dokáže řešit i lineárně neseperabilní problémy.

Změnou jádra je možné pomocí tzv. kernel trick mapovat vstup do vícerozměrného prostoru. Zjednodušeně lze říct, že každý dvourozměrný vektor lze převést na třírozměrný, přidáním třetího atributu závislého na původních dvou attributech. Tyto atributy jsou pomocí funkcí převedeny do nového vektoru. Cílem algoritmu je tudíž najít váhový vektor  $\omega$  a parametr  $b$  (bias).

$$f(x) = w^T x + b \quad (3.16)$$

Obecný popis lineárního klasifikátoru [16]

Pokud použijeme tento lineární klasifikátor pro řešení binárního klasifikačního problému, budeme předpokládat pozitivní klasifikaci  $y=1$ , jestliže  $f(x) \geq 0$  a negativní  $y = -1$  v případě  $f(x) < 0$ .

Následující rovnice znázorňují podrobněji rozepsaný klasifikátor zahrnující váhy pro jednotlivé prvky z trénovací množiny. Jádro  $k$  je lineární.

$$f(x) = \text{sign}\left(\sum_i \alpha_i y_i (x_i \top x) + b\right) \quad (3.17)$$

$$k(x, x') = (x * x') \quad (3.18)$$

Protože potřebujeme nalézt takový separátor, který maximalizuje vzdálenosti prvku dvou tříd (při binární klasifikaci). Výraz se maximalizuje prostřednictvím minimalizace váhových koeficientů.

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (3.19)$$

$$\text{kde} : \alpha_i \geq 0, \sum_i \alpha_i y_i = 0, i = 1, \dots, m \quad (3.20)$$

$$\min \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (3.21)$$

*kde* :  $x_i \top$  – podpůrný vektor,  $\alpha_i$  – váhový koeficient,  $x_i$  – vstupní příklad,  
 $y_i$  – klasifikace vstupního příkladu  $y_i \{-1, +1\}$

Prakticky se minimalizace výše zmíněného výrazu řeší pomocí některé z metod kvadratického programování. Toto odvětví řeší optimalizaci (minimalizaci nebo maximalizaci) vybrané kvadratické a konvexní funkce, přičemž omezující podmínky jsou lineární.

Knihovna OpenCV 2.4 obsahuje SVM algoritmus založený na knihovně LibSVM [5]. Knihovna využívá pro trénování klasifikátoru pomocných vektorů algoritmus SMO (Sequential Minimal Optimization) [20]. Jedná se o iterativní algoritmus pro efektivní řešení optimalizačního problému kvadratického programování, založený na rozdělení řešené úlohy na několik menších pod částí, které lze řešit analyticky.



Podle některé heuristiky se postupně volí Lagrangeovy multiplikátory  $\alpha_i$  a  $\alpha_i$  tak, aby výše uvedený výraz dosahoval maximální hodnoty. Pro zjišťování zda, algoritmus konverguje k optimálnímu bodu jsou použity tzv. Karush–Kuhn–Tucker podmínky.

Jednotlivé provedení algoritmu SMO se mohou navzájem značně lišit, především proto, že jednotlivé verze obsahují jiné optimalizační metody a heuristiky.

### Zjednodušený SMO algoritmus

Následující verze SMO algoritmu [16] je zjednodušena především tím, že pro výběr Lagrangeova multiplikátoru  $\alpha$  nepoužívá žádnou heuristiku, nýbrž indexy  $(i,j)$  identifikující multiplikátory volí náhodně.

Algoritmus začíná výpočtem hranice  $L$  a  $H$  mezi kterými leží ( $L \geq \alpha_j \leq H$ ) hodnota multiplikátoru  $\alpha_j$ .

$$L = \max(0, \alpha_j - \alpha_i), H = \min(C, C + \alpha_j - \alpha_i) \quad \text{pokud } y^{(i)} \neq y^{(j)} \quad (3.22)$$

$$L = \max(0, \alpha_j + \alpha_i - C), H = \min(C, \alpha_j + \alpha_i) \quad \text{pokud } y^{(i)} = y^{(j)} \quad (3.23)$$

Vypočítání hranic  $L$  a  $H$

Následně se vypočte nová hodnota  $\alpha_j$  pro maximalizaci 3.17. Jestliže nově získaná hodnota leží mimo hranice  $L$  a  $H$ , bude novou hodnotu hodnota nejbližší hranice. Po získání nové hodnoty  $\alpha_j$  je možné dopočítat novou hodnotu  $\alpha_i$ .

$$\alpha_j := \alpha_j - \frac{y^{(j)}(E_i - E_j)}{\eta} \quad (3.24)$$

$$E_k = f(x^{(k)}) - y^{(k)} \quad (3.25)$$

$$\alpha_j := \begin{cases} L & \text{pokud } \alpha_j > H \\ \alpha_j & \text{pokud } L \leq \alpha_j \leq H \\ H & \text{pokud } \alpha_j < L \end{cases} \quad (3.26)$$

$$\eta = 2 \langle x^{(i)}, x^{(j)} \rangle - \langle x^{(i)}, x^{(i)} \rangle - \langle x^{(j)}, x^{(j)} \rangle \quad (3.27)$$

$$\alpha_i = \alpha_i + y^{(i)}y^{(j)}(\alpha_j^{old} - \alpha_j) \quad (3.28)$$

Nakonec maximalizačního kroku se aktualizuje bias hodnota. V případě, že  $\alpha_i$  leží na intervalu  $0 < \alpha_i < C$  bude  $b = b_1$ , pokud bude ležet  $\alpha_j$  na intervalu  $0 < \alpha_i < C$  bude  $b = b_2$ . Za ostatních okolností tato hodnota bude průměrem  $b_1$  a  $b_2$ .

$$b_1 = b - E_i + y^{(i)}(\alpha_i - \alpha_i^{old}) - \langle x^{(i)}, x^{(i)} \rangle - y^{(j)}(\alpha_j - \alpha_j^{old}) - \langle x^{(i)}, x^{(j)} \rangle \quad (3.29)$$

$$b_2 = b - E_j + y^{(i)}(\alpha_i - \alpha_i^{old}) - \langle x^{(i)}, x^{(j)} \rangle - y^{(j)}(\alpha_j - \alpha_j^{old}) - \langle x^{(j)}, x^{(j)} \rangle \quad (3.30)$$

$$(3.31)$$

$$b = \frac{b_1 + b_2}{2} \quad (3.32)$$

---

**Algoritmus 3:** Pseudokód trénovacího algoritmu SMO

---

**Vstup:**

C: regulační parametr

tol: tolerance

max\_passes: maximální počet kroku při nezměněné váze  $\alpha_i$  $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$ : množina trénovacích dat**Výstup:** $\alpha \in \mathbb{R}^m$ : Lagrangovy multiplikátory $b \in \mathbb{R}^m$ : práhInicializace vah:  $\alpha_i = 0, \forall i, b = 0$ 

Inicializace passes=0

**while** passes < max\_passes **do**

num\_changed\_alphas = 0

**for**  $i, \dots, m$  **do**        Vypočítej  $E_i = f(x^{(i)} - y^{(i)})$  pomocí 3.17        **if**  $(y^{(i)}E_i < -tol \ \&\& \ \alpha_i < C) \vee (y^{(i)}E_i > tol \ \&\& \ \alpha_i > 0)$  **then**            Náhodný výběr  $i \neq j$             Vypočítej  $E_j = f(x^{(j)} - y^{(j)})$  pomocí 3.17            Uložení původních  $\alpha$ :  $\alpha_i^{(old)} = \alpha_i, \alpha_j^{(old)} = \alpha_j$ 

Výpočet L a H podle 3.22 3.23

**if**  $L = H$  **then**

| pokračování na další i

**end**            Výpočet  $\eta$  podle 3.27            **if**  $\eta \geq 0$  **then**

| pokračování na další i

**end**            Výpočet nové hodnoty pro  $\alpha_j$  podle 3.24            **if**  $|\alpha_j - \alpha_j^{(old)}| < 10^{-5}$  **then**

| pokračování na další i

**end**            Výpočet hodnoty  $\alpha_i$  pomocí 3.28            Výpočet  $b_1$  a  $b_2$  podle 3.29 3.30            Výpočet  $b$  podle 3.32

num\_changed\_alphas := num\_changed\_alphas + 1

**end**    **end**    **if** num\_changed\_alphas == 0 **then**

| passes := passes + 1

**else**

| passes := 0

**end****end**Ukončení

---



## Vlastnosti

Mezi hlavní nevýhody patří delší doba učení než u předchozího algoritmu a větší složitost. Statistickou nevýhodou může být, že nelze určit pravděpodobnost správné klasifikace. Složitost algoritmu roste s  $N^3$ .

Testování zdali se jedná o klasifikovaný objekt se provádí pomocí aplikace funkce  $h(x)$  na obraz neklasifikovaného objektu. Pokud je  $h(x) \geq 0$ , pravděpodobně se jedná o klasifikovaný objekt.

## 3.5 Detektory

### 3.5.1 Viola-Jones detektor

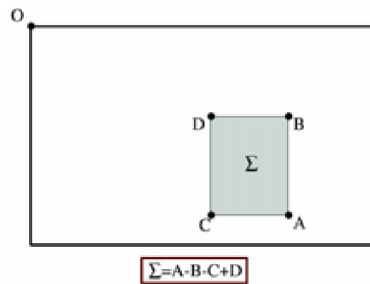
Tento detektor patří do skupiny detektorů pro identifikaci objektu v reálném čase. Princip této metody popsali v roce 2001 Paul Viola a Michael Jones [25], do té doby existovala pro detekci objektů metoda Schneiderman-Kanade (1998), která byla o několik řádů pomalejší.

Detektor Viola-Jones je primárně určený pro detekci obličejů, používá se ovšem i pro detekci jiných typů objektů. V principu je založený na metodě, která je specifická především tím, že využívá kaskádu klasifikátoru, ve které se rozhoduje sekvenčně. Detekovaný objekt je popsán pomocí příznaků, nejčastěji Haarových.

Zjednodušeně se dá říct, že Haarovy příznaky jsou založený na rozdílu jasů obdélníkových oblastí. Zmíněné příznaky vycházejí z Harrových vlnek, které jsou popsány dle funkce uvedené níže. Důvodem použití těchto příznaků je rychlost vyčíslení, přičemž rozlišovací schopnosti příznaku jsou jen o něco málo horší. Vyčíslení se provádí nad integrálním obrazem, díky kterému probíhá výpočet příznaku v konstantním čase s minimem operací. Učení probíhá na základě algoritmu AdaBoost.

$$\psi(t) = \begin{cases} 1 & 0 < t \leq 1/2 \\ -1 & 1/2 < t < 1 \\ 0 & \text{jinde} \end{cases} \quad (3.33)$$

Harrova funkce [26]



Obrázek 3.6: Obrázek znázorňuje výpočet hodnoty plochy s integrálním obrazem.

## Vlastnosti

Hlavní výhodou je především rychlost algoritmu a dostatečná přesnost při rozpoznávání obličejů.

### 3.5.2 Detektor s SVM algoritmem

Detektor navržený v této práci je založený na původním výzkumu binární klasifikace na základě histogramů orientovaných gradientů pro detekci osob publikovaného v roce 2004 [6]. Mezi základní předpoklady pro tento detektor patří především přesnost zpracování a rychlost umožňující zpracování v reálném čase. Nejkritičtější částí výpočtu je extrakce příznaku pro HOG deskriptor, oproti původnímu výzkumu je proto použita metoda výpočtu HoG z integrálního obrazu. Navržený detektor je tvořen HOG příznaky pro popis charakteristický rysů a SVM klasifikátorem s lineárním jádrem. Na internetu je možné nalézt několik porovnání, jak jednotlivých deskriptorů, tak klasifikačních algoritmů. Ovšem výsledky těchto výzkumů jsou vždy vázané na konkrétní implementaci a specifikaci užití, tudíž generické specifikování, který deskriptor ze zmíněných podává lepší výsledky nebo horší je de facto nemožné.

Jelikož video je získáváno ze statické kamery a dá se zachytit pozadí scény, byla pro segmentaci objektů zvolena metoda odčítání pozadí. Výhodou této metody oproti metodě využívající Sliding Window je především to, že se provádí klasifikace pro konkrétní objekt. V případě Sliding Window by se musela provádět klasifikace pro každé podokno a tím by se zvýšila výpočetní náročnost i riziko falešných poplachů.

### 3.5.3 Shrnutí

V této kapitole byly popsány základní techniky používané pro vytvoření detektoru specifických objektů, v případě této práce se jedná o detektor osob. První kapitola shrnuje proces analýzy obrazu v počítačovém vidění a zjednodušeně popisuje, jaké techniky se v počítačovém vidění používají, důležité techniky jsou dále rozepsány v jednotlivých podkapitolách. Uvedené informace vycházejí z původních prací a několika obsáhlých zdrojů, které jsou mezi sebou ověřené. Za nejdůležitější části z hlediska dalšího vývoje této práce lze považovat části o SIFT a HOG deskriptoru.

## Kapitola 4

# Implementace algoritmu

Další majoritní částí této práce je implementace algoritmu pro detekci objektů v jednotlivých snímcích videa s možností binární klasifikace, zda zjištěným objektem je osoba nebo ne. Algoritmus je navržen na základě metod počítačového vidění rozebraných v předchozí kapitole. Striktně se předpokládá, že vstupní video data jsou získána statickou kamerou a pro správnou funkci algoritmu bude možné zachytit z několika prvních snímků pozadí scény.

Detekce osob je založena na klasifikaci prostřednictvím HOG deskriptoru z části obrazu zachycující neznámý objekt, klasifikátor je natrénovaný pomocí SVM algoritmu (na množině negativních a pozitivních příkladů). V implementaci je dále zahrnuta možnost využití dalších dvou deskriptorů založených na klíčových bodech extrahovaných pomocí SIFT algoritmu. První z nich je reprezentován histogramem vektorů popisujících jednotlivé klíčové body a druhý využívá histogram poloh klíčových bodů. Na těchto dvou typech deskriptorů není tato práce primárně založena, slouží pouze pro experimentování a zhodnocení výhod HoG deskriptoru.

Program je naprogramovaný v programovacím jazyku C++ s využitím knihoven OpenCV (verze 2.4) a cvBlob (verze 0.10.4). I přesto, že nebyla využita akcelerace některých výpočtů pomocí grafického procesoru (GPU), dokáže algoritmus za jistých předpokladů zpracovávat video v reálném čase. Pro vývoj programu bylo použito vývojové prostředí Visual Studio 2010 ve verzi Express s příslušným kompilátorem. Pro uživatelské prostředí je využita knihovna Qt, protože jsou použity knihovny platformě nezávislé, je možné přeložit program i pod jinými operačními systémy. Výsledný program slouží jako detektor objektu s možností případné klasifikace osob. Vstupní snímky musí odpovídat formátu struktury `Ip1Image`, která je určena pro práci s obrázky.

### 4.1 Segmentace objektů

Pokud je možné získat pozadí scény a následně scénu s objekty, nabízí se pro první krok segmentace objektů metoda odčítání pozadí [3.2.1](#). Před prvním voláním metody `detect` se musí nastavit pomocí metody `setBackground` pozadí, jejímž jediným vstupním parametrem je snímek zachycující pozadí. Jestliže se tak nestane, vykonávání metody `detect` se po vypsaní chyby na standardní chybový výstup ukončí. Základní varianta metody odčítání pozadí dokáže data zpracovávat v konstantním čase, provedení této metody je možné realizovat prostřednictvím jednoho průchodu obrazových dat, což je vlastnost pro zpracování v reálném čase více než vhodná. Jedním ze vstupních parametrů algoritmu je hodnota prahu rozdílu pixelů, který se nastavuje tak, aby maska neobsahovala nepodstatné detaily a zá-

roven obsahovala data o požadovaných objektech. Ovšem ani použití prahování nezabrání mnohým defektům ve výsledné masce, proto se jako potřebné ukázalo použití morfologických filtrů. Jako první se na masku získanou pomocí metody odčítání pozadí aplikuje filtr eroze, který odstraňuje nejmenší objekty a mnohdy eliminuje i šum, který se dostal přes prahování. Po aplikaci tohoto filtru dojde k nežádoucímu poškození masky vymezující objekt, když se k tomuto přidá skutečnost, že maska objektů není dokonalá ani před aplikací tohoto filtru, je zapotřebí použít další morfologický filtr, kterým je filtr typu dilatace. V drtivé většině případů ovšem nestačí jedna aplikace těchto filtrů, ale několik aplikací, čímž se účinnost násobí. Počet aplikací filtrů eroze a dilatace, tvoří další dva parametry algoritmu.

K segmentaci jednotlivých objektů se používá knihovna `cvBlob`, získání oblasti zájmu (ROI) probíhá voláním metody `cvLabel`, která vrátí v datové šabloně typu vektor informace o objektech v obraze. Následně se odfiltrují metodou `cvFilter` malé objekty, které jsou pro další činnost algoritmu nepotřebné (prakticky se jedná o padající listí, malé hlodavce atd.). Dále jsou ignorovány objekty, jejichž poměr stran neodpovídá vzpřímené lidské postavě, je třeba poznamenat, že klasifikátor je trénován na množině chodců, takže klasifikovat osobu, která se plazí je značně problematické. Při dílčím testování se ovšem ukázala tato část jako výpočetně nejnáročnější a při segmentaci více objektů docházelo k drtivému poklesu celkového výkonů. Aby bylo možné, alespoň částečně tento problém eliminovat, je maska v případě vyšších rozlišení obrazových dat převzorkovaná na nižší rozlišení. Souřadnice ROI jsou následně vynásobeny koeficienty získaných z poměru původních rozměrů snímku ku rozměrům převzorkovaného snímku určeného pro vytvoření masky.

Jestliže by byla segmentace ukončena v této fázi, dostali bychom souřadnice oblasti, které s nějakou tolerancí vymezují objekt ve scéně. Pokud se podíváme na trénovací pozitivní snímky ze sady INRIA, zjistíme, že kolem zachycených osob se vyskytuje okraj. Aby klasifikátor mohl pracovat s co největší přesností, je zapotřebí vytvořit podobné podmínky. Z tohoto důvodu se oblast kolem segmentovaného objektu o určitou velikost zvětšuje podle poměrové konstanty. Tuto konstantu je možné změnit pouze ve třídě detektoru, nejedná se totiž o vstupní parametr detektoru. Hodnota této konstanty byla získána experimentováním a odhadem průměrné vzdálenosti osoby od okraje snímku. Po implementaci tohoto drobného vylepšení se výrazně zlepšila přesnost klasifikace.

V případě, že se na scéně delší dobu neobjeví žádný nový objekt nebo objekt minimální velikosti, uloží se nový snímek pozadí. Tato funkce přináší částečnou invarianci vůči změnám osvětlení a změnám jasu. Nevýhodou ovšem je, že pokud dojde k náhle změně pozadí, algoritmus tuto změnu bude stále detekovat jako objekt, tudíž nedojde k obnově pozadí. Řešením by bylo ukládat do seznamu polohu objektů a v případě, že objekt bude setrvávat delší dobu na stejném místě, provede se i přes tuto skutečnost uložení nového pozadí.

## 4.2 Klasifikace objektů

Tato část slouží k binární klasifikaci, zda narušitelem je člověk nebo ne. Výsledné rozhodnutí ovšem nelze brát jako absolutní, protože klasifikace se vždy provádí s jistou pravděpodobností. V předchozím kroku se podařilo získat pozici a rozměry jednotlivých objektů v obraze. Pomocí iterátoru se postupně prochází data o každém objektu, přičemž se provádí jeho klasifikace. Ze vstupního obrázku převedeného do odstínu šedi se vybere objekt, který se následně převede do rozlišení  $64 \times 128$  px (na základě [6]), poté následuje výpočet HOG deskriptoru. Pro další zlepšení klasifikace je před výpočtem HOG deskriptoru provedeno Gaussovské rozostření. Knihovna OpenCV obsahuje třídu s metodou umožňující jednoduchý výpočet histogramu gradientů. Takto získané hodnoty se vynásobí s koeficienty

získanými z klasifikační funkce, celková suma těchto koeficientů tvoří výslednou hodnotu predikce. Koeficienty pro klasifikaci byly získány z výchozího detektoru pro detekci lidí, který je obsažen v knihovně OpenCV. Vznikly trénováním pomocí algoritmu SVM na snímcích z databáze osob a pozadí INRIA. Na základě těchto faktů bylo možné použít koeficienty detektoru, jako hodnoty klasifikační funkce. Pokud je výsledná hodnota větší nebo rovná nule jedná se patrně o osobu. Stejným způsobem se postupně aplikuje i na ostatní objekty. Vstupním parametrem lze aktivovat možnost vykreslení ohraničení kolem detekovaného objektu a objektu klasifikovaného jako osoba.

Na závěr této malé podkapitoly je třeba říct, že knihovna OpenCV obsahuje hotový detektor osob, který dokonce dokáže zpracovat i video data z nestatické kamery s využitím metod Sliding Window. Tento detektor ovšem má úskalí v rychlosti zpracování (pokud se nevyužije verze s možností akcelerace přes GPU), některé části tohoto detektoru byly použity i v prvních verzích algoritmu, ovšem jejich použití se ukázalo jako značně komplikované s výrazně horším výsledkem. Jediný prvek, z detektoru obsaženého v knihovně OpenCV, použitý v tomto algoritmu je třída obsahující koeficienty pro klasifikátor.

### 4.3 Deskriptory založené na SIFT

Vytvoření deskriptoru založeného na klíčových bodech extrahovaných pomocí algoritmu SIFT, je komplikovanější než v případě HOG deskriptoru. Aby bylo možné obraz popsat, je zapotřebí vytvořit reprezentaci v podobě histogramu. První deskriptor bude tvořen popisujícími vektory klíčových bodů, přímé vytvoření histogramu je ovšem prakticky neproveditelné. Proto se prakticky využívá rozdělení celkové množiny trénovacích dat do shluku pomocí algoritmu strojového učení bez učitele k-means. Pomocí tohoto algoritmu je možné získat tzv. shluky (anglicky cluster), do každého z nich je přiřazena část z trénovací množiny. Konkrétně v OpenCV lze nalézt algoritmy pro techniky zvané Bag of Words, které slouží pro vytvoření histogramu z vícenásobných vektorových deskriptorů. Při vytváření tohoto deskriptoru se získané klíčové body z obrázku přiřazují do jednotlivých shluků, čímž vzniká histogram, jehož jednotlivé třídy jsou reprezentovány shluky. Přiřazování do jednotlivých tříd histogramu probíhá pomocí objektu třídy `BruteForceMatcher`. Velikost výsledného popisovače je ekvivalentní počtu tříd v histogramu, v případě této implementace je mapa clusteru natrénovaná do 500 tříd. Deskriptor je vypočítán na obrázku objektu o rozměrech  $64 \times 128$  pixelů v odstínech šedi, důvodem použití takového nastavení je snaha vytvořit stejné vstupní podmínky pro všechny deskriptory, aby výsledné porovnání mělo co největší vypovídající hodnotu.

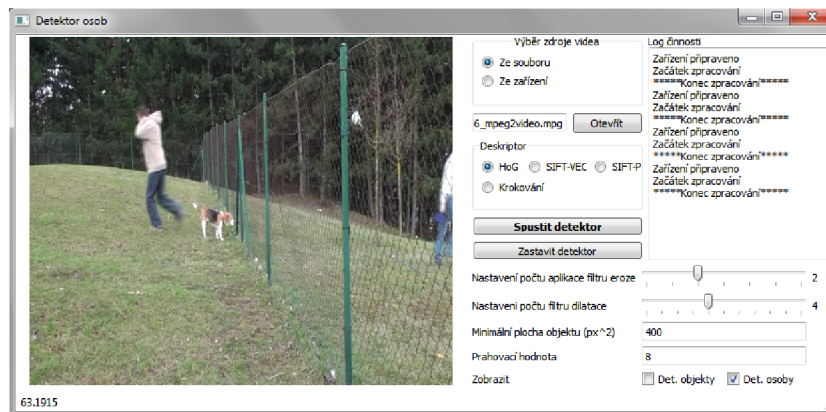
V případě vytvoření deskriptoru na základě klíčových bodů je obraz rozdělen do buněk o velikost  $8 \times 8$  obrazových bodů. Každá z buněk tvoří jednu složku histogramu. Tento typ popisovače nebyl popsán v žádném z prohledávaných zdrojů a je pouze implementován jako experimentální. Klasifikátor je natrénovaný pomocí algoritmu SVM, koeficienty získané z trénování byly uloženy do zvláštního souboru a při inicializaci detektoru jsou pokaždé načteny. Počáteční řešení, že by koeficienty byly přidány přímo do zdrojového kódu, nebylo možné provést z důvodu, že třída `CvSVM` dovoluje uložení pouze do externích XML souborů. Pokud se tyto soubor nepodaří najít v adresáři odkud je program spouštěn, dojde k vypsání chyby na standardní chybový výstup a klasifikace se pomocí SIFT popisovačů neprovede.



## 4.4 GUI

Z hlediska použitelnosti je nejlepším řešením oddělení detekčního algoritmu od uživatelského rozhraní pro nastavování parametrů. V podstatě se dá říct, že detektor tvoří samostatnou knihovnu, která je staticky linkovaná. Nabízí se i možnost vytvořit dynamickou knihovnu a tu přilinkovat k programu a volat její veřejné metody. Pro otestování algoritmu bylo vytvořené uživatelské rozhraní s využitím knihovny Qt. Tato knihovna je jedna v době psaní této práce jedna z nejrozšířenějších multiplatformních open source knihoven. Dalším důvodem pro použití této knihovny, je její neustálý vývoj a široká komunita, která se stará o testování, doladování chyb a nabízí na internetu mnoho článků o tom, jak knihovnu používat.

### 4.4.1 Prostředí



Obrázek 4.1: Uživatelské rozhraní

Samotné uživatelské prostředí je navrženo tak, aby mohly být nastaveny veškeré důležité parametry algoritmu a aby umožňovalo vizuální kontrolu zpracovávaného obrazu. Jestliže nastane úspěšná detekce narušitele (osoby) je objekt ohraničen červenou barvou, v opačném případě je objekt detekovaný objekt označen zelenou barvou. Vykreslování ohraničení objektů nebo osob lze v uživatelském rozhraní vypnout. Výstupní snímky z detektoru jsou zobrazovány na ploše widgetu QLabel.

Pro možnost porovnání deskriptorů je v uživatelském rozhraní možnost krokování po daném počtu snímku, aby bylo možné získat výsledky do porovnávacího grafu (viz. kapitola Testování 5).

### 4.4.2 Získávání obrazu prostřednictvím vlákna

Dle popisu v předchozí sekci je detektor, založen na postupném zpracování snímku získaných ze snímacího zařízení. Proto je zapotřebí vytvořit proces, který bude postupně snímky získávat a předávat je knihovně detektoru. Jelikož hlavní vlákno aplikace má na starosti základní běh uživatelského rozhraní, musí se činnost zachytávání delegovat do dalšího vlákna. Knihovna Qt obsahuje pro práci s vlákny třídu `QThread` umožňující jednoduché vytvoření nového vlákna s mnoha metodami pro řízení běhu. Vlákno pro zachytávání je ovládáno prostřednictvím GUI, které má volby pro pozastavení a obnovení běhu zachytávání a tím i detektoru. Pokud se vlákno neukončí samostatně, tzn. v případě zpracování celého videa, dojde k jeho ukončení se zavřením aplikace.

Protože výsledná aplikace by mohla běžet neustále online, nezbytně dochází k uvolňování paměti po každém provedení činnosti detektoru.

## 4.5 Shrnutí

Algoritmus pro detekování objektů a detekci osob je implementován v programovacím jazyce C++ jako třída `HumanDetector`, jejíž metoda `detect` se volá na každý získaný snímek ze zachytávacího zdroje, tzn. datový proud z kamery nebo videosouboru. Před voláním této metody musí být nastaveno pozadí pomocí `setBackground`.

Rychlost algoritmu je závislá na počtu objektů ve scéně, vstupnímu rozlišení obrazu, nastavení počtů aplikací filtru a maximální velikosti objektu. Výhodou tohoto algoritmu patří, že provádí klasifikaci segmentovaných objektů, na rozdíl od metody Sliding Window. Z toho pramení výhoda menší pravděpodobnosti falešných poplachů. Nevýhodou je nemožnost použití za situace v které nelze získat pozadí.

## Kapitola 5

# Testování

Pro otestování detektoru řešeného v této práci byly použité videozáznamy pořízené statickou HD kamerou. Aby se zmenšila výpočetní náročnost na zpracování, byly tyto záznamy převedeny do rozlišení  $640 \times 480$  obrazových bodů. Testovací záznamy obsahují pohyby osob v různých směrech a pozicích. Cílem testování je získat poznatky na reálných scénách, především je zapotřebí určit jaká je úspěšnost detekce osob. Dalším cílem testování je porovnání implementovaných deskriptorů. Nejdůležitějším aspektem pro správnou funkci algoritmu a pro dosažení nízké chybovosti klasifikátoru je potřeba správného nastavení parametrů. Především se jedná o parametr prahování, používaný při aplikaci metody odčítání pozadí a dále o parametry počtu aplikací filtru eroze a dilatace. Parametr minimální velikosti objektu není z hlediska funkčnosti algoritmu nikterak kritický, hlavní jeho úkol je ignorování nepodstatných objektů. Pokud je parametr špatně nastavený, může dojít k ovlivnění výsledných statistik.

Pro porovnání klasifikátoru se v praxi používá několik metodik. V původní práci detektoru založeného na základě histogramu orientovaných gradientů byla použita metodika pro porovnávání na základě DET křivky (Detection Error Tradeoff). Tato křivka je tvořena pravděpodobnosti, že bude správně detekovaná hledaná osoba v závislosti na FPPW (False Positives Per-Window). Metodika je ovšem vhodná pouze v případě, že se detekce osob provádí pomocí metody Sliding Window a použitých oken je několik desítek, ve výsledku až několik tisíc. V případě algoritmů se segmentací objektů pomocí metody odčítání pozadí se jeví jako lepší použít DET křivku v závislosti na FPPI (False Positives Per-Image). Jedná se o podobnou metodiku jako v předchozím případě, akorát se určuje počet falešných detekcí negativních snímků na jeden snímek. Protože videosekvence obvykle obsahují obrovské množství snímků, je možné vybírat jen snímky, podle daného kroku. Jak už to tak bývá, toto zjednodušení přináší také své úskalí v případě, že krok bude nastaven tak, aby počítal se snímky, kde zrovna nedochází k žádné detekci, může být chybovost podstatně vyšší. Toto úskalí lze eliminovat použitím menšího kroku. Zmíněné metody se používají především pro porovnávání klasifikátoru, přičemž statistická data se získávají pomocí automatického testování. Následující rovnice zachycují výpočet FPPW, v případě výpočtu by se počet detekovaných oken nahradil počtem snímků.

$$FPPW = \frac{\text{Počet pozitivních detekcí}}{\text{Počet detekovaných oken}} \quad (5.1)$$

Výpočet FPPW - False Positives Per-Window



$$\text{Recall rate} = \frac{\text{Celkový počet pozitivních oken}}{\text{Počet pozitivně označených oken}} \quad (5.2)$$

Úspěšnost pozitivně označených oken

$$\text{Precision} = \frac{\text{Správně pozitivně detekovaná okna}}{\text{Počet všech pozitivně detekovaných oken}} \quad (5.3)$$

Preciznost

$$\text{DET} = \frac{1 - \text{Recall Rate}}{\text{FPPW}} \quad (5.4)$$

Vývoj chybovosti detekce

Pro porovnání deskriptorů je zvolena závislost preciznosti v závislosti na falešných detekcích za jeden snímek (FPPI). Preciznost vyjadřuje pravděpodobnost, že pozitivní detekce je provedená na správný objekt. Důvod použití této metodiky pro porovnání je ten, že v případě vyobrazení DET křivky by hodnota pravděpodobnosti získání pozitivního výsledku byla u SIFT deskriptoru neustále 1, protože klasifikátor založený na těchto označuje za pozitivní skoro všechny objekty, takže by takto získaný výsledek neměl žádnou vypovídající hodnotu.

Program byl přeložen i spuštěn na notebooku s CPU Core i5 a operačním systémem Microsoft Windows 7 Professional 64bit, přičemž program je zkompileovaný ve Visual Studiu 2010 SP1 32 bit s optimalizací na výkon.

Algoritmus pro otestování detektoru, byl nastaven s těmito parametry:

Testovací video	Prahovací hodnota	Eroze	Dilatace
Video č.1	8	2	3
Video č.2	8	2	3
Video č.3	8	2	4

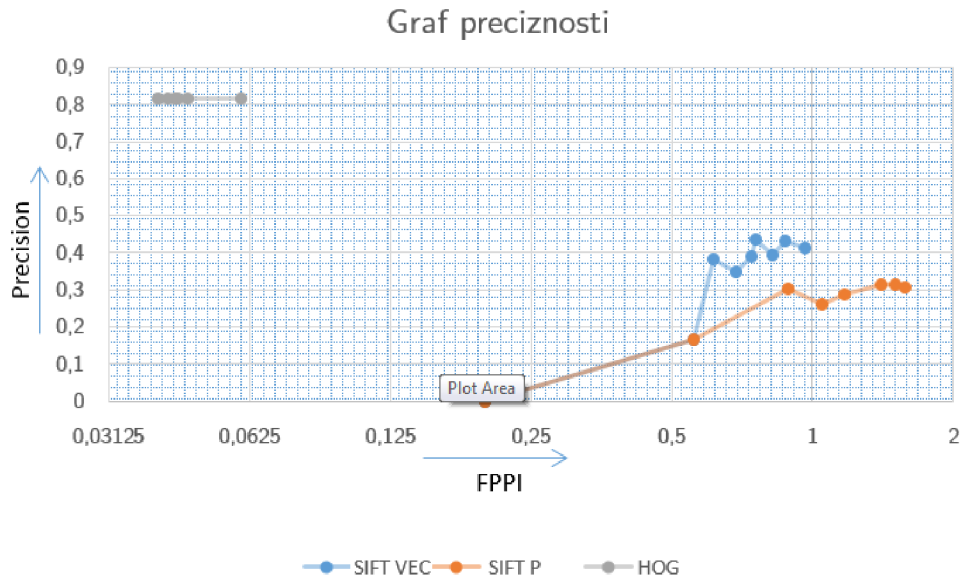
Hodnota poměru, od kterého se nebudou objekty klasifikovat, byla nastavena prostřednictvím konstanty ve třídě `HumanDetector` na hodnotu 1 (výška k šířce).

Přičemž video č. 1 a video č. 2 znázorňuje prostor, kde se pohybují osoby a psi. Zatímco video č. 3 znázorňuje scénu, kde se vyskytují pouze psi (jediná osoba je na okraji videa).

## 5.1 Detekce na videu s výskytem osob

Video č. 1 zachycuje scénu s výskytem více osob a zvířaty, zároveň zobrazuje osoby ve více měřítcích. Na tomto videu je možné detekovat několik osob v různých situacích. Toto testovací video bylo zároveň použité pro porovnání deskriptoru, na následujícím grafu je možné vidět vývoj preciznosti, graf byl vytvořen z několika snímků získaných s krokem 20 snímků za sekundu. Čím je křivka níže, tím horší dosahuje detektor preciznosti. V případě křivky reprezentující detektor s využitím HOG deskriptoru, lze vidět jen výsek křivky, protože

téměř nedocházelo k falešným detekcím, zatímco u SIFT deskriptorů byl počet falešných detekcí relativně vysoký. U detektoru s HOG deskriptorem je preciznost kolem 82 %, z čehož lze usoudit, že k falešným detekcím téměř nedochází. Ovšem detekce osob probíhá u malého množství snímků, nejdříve u 21 %, v případě výskytu dvou osob v 17 % případů a v situaci, kdy se osoba vzdálí daleko od kamery, je tato hodnota cca 14 %. V reakci na tuto situaci byla do detektoru implementována možnost aplikace detekčních oken s různými vzdálenostmi okrajů od objektů, ale toto řešení zlepšilo výsledek odhadem o několik málo procent.



Obrázek 5.1: Graf zachycující vývoj preciznosti detektoru.



(a) Video č. 1



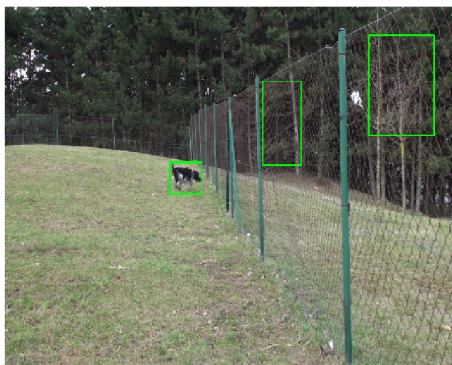
(b) Video č. 2

Obrázek 5.2: Ukázky z testovacích videí (zeleně - detekovaný objekt, červeně - detekovaná osoba)

Video č.2 je podobné jako první video, zachycuje ovšem jinak situované pohyby. Preciznost HOG detektoru se u tohoto videa pohybovala kolem 87 - 95 %. Detekce byla provedena v 17 % případů.

## 5.2 Detekce na videu bez výskytu osoby

Na tomto videu se vyskytují pouze psi, kteří se pohybují ve sledovaném prostoru. V této situaci by nemělo teoreticky dojít k žádné pozitivní detekci. V případě SIFT deskriptorů se opět potvrzuje množství falešných detekcí. Množství pozitivních detekcí se v případě tohoto videa snižuje tím, že jsou filtrovány objekty o poměru, který neodpovídá vzpřímené postavě. S klasifikátorem založeným na HOG deskriptoru došlo k zachycení falešných detekcí na třech snímcích. V případě, že byl krok nastaven na 20 snímků, nebyla zachycena žádná falešná detekce. Detektor aplikovaný na toto video pracuje s přesností blízkou se 100 %.



Obrázek 5.3: Ukázka z videa č. 3 na které je zachycen pes detekovaný jako objekt.

## 5.3 Souhrn

Na základě několika testovacích videí byl otestován detektor osob. V případě použití SIFT deskriptorů založených na rozložení klíčových bodů v okně klasifikovaného objektu a na základě vektorů popisující klíčové body, byla úspěšnost klasifikace velice špatná. Tato situace může být způsobena například tím, že u každého pozitivního trénovacího snímku jsou zachyceny jak klíčové body osoby, tak i pozadí. Navíc klíčové body jednotlivých snímků mají malý společný průnik vlastností (pozic, vektoru). V případě dvou různých osob, které se snaží vystihnout stejnou polohu v jiném prostředí a za jiných světelných podmínek, nemusí být rozložení klíčových bodů stejné. Z těchto důvodů klasifikátor založený na těchto rysech dosahuje horších výsledků než klasifikátor natrénovaný na množině deskriptorů histogramu gradientů. Dalším důležitým aspektem je fakt, že počet klíčových bodů nalezených v případě snímku s osobou je průměrně kolem třiceti (zjištěno výpisem klíčových bodů při ladění programu), pokud obrázek zachycuje členitě a geometricky složitější pozadí je tento počet vyšší. SIFT deskriptory v této konfiguraci, lze hodnotit jako prakticky nepoužitelné.

V případě použití HOG deskriptorů byla přesnost detekce průměrně 85 %. Obecně se dá tvrdit, že detekce je úspěšná v případě jasně zachytitelné vzpřímené postavy, pokud se povede segmentace. V případě, že je postava osoby narušená nějakým objektem, například sloupkem plotu, je detekce neúspěšná. Z testování vyplývá také jedna záporná vlastnost detektoru, detekce je provedena jen u cca 18 % snímků. Tato vlastnost by se dala zlepšit natrénováním klasifikátoru pro konkrétní prostředí, kde by se algoritmus používal. Veškeré uvedené číselné údaje slouží pouze jako orientační získané při konkrétním nastavení na konkrétních videích. Detektor byl také vyzkoušen na testovacích videích v původním rozlišení, detektor ovšem takřka nefungoval. Důvod špatného fungování je pravděpodobně v nevhodném algoritmu pro převzorkování obrazu nebo v nedostačujícím kodeku.

# Kapitola 6

## Závěr

V rámci této práce vznikl detektor osob a objektů pro použití v reálném čase. Program byl navržen tak, aby byl použitelný v oblasti sledování hlídaného prostoru a detekci narušitele. Pro otestování detekčního algoritmu byla vytvořena aplikace s uživatelským rozhraním sestaveným za pomoci knihovny Qt, veškerý zdrojový kód je napsán v programovacím jazyce C++ s využitím knihovny OpenCV verze 2.4.. Knihovna pro segmentaci objektu z masky cvBlob (verze 0.10.4) je linkovaná k programu staticky. V této práci byly porovnány tři různé deskriptory, přičemž nejlepší výsledky podával deskriptor založený na histogramu orientovaných gradientů. Podle teoretických předpokladů klasifikátor založený na deskriptorech využívající klíčových bodů získaných pomocí extrakce algoritmem SIFT, nedosahoval preciznosti klasifikátoru založeného na HOG. HOG deskriptor byl zvolen hlavně proto, že dokáže popsat objekt pomocí vektorů. Každý objekt je převzorokován do rozlišení  $64 \times 128$  obrazových bodů, v případě, že by oblast vymezující objekt byla vzdálena poměru 1:2, docházelo by k radikální deformaci tvaru. Tento problém řeší odfiltrování objektů s určitým poměrem. Možnost použití v reálném čase, byla dosažena na základě použití metody odčítání pozadí.

Aplikace klasifikátoru se provádí jen na segmentované objekty, čímž se šetří výpočetní čas. Oblast segmentovaných objektů bylo třeba o určitou poměrnou část z celkového obrazu rozšířit, aby získaný objekt přibližně korespondoval se snímkem, na kterých byl klasifikátor natrénovaný. I přesto, že detektor nedokáže detekovat osobu na všech snímcích, zachytí její pohyb alespoň na tolika snímcích, aby bylo možné prohlásit, že sledovaný prostor byl narušen osobou. Pro zvýšení počtu správných pozitivních detekcí byl algoritmus rozšířen o možnost vícenásobné detekce objektu různě velkými okny (změnou velikosti okrajů od oblasti objektu). Toto řešení ovšem nepřineslo výrazné zlepšení. V případě, že by se osoba potencionálně pohybovala v oděvu barevně blízkému pozadí, byla by detekce neúspěšná.

Oblast segmentace objektů a detekce osob je poměrně rozsáhlá, tato práce se zabývá pouze její malou částí. Teoretická část práce obsahuje popisy základních metod a algoritmů pro klasifikaci osob. Na základě implementovaného algoritmu byly získané poznatky o segmentaci objektů, extrakci charakteristických příznaků a klasifikaci osob. Některé z nich korespondovaly s poznatky uvedenými v původní práci detektoru s využitím HOG, především aplikace filtru pro odstranění vysokofrekvenčních složek (Gaussovské rozostření) na klasifikovaný obraz. Z vlastních poznatků uvedených v této práci se dá zmínit možnost segmentování objektů na obrazu převedeného do nižšího rozlišení.

I přesto, že algoritmus splňuje zadané požadavky této práce, nabízí spoustu možností rozšíření. V části segmentování objektů je možné zlepšit, jak vlastní segmentaci, tak filtrování oblastí, ve kterých nemůže být člověk atd.. Klasifikaci je možné zlepšit implementováním kaskády klasifikátoru např. v kombinaci dvou HoG klasifikátoru natrénovaných na jiné mno-

žině dat. Ovšem možnosti rozšíření je ve skutečnosti daleko více. Deskriptory založené na klíčových bodech získaných pomocí SIFT pro detekci osob v reálném čase, nemá moc velký vyznám dále rozšiřovat, tato cesta se jeví jako slepá.

Když celou práci shrneme, zjistíme, že byl vytvořen detektor pro detekci osob s rotační invariantností a částečnou invariantností vůči světelným změnám. S tím, že dokáže při rozlišení  $640 \times 480$  obrazových bodů (hypoteticky i při vyšším rozlišení) zpracovávat snímky v reálném čase. Nutnou podmínkou pro správnou funkci je použití videozáznamu získaného ze statické kamery.

# Literatura

- [1] Barnum, P.: Pedestrian Detection Histograms of Oriented Gradients for Human Detection Navneet Dalal and Bill Triggs . [online], 2011.  
URL [http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide\\_corso/A54b%20Dalal%20&%20Triggs'%20Pedestrian%20detector.pdf](http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide_corso/A54b%20Dalal%20&%20Triggs'%20Pedestrian%20detector.pdf)
- [2] Bay, H.; Tuytelaars, T.; Gool, L. V.: Surf: Speeded up robust features. In *In ECCV*, 2006, s. 404–417.
- [3] Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [4] Casey, M.: What Are PIR Sensors? [online], 2013.  
URL <http://www.wisegeek.com/what-are-pir-sensors.htm>
- [5] Chih-Chung Chang, C.-J. L.: LIBSVM: A Library for Support Vector Machines. [online], 2012.  
URL <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>
- [6] Dalal, N.; Triggs, B.: Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, ročník 1, 2005, ISSN 1063-6919, s. 886–893 vol. 1.
- [7] Dalal, N.; Triggs, B.: The Simplified SMO Algorithm. [online], 2009.  
URL <http://cs229.stanford.edu/materials/smo.pdf>
- [8] Freund, Y.; Schapire, R. E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, ročník 55, č. 1, 1997: s. 119 – 139, ISSN 0022-0000.  
URL <http://www.sciencedirect.com/science/article/pii/S002200009791504X>
- [9] Freund, Y.; Schapire, R. E.: A Short Introduction to Boosting. [online], 1999.  
URL <http://www.yorku.ca/gisweb/eats4400/boost.pdf>
- [10] Kaisler, R.: Detektory PIR - 1. díl. [online], 2013.  
URL <http://www.slaboproudy.cz/index.php/sezncla/86-pujsi1>
- [11] Liu, Y.; Yao, H.; Gao, W.; aj.: Nonparametric background generation. *Journal of Visual Communication and Image Representation*, ročník 18, č. 3, 2013: s. 253 – 263, ISSN 1047-3203.  
URL <http://www.sciencedirect.com/science/article/pii/S1047320307000028>
- [12] M., P.; T., O.: Texture analysis in industrial applications. *Springer-Verlag*, 1996.

- [13] Železný Miloš: ADABOOST Detekce objektu v obraze. [online], 2013.  
URL [http://www.kky.zcu.cz/uploads/courses/zdo/lesson8/ADABOOST\\_cz.ppt](http://www.kky.zcu.cz/uploads/courses/zdo/lesson8/ADABOOST_cz.ppt)
- [14] Měkota, I. V.: Davová bezpečnost - Online identifikace osob. [online], 2002.  
URL <http://itsolutions.vitkovice.cz/37/cs/node/2183>
- [15] Nán, C. C. L.: cvBlob. <http://cvblob.googlecode.com>.  
URL <http://cvblob.googlecode.com>
- [16] Ng, A.: The Simplified SMO Algorithm. [online], 2009.  
URL <http://cs229.stanford.edu/materials/smo.pdf>
- [17] Nixon, M.; Aguado, A.: *Feature Extraction & Image Processing*. Academic Press, 2008, ISBN 9780123725387.  
URL <http://books.google.cz/books?id=jXmJqzQgdY8C>
- [18] Ojala, T.; Pietikäinen, M.; Mäenpää, T.: A Generalized Local Binary Pattern Operator for Multiresolution Gray Scale and Rotation Invariant Texture Classification. In *Proceedings of the Second International Conference on Advances in Pattern Recognition, ICAPR '01*, London, UK, UK: Springer-Verlag, 2001, ISBN 3-540-41767-2, s. 397–406.  
URL <http://dl.acm.org/citation.cfm?id=646260.685274>
- [19] Pietikäinen, M.: Local Binary Patterns. ročník 5, č. 3, 2010: str. 9775.
- [20] Platt, J.; aj.: Sequential minimal optimization: A fast algorithm for training support vector machines. [online], 1998.
- [21] Pošík, P.: Optimální rozdělující nadplocha.Support vector machine. Adaboost. [online], 2012.  
URL [http://cw.felk.cvut.cz/lib/exe/fetch.php/courses/a3m33ui/prednasky/svm\\_adaboost-handouts.pdf](http://cw.felk.cvut.cz/lib/exe/fetch.php/courses/a3m33ui/prednasky/svm_adaboost-handouts.pdf)
- [22] Sinha, U.: SIFT: Scale Invariant Feature Transform. Květen 2010.  
URL <http://www.aishack.in/2010/07/implementing-sift-in-opencv/>
- [23] Skalka, M.: Prostředky relativní lokalizace. [online], 2002.  
URL <http://marek.sk.sweb.cz/lokalizace/kapitola3.html>
- [24] Tamersoy, B.: Background Subtraction. [online], 2009.  
URL [http://www.cs.utexas.edu/~grauman/courses/fall2009/slides/lecture9\\_background.pdf](http://www.cs.utexas.edu/~grauman/courses/fall2009/slides/lecture9_background.pdf)
- [25] Viola, P.; Jones, M.: Robust Real-time Object Detection. In *International Journal of Computer Vision*, 2001.
- [26] Weisstein, E. W.: Haar function. Visited on 27/03/13.  
URL <http://mathworld.wolfram.com/HaarFunction.html>