

České vysoké učení technické v Praze  
Fakulta elektrotechnická

# **DIPLOMOVÁ PRÁCE**



## **Detekce osob v dohledových videozáznamech**

Bc. Petr Vávra

Vedoucí diplomové práce: Ing. Jaromír Doležal, Ph.D.

2015

České vysoké učení technické v Praze  
Fakulta elektrotechnická

Katedra kybernetiky

## ZADÁNÍ DIPLOMOVÉ PRÁCE

**Student:** Bc. Petr V á v r a  
**Studijní program:** Kybernetika a robotika (magisterský)  
**Obor:** Robotika  
**Název tématu:** Detekce osob v dohledových videozáznamech

### Pokyny pro vypracování:

1. Proveďte studii stavů a událostí, které je ve vnitřním prostředí možno detekovat na základě analýzy záznamu z kamerového dohledového systému. Cílem je detekce obvyklých a neobvyklých událostí - počítání osob v místnosti, jejich průchodů určitými zónami nebo detekce pádu osoby či trvale ležící osoby v místech nezvyklých apod...
2. Implementujte SW v programu Matlab, který bude zpracovávat záznamy z kamer, detekovat pády osob a případné další události. SW má sloužit pro podporu dohledu nad seniory či postiženými osobami v domově, stacionáři či nemocničním prostředí.
3. Ověřte funkci SW na záznamech z kamerového systému v Centru Asistivních Technologií.

### Seznam odborné literatury:

- [1] Hlaváč V., Sedláček M.: Zpracování signálů a obrazů, skriptum, Vydavatelství ČVUT, 2009.
- [2] Fisher R. B., et al.: Dictionary of Computer Vision and Image Processing, New York: Wiley, 2005.
- [3] Šonka M., Hlaváč V., Boyle R.: Image Processing, Analysis and Machine Vision, Thomson, 3rd edition, ISBN 978-0-495-08252, 2007.

**Vedoucí diplomové práce:** Ing. Jaromír Doležal, Ph.D.

**Platnost zadání:** do konce zimního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic  
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 8. 7. 2014

## **Abstrakt**

Diplomová práce se zabývá studií možných detekovatelných stavů ve vnitřních prostorách pomocí kamerového dohledového systému. Nejdůležitějším stavem, který je třeba detekovat je pád osoby. Detekce pádu osoby je důležitá zejména u starších osob, protože když nedojde k okamžité pomoci, může mít pád vážnější následky. V práci navrhuji software pro automatickou detekci pádu. Nejprve detekuji pohybující se objekty v obraze základní metodou pro odečítání pozadí, čímž vznikne silueta osoby. Pád osoby detekuji z rychlosti a směru pohybu těžiště siluety. Software testuji na kamerách v prostorách Centra asistivních technologií, zda dosáhne, dostatečné přesnosti detekce pro možné nasazení v praxi.

## **Abstract**

The diploma thesis deals with the study of possible states detectable in indoor areas using surveillance videos. The most important state which is necessary to detect is the fall of the person. The fall detection is especially important for older people, because if there is no immediate help, the fall may have serious consequences. In this paper, I propose software for the automatic fall detection. First I detect moving objects in the image using method for background subtraction, so I create a human silhouette. Fall of person I detect from the speed and direction of movement of the center of gravity silhouettes. I test the software for the cameras at the Centre of assistive technology that achieves sufficient accuracy of the detection for possible use in practice.

### **Prohlášení autora práce**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne .....

.....

Podpis autora práce

## **Poděkování**

Rád bych poděkoval vedoucímu mé diplomové práce Ing. Jaromíru Doležalovi, Ph.D. za trpělivost a cenné rady při vytváření této práce. Děkuji také své rodině, která mě podporovala během studia.

# Obsah

1 Úvod.....	1
2 Stav současného poznání .....	2
2.1 Demografie.....	2
2.2 Monitoring osob v domácím prostředí.....	3
2.3 Práce jiných autorů.....	5
2.3.1 Model pozadí po detekci pohybujících se objektu .....	6
2.3.2 Detekce pohybujících se objektů bez modelu pozadí.....	7
2.3.3 Detekce pádu .....	9
3 Cíle práce .....	14
4 Teoretická část .....	15
4.1 Použité metody zpracování obrazu .....	15
4.1.1 Gaussovský filtr.....	15
4.1.2 Matematická morfologie.....	16
4.2 Studie stavu a událostí.....	18
4.3 Problematika videí v reálných podmínkách.....	19
5 Praktická část .....	21
5.1 Matlab Vision Toolbox .....	21
5.1.1 Shrnutí výsledků Matlab Vision Toolboxu .....	26
5.2 Algoritmus.....	27
5.3 Aktualizace pozadí .....	28
5.4 Vytvoření siluety .....	32
5.5 Úprava siluety .....	36
5.6 Párování objektů mezi snímky .....	41
5.7 Detekce pádu.....	45
5.7 Přehledové video .....	48
6 Závěr .....	49
Literatura.....	50
Obsah CD.....	52

# Seznam obrázku

Obr. 2. 1 Složení obyvatelstva v roce 2010 a 2030 .....	2
Obr. 2. 2 IP kamery a jejich komunikace.....	4
Obr. 2. 3 Náramkový detektor pádu a náramek pro přivolání pomoci .....	5
Obr. 2. 4 Silueta .....	6
Obr. 2. 5 Vektorové pole pohybu.....	8
Obr. 2. 6 Trojrozměrná reprezentace pohybu .....	9
Obr. 2. 7 Obrázek pozadí, klečící osoby v popředí a výsledné siluety .....	9
Obr. 2. 8 Hrana siluety .....	10
Obr. 2. 9 Obrázek ležící a kráčeující siluety .....	10
Obr. 2. 10 Označená osoba a její hrubá silueta.....	11
Obr. 2. 11 Siluety při různém přístupu aktualizace pozadí.....	12
Obr. 2. 12 Obrázek znázorňující množství pohybu v obraze.....	13
Obr. 4. 1 Gausián.....	16
Obr. 4. 2 Příklad dilatace .....	17
Obr. 4. 3 Příklad eroze .....	17
Obr. 5. 1 Základní detekce obličejů .....	22
Obr. 5. 2 Detekce obličejů se změněným Scale Factorem.....	23
Obr. 5. 3 Detekce obličejů se změněným Merge Thresholdem .....	24
Obr. 5. 4 Základní detekce vzpřímené postvy .....	24
Obr. 5. 5 Detekce vzpřímené postvy se změněnou velikostí učících obrázků.....	25
Obr. 5. 6 Detekce vzpřímené postavy se změněnou velikostí Classification Thresholdu ...	26
Obr. 5. 7 Ukázka siluet pro různé hodnoty $\alpha$ .....	30
Obr. 5. 8 Ukázka pozadí s částečným otiskem pohybujícího se objektu.....	31
Obr. 5. 9 Skutečný snímek, kde se nachází „duch“ .....	32
Obr. 5. 10 Odečtení aktuálního snímku od pozadí v ideálních podmínkách .....	33
Obr. 5. 11 Odečtení aktuálního snímku od pozadí v reálných podmínkách .....	33
Obr. 5. 12 Ukázka siluet pro různé hodnoty $p$ .....	34
Obr. 5. 13 Binární obrázek siluety v ideálních podmínkách.....	35
Obr. 5. 14 Binární obrázek siluety v reálných podmínkách .....	36
Obr. 5. 15 Červeně označené části člověka v ideálních podmínkách .....	38
Obr. 5. 16 Červeně označené části člověka v reálných podmínkách.....	39
Obr. 5. 17 Celistvá silueta po aplikaci morfologických operacích .....	40
Obr. 5. 18 Celistvá silueta po aplikaci morfologických operacích .....	40
Obr. 5. 19 Skutečný a binární snímek sloučení dvou silueta .....	43
Obr. 5. 20 Skutečný a binární snímek rozdělení dvou silueta .....	44
Obr. 5. 21 Žádný pád .....	46
Obr. 5. 22 Znázorněné příklady možného pádu.....	47
Obr. 5. 23 Pád .....	48





# 1 Úvod

V posledních letech se stále více pozornosti věnuje detekci lidských osob ve videích monitorovacích systémů. Tyto systémy mají široké použití v aplikacích např. detekci různých událostí, charakterizaci lidské chůze, počítání osob v místnosti, identifikaci osob, klasifikaci pohlaví a pro moji práci nejdůležitější, detekci pádu starších či nemocných lidí. Proto v této práci nejprve provedu studii, ve které zjistím, jaké stavy je možné pomocí dohledových videozáznamů detekovat. Podívám se, jak tyto stavy detekují jiní autoři a které metody používají. Zjistím, jaké jsou jejich výhody a nevýhody, případně které by se daly použít pro můj vlastní software.

Zaměřím se hlavně na detekci pádů, která je velice důležitá zejména u starších osob, kde pokud nedojde k okamžité pomoci spadlé osoby, může mít pád mnohem vážnější následky. Tradiční bezpeční nebo kamerové systémy spoléhají na nepřetržité centralizované sledování videozáznamu operátorem nebo je třeba nosit senzory pro detekci pádových událostí. Tyto způsoby detekce pádu jsou velice neefektivní a náročné na lidské zdroje nebo nepříjemné pro monitorované osoby. Já se naopak budu snažit navrhnout software pro automatickou detekci nebezpečných stavů, který by umožnil sledovat velké množství osob s malými nároky na lidské zdroje a přitom nikterak neobtěžoval a neznepříjemňoval život monitorované osobě.

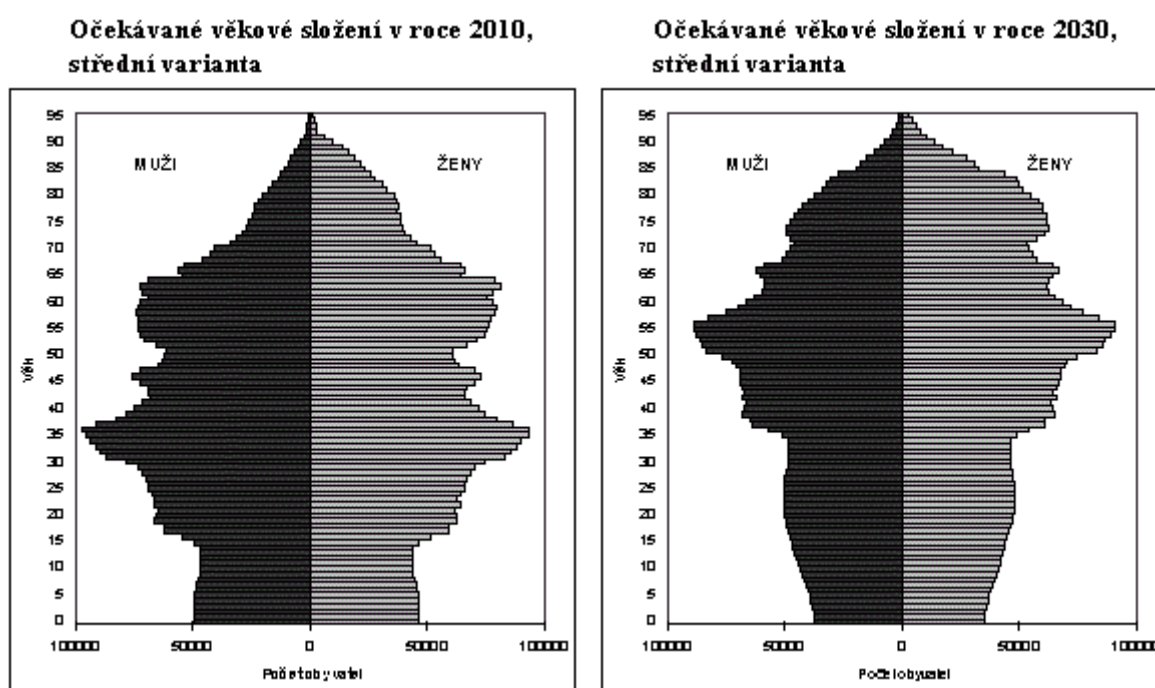
Můj software bude vycházet ze znalostí metod, které získám při studii detekovatelných stavů. Jejich pomocí navrhnu vlastní software pro detekci pádů. Tento software budu navrhovat tak, aby se dal použít v praxi. Tedy, aby byl dost robustní a vypořádal se s běžnými problémy celodenního sledování (změna osvětlení, drobný pohyb na pozorované scéně a další).

Nakonec budu přesnost mého softwaru testovat v Centru asistivních technologií, kde se nachází vybavený demonstrační byt s kamerami pro domácí dohled. V těchto prostorách budu můj software testovat v ideálních podmínkách, pro co nejvyšší přesnost softwaru, ale i ve skutečných podmínkách, aby co nejvěrněji reprezentovaly nasazení softwaru v praxi. Porovnáím, jakou úspěšnost detekce software bude dosahovat v obou podmínkách.

## 2 Stav současného poznání

### 2.1 Demografie

V současné době všechny země musí čelit rostoucí populaci starších osob. Konkrétně obyvatelstvu České republiky, obdobně jako ve všech ostatních evropských zemích, dlouhodobě klesá podíl dětské složky a naopak narůstá podíl osob ve vyšším věku. Podíl dětského obyvatelstva by podle střední varianty projekce měl v roce 2030 dosahovat 12 %, podíl osob starších 60 let 31 %, starších 65 let více než 24 %. Očekávané změny věkového složení jsou zobrazeny na obrázku Obr. 2.1.



Obr. 2. 1 Složení obyvatelstva v roce 2010 a 2030

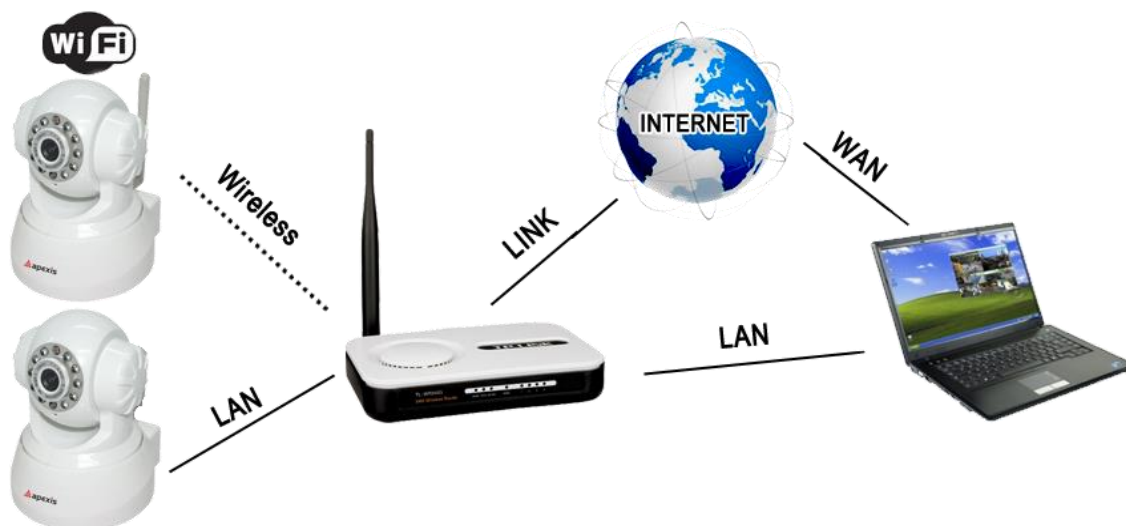
Konstrukce bezpečnostních prostředí poskytujících kvalitní zdravotní péči se stává stále více důležitou otázkou, zejména u starších lidí, kteří žijí sami. Pro většinu starších lidí vznikají jedny z nejvážnějších zranění právě při pádech. Studie [20] ukázala, že 20 až 30% starších lidí, kteří upadli, utrpěli středně těžká zranění, jako jsou zlomeniny krčku stehenní kosti a poranění hlavy, které ovlivňují jejich mobilitu a nezávislost. Kromě toho, je pád také způsoben jinými závažnými onemocněními, např. infarkt myokardu nebo cévní mozková příhoda, zpoždění při záchraně upadlé starší osoby může mít za následek nevratné zmrzačení nebo dokonce i smrt.

## 2.2 Monitoring osob v domácím prostředí

V dnešní době existují firmy zabývající službou tísňové péče, jejíž pomocí lze snížit zdravotní a bezpečnostní rizika občanů, například seniorů, osob žijících osaměle, fyzicky handicapovaných osob a ostatních, kteří potřebují bezpečí. Tyto firmy nabízejí mnoho služeb například nepřetržitý 24h dohled, přijetí nouzového signálu, hlasová komunikace s hlídanou osobou, zajištění záchranné služby či zajišťujících osob, komunikace se záchrannou službou a ověření stavu osoby, kontaktování lékaře, pravidelné volání 7x týdně, upomínková služba (připomenutí užití léku, návštěvy lékaře, výročí členů rodiny atd.).

Ovšem většina těchto systémů pracuje tak, že v případě nouze stisknete nouzové tlačítko, které můžete nosit na ruce jako hodinky, či jako přívěšek. V okamžiku stisknutí se aktivuje zařízení a jste okamžitě spojeni s centrálním dispečinkem, kterému se o Vás zobrazí potřebné informace na displeji počítače. V případě, že máte pádový detektor (o jejich druzích se budeme bavit níže) a dojde u Vás k pádu či ztrátě vědomí, bude tento detektor aktivován automaticky. Ale přímo kamerový monitoring, který je hlavním tématem této práce, buď nepoužívají nebo ho používají jen pro zkontrolování situace, obdrží-li nouzový signál a ne pro samotnou detekci krizové situace. Mnohem častěji se kamerový monitoring používá při hlídání určitých prostor, jako například firemních budov či skladů, ale toto není tématem této práce.

Jak uvádí článek [1] obraz získaný z monitorovacího videa je obvykle s nízkým rozlišením a je pořízen statickou kamerou s minimální změnou pozadí. Pro pořízení monitorovacího videa se nejčastěji používají IP kamery, kterých je velký výběr, a jsou poměrně levné. Dají se ovládat přístupem po síti přes TCP/IP protokol a obsahují mikrofon + reproduktor, pro komunikaci s dohlíženou osobou. Jejich nevýhoda je velká změna pozorovací scény v závislosti na osvětlení a v noci nevidí vůbec, toto se částečně řeší IR diodami pro podsvícení v noci. Tyto problémy se dají řešit použitím přímo IR kamer, které vidí stejně dobře za jakýchkoliv světelných podmínek, ovšem jsou mnohonásobně dražší a v praxi by byl monitorovací systém příliš drahý pro monitoring seniorů. Já používám IP kamery, jejichž obrázek a komunikace s obsluhou je znázorněna na Obr. 2.2.



*Obr. 2. 2 IP kamery a jejich komunikace*

IP kamery jsou dostatečně levné, aby se daly použít v aplikaci pro kamerový dohled, která musí mít přijatelnou cenu, aby se dala použít v praxi. Proto tyto kamery mají i určité nevýhody, jako například, že nedokáží zaručit konstantní snímkovou frekvenci. Video z kamery je potom trhané a není plynulé. To značně ztěžuje jeho následné zpracování. Objekty se ve videu pohybují trhaně a to ztěžuje jejich identifikaci i odhad rychlosti s jakou se pohybují, která je zásadní pro detekci pádu, jak bude uvedeno v pozdější kapitole.

Většina stávajících monitorovacích systémů spoléhá na lidské pozorovatele pro detekci speciálních událostí na monitorovacích displejích. Nicméně toto je omezeno lidskou schopností sledovat současně více událostí na více displejích. Z tohoto důvodu se analýza lidského pohybu v automatizovaném kamerovém systému stala velice atraktivním a aktivním tématem výzkumu v oblasti počítačového vidění a rozpoznávání.

Systémy pro detekci nebezpečných stavů se dají rozdělit do dvou skupin, na nositelné přístroje a přístroje v prostředí. Nositelné přístroje jsou například mikropřínače, akcelerometry a gyroskopy, které jsou vloženy do oděvu nebo vycházkových holí. Ukázky náramků pro seniory, kterými se dá vyslat nouzový signál, jsou na obrázku Obr. 2.3. Tyto senzory mohou detekovat změny v držení těla nebo pohybu nositele. Je-li detekována abnormální změna v pohybu, je spuštěn alarm. Nositelné zařízení jsou jednoduché a levné, ale narušují normální život uživatele. Ve výsledku mnozí uživatelé nejsou ochotni je nosit nebo je zapomenou použít. Přístroje v prostředí se používají například tlakové a vibrační čidla v podlaze nebo pod postelí. Mohou zjistit přítomnost a pozici osoby. Tyto přístroje jsou levné a neruší uživatele, ale produkují velké množství falešných poplachů.



*Obr. 2. 3 Náramkový detektor pádu a náramek pro přivolání pomoci*

Překonat nedostatky z výše uvedených systémů se v poslední době lidé pokouší pomocí kamerových vizuálních systémů. Vizuální systémy mají tu výhodu, že nenarušují normální život uživatele, ale tu nevýhodu, že shromažďují informace o soukromém životě. Osobní soukromí jednotlivce závisí na osobě sedící u monitorů tohoto vizuálního systému. Z tohoto důvodu jsou zapotřebí automatické detekční systémy, které extrahují některé prvky z video snímků a na jejich základě provádí rozhodování a detekci nebezpečných stavů. Osobě u monitorů mohou zobrazovat jen takový záznam, který by chránil soukromý sledované osoby.

Proto se mnozí odborníci snaží navrhnout řešení založené na vizuální detekci nebezpečných stavů. Používají se videokamery pro zachycení lidské činnosti v rizikových oblastech a aplikují se pohyb detekující techniky k identifikaci nebezpečných stavů. Díky integraci mechanismu s detekcí nebezpečných stavů v monitorovacích systémech si mohou pečovatelé nebo rodinní příslušníci ověřit pomocí monitorovací kamery, co se stalo, ihned jakmile byl detekován například pád nebo uklouznutí starší osoby.

## **2.3 Práce jiných autorů**

Detekce pohybujících se objektů většinou probíhá ve dvou krocích: detekce objektů a klasifikace objektů. Pro detekci objektů potřebujeme inteligentní systém, který detekuje pohyb a zachycuje informace o pohyblivých cílech pro přesnou klasifikaci objektů. Klasifikovaný objekt je sledován pro podrobnější analýzu. V této práci se zaměřím přímo na detekci lidí a ne přímo na jejich konkrétní činnosti. Detekce lidí je obtížný úkol z pohledu strojového vidění, protože je ovlivněna velkým rozsahem možností vzhledu detekovaného cíle kvůli rozdílným tělesným propozicím každého člověka, měnícímu se oblečení, osvětlení a pozadí. Ale při částečné znalosti těchto vlastností můžeme výrazně zlepšit detekci. Tato detekce se může provádět odečtením pozadí, optickým tokem nebo časově-prostorovou filtrací.

### 2.3.1 Model pozadí po detekci pohybujících se objektu

Odečítání pozadí je oblíbená metoda pro detekci objektů, kde se detekce pohybujících se objektů provede z rozdílu mezi aktuálním snímkem a snímkem pozadí pro každý pixel nebo pro jednotlivé bloky v obraze. Nejdůležitější pro tuto metodu je kvalitně vytvořený model pozadí. Metodami pro jeho vytvoření se budeme zabývat v této kapitole. Po odečtení pozadí od aktuálního snímku vzniká silueta pohybujícího se objektu, která je vyobrazena na obrázku Obr. 2.4.



*Obr. 2. 4 Silueta*

Autoři [2] vytvářejí model pozadí jako směs adaptivních Gausiánů, která velmi dobře zvládá změny v dynamické scéně způsobené změnami osvětlení, vnějšími událostmi, a dalšími. Spíše než modelování hodnot všech pixelů v obraze jako jeden konkrétní druh distribuce, je každý pixel modelován jako směs Gausiánů, kdyby se v průběhu času měnilo pouze osvětlení, stačil by k modelaci každého pixelu jeden Gausián, ovšem v obraze se mění více prvků, proto se používá směs 3 až 5 Gausiánů. V průběhu doby je nová hodnota pixelu směsi Gausiánů aktualizována online K-means aproximací.

Někdy je optimalizace parametrů pro konkrétní prostředí velice obtížný úkol, proto se vyvinula technika modelace ne-parametrického pozadí, které používají autoři [3]. Tento model pozadí může řešit situace, kdy je pozadí scény přeplněná a není zcela statické, ale obsahuje drobné pohyby, jako jsou pohyby větve stromů a keřů. Model odhaduje pravděpodobnost, jakou intenzitní hodnotu bude mít pixel na základě statistického chování v čase. Tento model se rychle přizpůsobuje na změny scény, čímž umožňuje velmi citlivou detekci pohyblivých cílů. Bere v úvahu statistické chování

určitých prvků v obraze pro rozlišení pohyblivého popředí od statického pozadí. Ovšem tato metoda je výpočetně velice náročná.

V tomto článku autoři [4] navrhnou nový přístup odečítání pozadí pro přesnou detekci pohybujících se objektů, kterou nazývají časové odlišení. Tato metoda zahrnuje tři důležité moduly: blok alarmu, modelování pozadí a extrakce objektu. Blok alarm účinně kontroluje každý blok snímku na přítomnost pohybujícího se objektu nebo nové informace o pozadí. Toto je prováděno pomocí časového odlišení pixelů snímku a umožňuje následně zpracovávat jen ty bloky, které toto obsahují. Dále je zde modelování pozadí za účelem vytvoření vysoce kvalitního adaptivního modelu pozadí pomocí dvoustupňového trénovacího procesu a mechanismus pro rozpoznávání změn v osvětlení. Jako poslední krok procesu je extrakce objekt a převedení na binární objekt za použití vhodných prahových hodnot.

Autoři [5] používají deformaci pozadí, která na rozdíl od většiny modelů, které detekují objekty v popředí jako variabilitu intenzity pixelů na určitém místě v obraze, jsou objekty v popředí detekovány jako deformace lokalit pixelů vyplývající z pohybu na pozadí. Pozadí je modelováno jako soubor deformovaných vrstev a objekty v popředí jsou tedy definovány jako ty, které nemohou být modelovány jakýmkoliv složením těchto deformovaných vrstev.

Autoři [6] používají hierarchické pozadí, které využívá toho, že pozadí snímku je rozděleno do několika segmentů pomocí mean-shift algoritmu. To má výhodu, že pro detekci pohybu v jednotlivých segmentech můžeme nastavit různé hodnoty detektoru specifické vždy jen pro daný segment. Nejprve jsou vybrány segmenty, ve kterých je zachycen pohyb. Poté je v daném segmentu extrahován pohybující se objekt v popředí. Tím, že jsou jednotlivé pixely zpracovávány jen v dané oblasti a ne na celém snímku se velice sníží výpočetní náročnost.

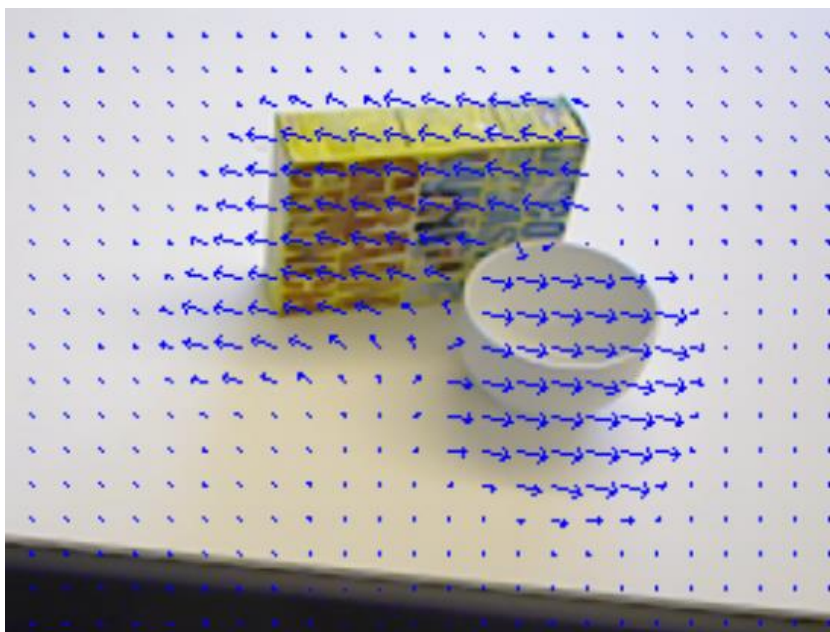
### **2.3.2 Detekce pohybujících se objektů bez modelu pozadí**

Optický tok, který používají autoři [7] je založen na detekční technice, která využívá vlastností vektorů toků pohybujících se objektů v průběhu času k detekci pohybujících se regionů v sekvenci obrazů. Příklad, jak může vypadat takovéto vektorové pole toků, je na obrázku Obr. 2.5. Na obrázku nahoře vidíme, jak se pohnuly objekty a dole na obrázku vidíme, jak vypadá výsledné vektorové pole tohoto pohybu, kde modré šipky znázorňují, jak se objekt pohnul. Tato metoda je vhodná pro velké množství pohybujících se osob v obraze a pohybující se kameru. Má ovšem velkou výpočetní náročnost a také je velice citlivá na nespojitosti v pohybu.

*Před*



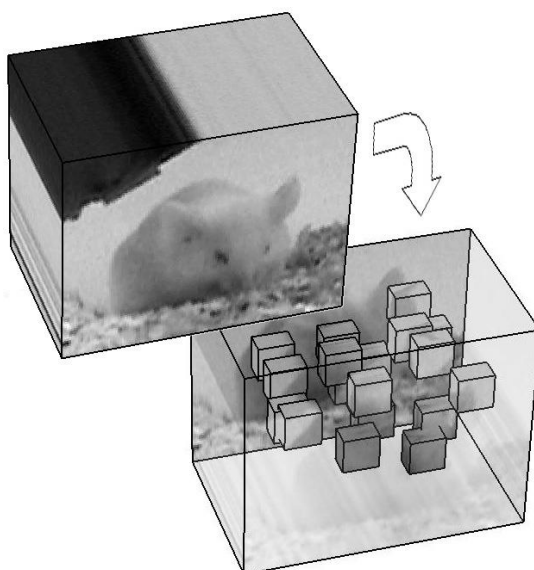
*Po pohybu*



*Obr. 2. 5 Vektorové pole pohybu*

Autoři [8] používají pro detekci pohybu, ale spíše charakteristiku chování objektů časoprostorovou filtraci, která zachycuje prostorové umístění bodů v závislosti na čase. Chování objektů se zkoumá z trojrozměrných dat vybraných dobře rozpoznatelných zájmových bodů v obraze. Ukázka, jak trojrozměrná časoprostorová dat vypadají, je na obrázku Obr. 2.6. Nahoře je kvádr časoprostorových dat celého snímku a dole jsou znázorněny jen malé kvádríky časoprostorových dat zájmových bodů. Časoprostorové metody jsou velmi vhodné pro scény s nízkým rozlišením nebo méně kvalitní videa, kde je obtížné použít jiné metody, jako je například optický tok nebo siluety. Dále jsou schopny lépe zachytit jak prostorové tak i časové informace o pohybu a chování objektů. Jejich výhodou je nízká výpočetní složitost a jednoduché provedení. Nicméně, jsou náchylné k šumu a nespojitému pohybu.





*Obr. 2. 6 Trojrozměrná reprezentace pohybu*

### 2.3.3 Detekce pádu

Pro detekci samotného pádu se nejčastěji používají vlastnosti siluety (rychlost pohybu, tvar, velikost a další.), která vzniká po odečítání pozadí od aktuálního snímku, některé metody jak odečítat pozadí jsme si popsali v předchozí kapitole, při které vznikne silueta člověka a z jejích vlastností se detekuje pád. Postup jejího vytváření je zobrazen na obrázku Obr. 2.7 (vlevo) obrázek pozadí, (uprostřed) obrázek klečící osoby, (vpravo) vzniklá silueta po odečtení prvních dvou obrázků.

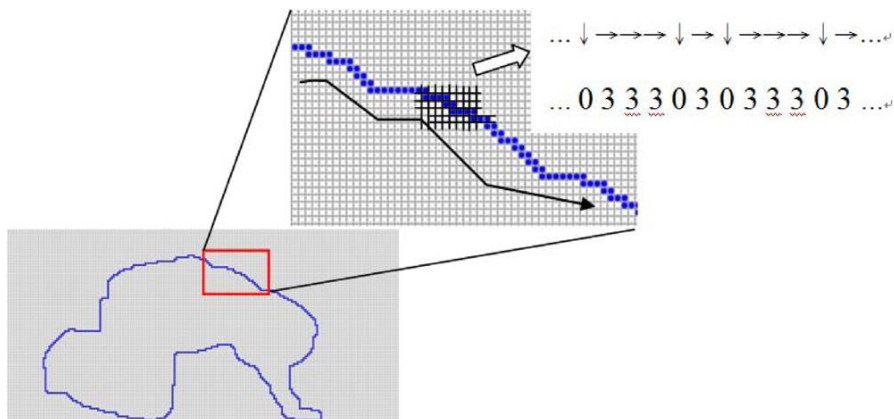


*Obr. 2. 7 Obrázek pozadí, klečící osoby v popředí a výsledné siluety*

Z obrázků vidíme, že nejprve máme pozadí, které bývá statické nebo jen velice málo se mění. Do tohoto pozadí nám vstoupí osoba, kterou chceme detekovat. Protože víme, jak vypadá pozadí bez osoby, je snadné tuto osobu detekovat a extrahovat pro další zpracování. Ovšem ne vždy nám vznikne tak pěkná silueta jako je na obrázku Obr. 3 (vpravo). Protože barva osoby může být velice podobná barvě pozadí. Silueta se převádí

na binární obrázek a upravuje pomocí morfologických operací, pro lepší vzhled a pozdější lepší určování vlastností siluety.

Autoři [9] pro zpracování siluety používají morfologické operace dilatace a eroze, aby vznikla celistvá a přesná silueta člověka. Poté se vytvoří jedno-pixelová hrana této siluety, která je převedena na číselný kód, jehož příklad je vidět na obrázku Obr. 2.8 pro její lepší strojové zpracování a uložení. Tato hrana je poté použita pro učení klasifikátoru k detekci pádu či ležícího člověka. Podle autoru tato metoda dosahuje přesnosti detekce pádu 90%, ale při použití 4-kamerového dohledu se dá přesnost zvýšit 2,1krát.



Obr. 2. 8 Hrana siluety

Autoři [10] vytváří siluetu člověka jen nahrubo. Ovšem siluety normalizují na stejnou velikost a umístění. Poté ji využívají k učení neuronových sítí. Jak je vidět na obrázku Obr. 2.9 (vlevo) ležící osoba a (vpravo) stojící osoba, pro pouhé oko nejsou siluety příliš snadno identifikovatelné, ale pro neuronové sítě jsou dostačující. Poté jejich pomocí určují, zda člověk padá a po pádu se rychle zvedá nebo trvale leží. Výsledkem z neuronové sítě je poté vektor, který jednoznačně identifikuje, zda osoba stojí, leží, klečí a další možné stavy. Nakonec podle hodnoty tohoto vektoru je detekován pád a spuštěn alarm. Zde autoři uvádí vysokou míru přesnosti detekce pádu, vyšší než 90%.



Obr. 2. 9 Obrázek ležící a kráčeující siluety

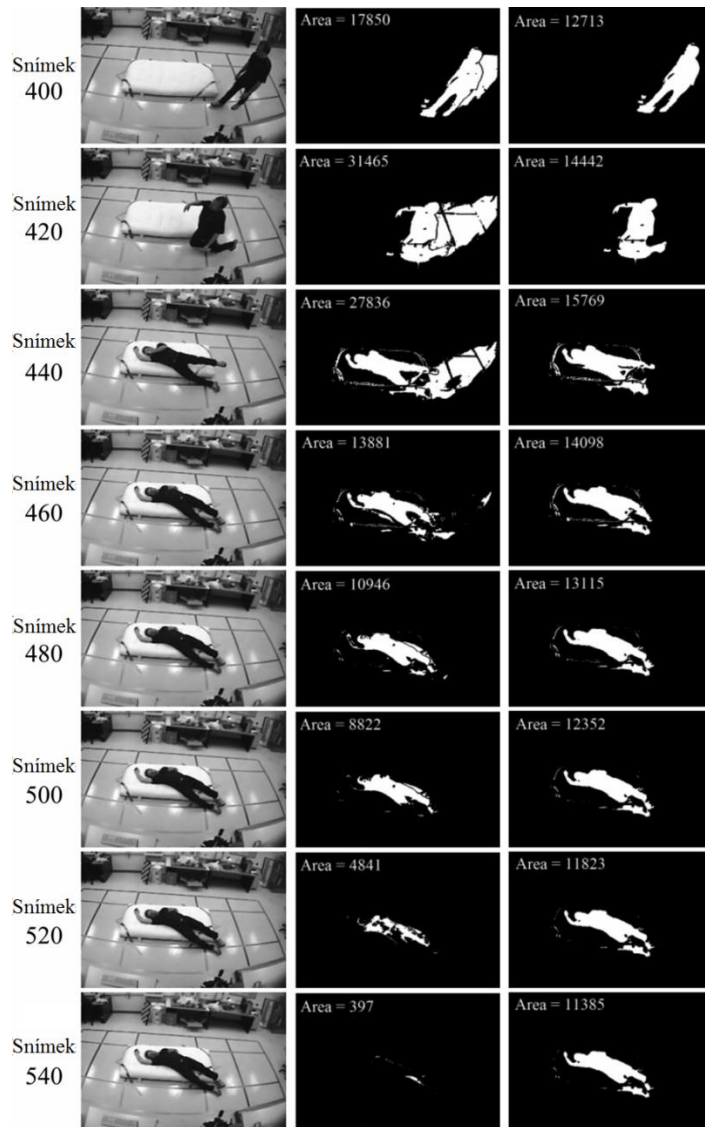
Autoři [11] si po odečtení pozadí spojí jednotlivé malé bloky pomocí čtyřbodového sousedství do velkých celistvých objektů reprezentující osobu. Tyto objekty vůbec nemusí připomínat člověka, jak je vidět na obrázku Obr. 2.10 (vpravo), protože to není potřeba. Stačí tyto objekty orámovat, jak je vidět na obrázku Obr. 2.10

(vlevo). Metoda zkoumá pouze velikost, natočení a poměr stran jednotlivých rámečků těchto objektů. Podle toho rozhoduje, zda se jedná o stojící, ležící postavu nebo něco mezi tím. Autoři článku uvádí úspěšnost detekce pádu 82,2 %.



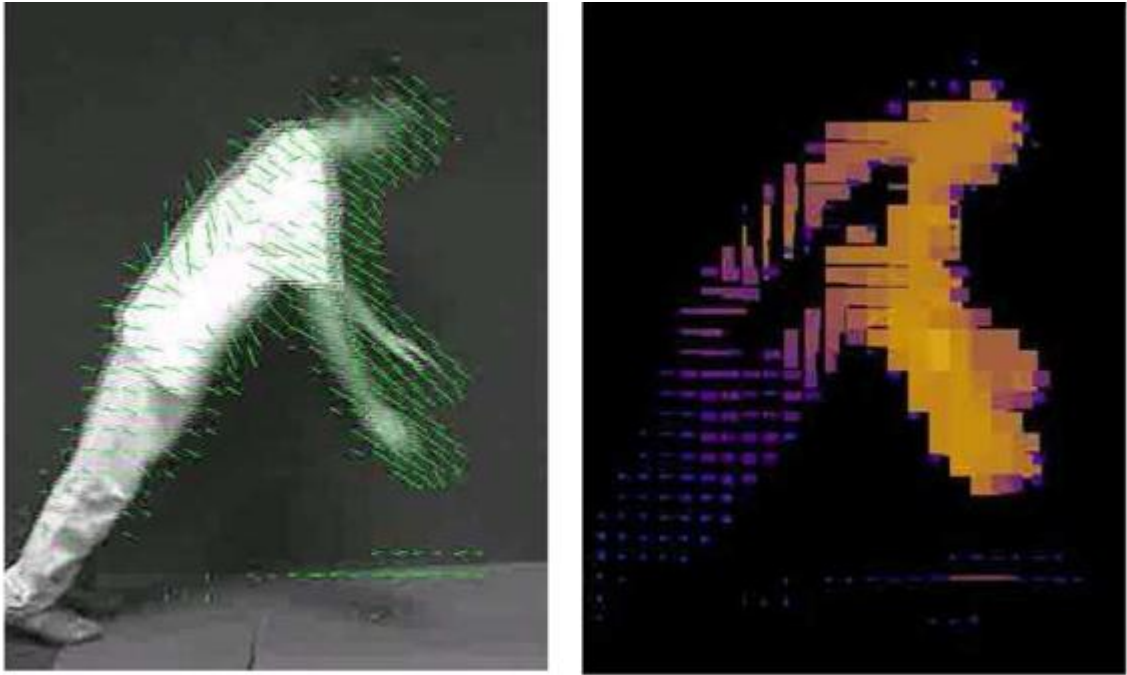
*Obr. 2. 10 Označená osoba a její hrubá silueta*

Autoři [12] problém detekce řeší přes dva různé přístupy k aktualizaci pozadí. Aktualizace pozadí, které se v průběhu sledování mění, je důležitou součástí detekce osob. Existují dva přístupy: první přístup, zobrazen na obrázku Obr. 2.11 (vpravo), aktualizuje pouze tu část, kde se nenachází silueta tedy, že silueta z obrazu nemizí a poté je její klasifikace, zda se jedná o pád, velice podobná jako v předchozí metodě, tedy z poměru stran a orientace siluety. Druhý přístup, zobrazen na obrázku Obr. 2.11 (uprostřed), aktualizuje celý snímek, tedy pokud se osoba nehýbe, tak silueta ze snímku pomalu mizí. Proto se počítá počet pixelů označujících siluetu, a pokud silueta pixely ubývají, až nakonec zmizí, signalizuje to, že člověk upadl a dál nehnutě leží. Problémem aktualizace pozadí se budu podrobně zabývat v kapitole 5.3. Podle autorů článku dosáhli touto metodou chybovosti pouze 4,8%.



*Obr. 2. 11 Siluety při různém přístupu aktualizace pozadí, (vpravo reálný snímek, uprostřed aktualizace celého pozadí, vpravo aktualizace jen pozadí, kde se nenachází silueta)*

Jiný přístup zvolili autoři [13], kteří používají metodu MVFI (Motion Vector Flow Instance), kde zobrazují časoprostorová data snímku, tedy rychlost a směr jednotlivých pixelů pro každý snímek, protože podle studie je pád 2-3 rychlejší než běžné denní činnosti dá se pohyb snadno rozpoznat a detekovat. Výsledný snímek po aplikaci metody a výpočtu časoprostorových dat je zobrazen na obrázku Obr. 2.12 (vlevo) zobrazení pohybových vektorů v obrázku, (vpravo) žlutě jsou vybarveny nejrychleji se pohybující části obrázku. U této metody autoři uvádí úspěšnost detekce pádu dokonce 99%.



*Obr. 2. 12 Obrázek znázorňující množství pohybu v obraze pomocí vektorového pole a barevnosti*

### 3 Cíle práce

Cílem této práce je provést studii stavů a událostí, které je ve vnitřním prostředí možno detekovat na základě analýzy záznamu z kamerového dohledového systému. Cílem je detekce obvyklých a neobvyklých událostí - hlavně detekce pádu osoby či trvale ležící osoby v místech nezvyklých. Zaměřím se hlavně na detekci pádu jen pomocí kamerového dohledu, protože jak již bylo uvedeno v předešlé kapitole, tato problematika je velice aktuální a žádaná. Existuje více přístupů jak tuto problematiku řešit a já se pokusím z nich vybrat pro moje účely ten nejvhodnější a ten dále upravovat, abych vytvořil vlastní, spolehlivě fungující detekční algoritmus. Nejprve však také otestuji již hotové funkce pro detekci osob v Matlab Vision Toolboxu, abych věděl, jakých dosahují výsledku a zda bych je mohl využít v mém algoritmu, který se pokusím navrhnout na základě nabytých vědomostí z metod představených v kapitole 2.4 a doporučené literatury [15,16,17].

Naimplementovaný software můžu testovat přímo v Centrum asistivních technologií (CAT) ČVUT. Jde o moderní centrum, které slouží pro výuku v oblastech zahrnujících péči o potřebné, o osoby nemocné, invalidní či seniory nebo osoby jinak handicapované, včetně dětí. Součástí prostor centra je také inteligentní byt, ve kterém bude testován můj software. Byt je plně vybaven, jako běžný byt, ve kterém se předpokládá, že bude tento software nasazen. Každá místnost je vybavena také IP kamerou pro kamerový dohled.

## 4 Teoretická část

### 4.1 Použité metody zpracování obrazu

V této části se budeme teoreticky zabývat funkcemi, které budou později použity v algoritmu, popíšeme si jejich funkci a matematicky je znázorníme.

#### 4.1.1 Gaussovský filtr

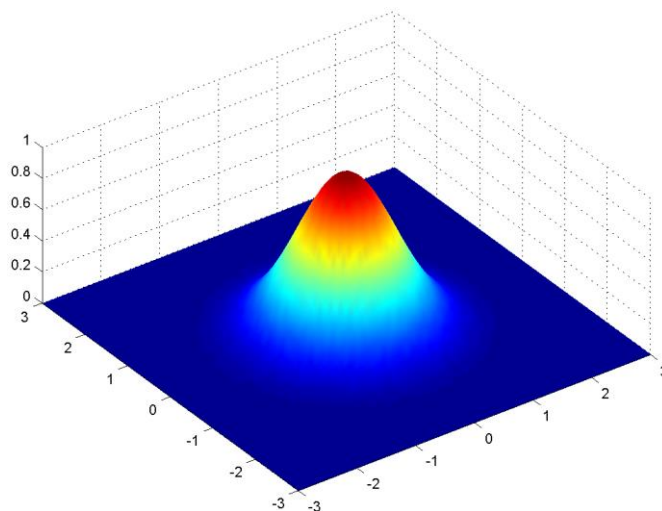
Gaussovský filtr [16,17,18] je pravděpodobně nejpoužívanější filtr na odstraňování šumu v obraze. Gaussovským filtrováním obrazu  $f$  definovaného obrazovou funkcí  $f: R^n \rightarrow R$  rozumíme konvoluci  $f$  gausiánem. Výsledkem je obraz  $g$  definovaný takto (s explicitní normalizací jádra):

$$g(x_0) = \frac{\int_{R^n} f(x)G(x-x_0,\sigma)dx}{\int_{R^n} G(x-x_0,\sigma)dx} \quad (4.1)$$

Kde Gausián  $G$  je funkce ve tvaru:

$$G(x, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{x^T x}{\sigma^2}} \quad (4.2)$$

Tvar Gausiánu je vidět na obrázku 4.1. Fourierova analýza Gaussovského filtru ukazuje, že jeho funkce spočívá v utlumení vyšších frekvencí obrazové informace. To je naprosto legitimní způsob odstraňování šumu, který fyzici používají odedávna. Na rozdíl od fyzikálních aplikací se tento postup ale pro filtrování šumu z video snímků příliš nehodí z jednoho prostého důvodu: Ve fyzikálních měřeních se často nachází zajímavá informace v úplně jiném pásmu než šum, a utlumení zašuměných frekvencí tedy informaci nijak neznehodnotí. To ale pro snímky neplatí, protože ve vyšších frekvencích, ve kterých se obrazový šum nachází, je zároveň obsažena detailní kresba snímku. To má za následek, že šum se odstraní, ale z hran se stanou pozvolné přechody. To je způsobeno tím, že statický lineární charakter filtrovacího mechanismu způsobuje, že do výsledné barvy je započítána barva z okolních pixelů, které přitom patří do jiného regionu. Toto ovšem v mém algoritmu nevadí, protože já počítám rozdíl dvou snímků a nepotřebuji ostré snímky s detailními hranami.



Obr. 4. 1 Gaussián

## 4.1.2 Matematická morfologie

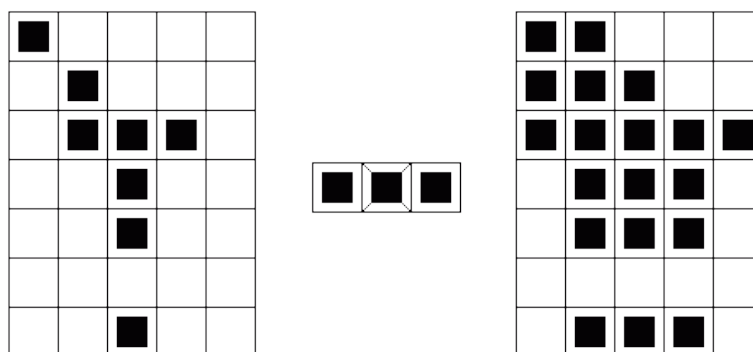
Matematická morfologie [14,15] vychází z vlastností bodových množin. Nejčastěji se aplikuje na binární obrazy. Morfologické operace se používají pro předzpracování obrazu (odstranění šumu, zjednodušení tvaru objektů), zdůraznění struktury objektů (kostra, ztenčování, zesilování, konvexní obal, označování objektů) a pro popis objektů číselnými charakteristikami (plocha, obvod, projekce, atd.).

Morfologické operace jsou realizovány jako relace obrazu (bodové množiny  $X$ ) s jinou menší bodovou množinou  $B$ , tzv. strukturním elementem. Morfologickou transformaci si lze představit jako určitý systematický pohyb strukturního elementu  $B$  po obrazu  $X$  a vyhodnocení odezvy podle typu operace. Mezi dvě základní morfologické operace patří dilatace a eroze. Dilatace skládá body dvou množin pomocí vektorového součtu  $u$  (též Minkowského množinový součet, např.  $(a, b) + (c, d) = (a + c, b + d)$ ). Objekty v obrazu jsou po aplikaci dilatace zvětšené o jednu "slupku" na úkor pozadí. Dilatace se používá k zaplnění děr popř. zálivů a její definiční vztah je dán následujícím vzorcem v rovnici (4.3).

$$X \oplus B = \{d \in E^2: d = x + b, x \in X, b \in B\} \quad X \oplus B = \bigcup_{b \in B} X_b \quad (4.3)$$

Následující obrázek 4.2 znázorňuje výsledek operace dilatace (vpravo) vzniklý z původního obrazu (vlevo) pohybem strukturního elementu (uprostřed).



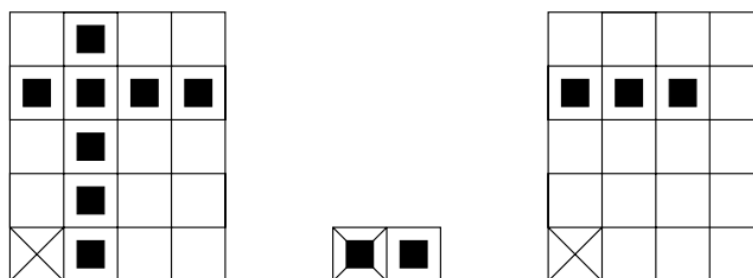


Obr. 4. 2 Příklad dilatace

Druhou základní morfologickou operací je eroze. Eroze skládá dvě bodové množiny s využitím vektorového rozdílu. Je duální (nikoliv inverzní) transformací k dilataci a používá se pro zjednodušení struktury objektů. Objekty jednotkové tloušťky (relativní jednotka) zmizí a složité objekty spojené čarami jednotkové tloušťky se rozloží na několik jednodušších objektů. Eroze je dána následujícím vztahem v rovnici (4.4).

$$X \ominus B = \{d \in E^2: d + b \in X \text{ pro } \forall b \in B\} \quad X \ominus B = \bigcap_{b \in B} X_{-b} \quad (4.4)$$

Předchozí vztah říká, že pro každý bod obrazu  $d$  se ověřuje, zda pro všechna možná  $d + b$  leží výsledek v  $X$ . Pokud ano, zapíše se v reprezentativním bodě do výsledného obrázku 1 a v opačném případě 0. Následující obrázek 4.3 znázorňuje výsledek operace dilatace (vpravo) vzniklý z původního obrazu (vlevo) pohybem strukturního elementu (uprostřed).



Obr. 4. 3 Příklad eroze

Další morfologické operace jsou otevření a uzavření a jsou to operace, které vzniknou vzájemnou kombinací elementárních operací dilatace a eroze. Výsledkem obou je zjednodušený obraz, který obsahuje méně detailů (odstraní detaily menší, než strukturní element, celkový tvar objektu se ale neporuší). Eroze následovaná dilatací se nazývá morfologické otevření (rovnice (4.5) vlevo), které budu používat v mém algoritmu. Otevření oddělí objekty spojené úzkou šíjí a tak zjednoduší strukturu objektů.

Dilatace následovaná erozí se naopak nazývá morfologické uzavření (rovnice (4.5) vpravo). Spojí objekty, které jsou blízko u sebe, zaplní díry a vyhladí obrys.

$$X \circ B = (X \ominus B) \oplus B \quad X \bullet B = (X \oplus B) \ominus B \quad (4.5)$$

## 4.2 Studie stavu a událostí

V dnešní době existují firmy, které nabízí velké množství aplikací s využitím kamerového dohledu. Jenda z nich je i firma NetRex [19], která nabízí tyto produkty:

**Detekce manipulace s kamerou** – tato aplikace upozorní systém nebo obsluhu na stav, kdy je, ať už vandalismem nebo náhodou narušen provoz kamery, aby se ochránila kamera a dohled byl stále funkční.

**Video detekce pohybu** – často je využívána pro upozornění obsluhy na pohyb ve sledované lokalitě. Například vstupu osob do serverovny nebo další střežených míst.

**Počítání lidí** – v dnešní době představuje systém přesného počítání návštěvníků ve spojení s výstupy dat z pokladních systémů nejúčinnější nástroj moderního řízení efektivnosti prodeje v prodejních řetězcích.

**Detekce překročení linie** – tato aplikace umožní definovat virtuální linie. Při překročení těchto linií pohybujícím se předmětem kamera a systém spustí definované akce. Například vyvolá poplach, odešle e-mail, SMS, rozsvítí světlo, zamkne dveře atd. Ideální je tato aplikace pro sledování vchodů a východů, nakládacích ramp, monitoring budov nebo parkovišť.

**Dohled nad pokladnou prodejny** – velmi žádanou aplikací je nyní inteligentní kamerový dohled pokladny. Integrace kamery s pokladním systémem vám umožní získat detailní přehled nad jakoukoliv činností prováděnou na pokladně. Díky tomu je možné velmi rychle získat například denní přehled všech provedených vratek, nedokončených transakcí, otevření pokladních zásuvek bez předchozího načtení zboží a jiných rizikových událostí.

Jak uvádí studie [1], kamerový dohled má mnohé další využití, jako například: **Charakterizace lidské chůze**, která je potřeba, aby se zjistilo, zda je daný člověk zdravý, zda nekulhá nebo nemá jiný zdravotní problém, také může sloužit jako pomocný nástroj u rozpoznávání pohlaví osob.

**Počítání lidí**, které je dobré, jak bylo uvedeno výše pro marketing, ale také z bezpečnostních důvodů, zda není v bytě nežádoucí osoba navíc nebo k zjišťování zda ke sledované osobě někdo chodí na návštěvy nebo žije úplně sama.

**Identifikace a sledování člověka**, které se využívá například na sportovištích pro sledování sportovců, například i určení, kde se pohybovali a kolik toho naběhali, toto se ovšem i velice hodí při sledování starších a nemocných lidí, k určení, zda toho každý den

nachodí zhruba stejně nebo jak se vyvíjí jejich fyzická zdatnost, zda se neprojevuje například počátky nemoci.

**Klasifikace pohlaví**, které se provádí na základě odlišnosti ve fyzické stavě těla či obličejové a může také sloužit k marketingu, aby se vědělo jako je podíl zastoupení v návštěvnosti mužů a žen.

**Detekce pádu starších lidí**, která je jednou z hlavních aplikací v dohledových videozáznamech, protože může velice chránit zdravé osob, pokud se k nim dostane pomoc včas. Hlavním rysem klasifikace je vektor rychlost pádu, který rozlišuje mezi skutečnou pádovou událostí a události, kde si člověk jen lehnul bez pádu.

Podle studie [9] je potřeba detekovat mnoho podobných stavu ve vnitřních prostorech, hlavně na schodištích a dlouhých chodbách, které jsou nebezpečně pro starší a nemocné lidi. Je potřeba odlišit nepádové a tudíž bezpečné stavy jako schůze, běh, sezení nebo vstávání od těch pádových jako uklouznutí, zakopnutí nebo omdlení. Samotný pád je detekován, pokud se spadlá osoba nedokáže sama zvednout do 10s. Toto všechno dokáže jejich naučený klasifikátor právě na tyto možné stavy.

Ve studii [10] učí neuronovou síť na další možnosti pohybu lidí v místnosti, jako jsou například klečení, ohýbání nebo dokonce modlení, které jsou velmi podobné pádu, ale dají se odlišit dobře naučenými vzorky.

Ve studii [8] pomocí trojrozměrných dat zájmových bodů rozpoznávají obličejové výrazy, myší chování a lidské aktivity. Je potřeba vždy naučit klasifikátor na dané činnosti, které má poté rozpoznat. Toto je velice užitečné pro dohled nad staršími a nemocnými lidmi, protože je dobré vědět, jaké činnosti zvládají sami a které už nedělají a z toho určovat jejich fyzickou zdatnost a duševní zdraví.

Po provedení studie, je vidět, že je velké množství stavů, které se dají detekovat pomocí dohledových videozáznamů. Ale jak už bylo psáno výše, poslední dobou je hlavní snahou detekovat pádové události, protože je to velice žádoucí u dohledu v bytech starších a nemocných osob, které žijí samy a proto je potřeba, aby na ně bylo dohlíženo a v případě nebezpečné situace byla včas přivolána pomoc. Proto se i já v této diplomové práci zaměřím hlavně na detekci pádových událostí.

## **4.3 Problematika dohledových videí v reálných podmínkách**

Nevýhodou při práci s videem v reálných podmínkách na rozdíl od simulovaných ideálních podmínek je to, že ve snímku je velké množství objektů a ne jen silueta osoby. Problém dělají odrazivé plochy jako skla a zrcadla, kde se odráží obraz člověka a algoritmus pro detekci pohybujících se objektů ho zaznamenává. Dá se odstranit tím, že za člověka považujeme jen objekt s určitými vlastnostmi (velikost, orientace, doba objektu na scéně). Ovšem může se stát, že i odraz má podobné vlastnosti jako člověk a

nedá se tedy rozeznat. Proto používám dvou stupňovou detekci pádu, kterou se budu zabývat níže, aby tyto odrazy nespouštěly pádové alarmy.

Dalším problémem je málo kontrastní rozdíl mezi barvou oblečení člověka a pozadím. Třeba když člověk v tmavém oblečení stojí naproti tmavým dveřím. Tehdy se může stát, že vzniklá silueta nekopíruje obrys člověka, ale jen jeho část. I s tím se musí počítat, že siluety nebudou mít během celého sledování stejnou velikost, ale to nevadí, pohyb těžišť, který je důležitý v mém algoritmu detekce pádu se dá sledovat i u malých objektů. Problém nastává, pokud silueta zmizí úplně, v takovém případě se nedá detekovat pád. Proto by práh pro vytvoření siluety měl být, co nejnižší, aby k takové situaci docházelo opravdu jen ojediněle.

Pohyb více osob na scéně je také problém, protože pokud dva lidé prochází blízko sebe, sloučí se do jedné siluety a naopak poté, co se opět vzdálí, jsou siluety opět rozděleny. Protože je potřeba sledovat každou osobou v obraze po celou dobu jejího výskytu na scéně, aby se mohl detekovat případný pád, je nutné umět rozpoznat osoby během sloučení i během rozdělení siluet. Tento problém podrobně řeším v kapitole 5.6. Je také potřeba člověka v obraze detekovat od samého počátku jeho výskytu. Tedy, i když je v obraze jen jeho malá část, tedy malé objekty na okraji obrazu se musí také považovat za člověka, protože by to mohl být člověk vstupující do obrazu.

Posledním velkým problémem jsou možné falešné detekce pádu, ať už chybou v párování objektů nebo se člověk mohl jen pro něco shýbnout k zemi a další podobné případy. Není žádoucí, aby tyto případy hned vyvolával poplach. Proto je potřeba pád detekovat ve dvou stupních. První je možný pád, který může nastávat často některým z uvedených případů. Ale nezpustí poplach, ale spustí druhý stupeň detekce, který začne zkoumat vlastnosti tohoto možného pádu, jako jestli se objekt dále hýbe nebo jestli jen nehybně leží. U výše zmíněných případů falešné detekce se objekt bude i nadále hýbat a proto algoritmus vyhodnotí, že se o pád nejedná a nebude spouštět poplach, ale pokud by se jednalo o skutečný pád, objekt zůstane nehybně ležet, podle článku [9] po dobu 10 sekund, a algoritmus ho vyhodnotí jako skutečný pád a spustí poplach.

Všechny tyto problémy bylo potřeba vyřešit v mém algoritmu a jejich přesné řešení a obrázky s výsledky jsou popsány a znázorněny v následující kapitole.

## 5 Praktická část

Abych mohl vůbec začít psát vlastní software, bylo potřeba natočit videa, na kterých by se dal software testovat. Pro tyto účely mám k dispozici plně vybavené Centru asistivních technologií přímo v budově školy ČVUT FEL na Dejvicích. Tam se nachází plně vybavený byt s kamerami, ke kterým se dám vzdáleně připojit a začít kdykoliv natáčet. Nejprve jsem natočil videa, kde se pouze pohybovali lidé bez pádu, abych otestoval, která metoda detekci pohybujících se objektů (lidí) bude pro můj software nejvhodnější. Když už jsem správně detekoval pohybující se objekty, natočil jsem i videa, na kterých simuluji pád, abych mohl otestovat i mnou navrhnutý detekční algoritmus pádu.

Nejprve jsem natočil několik demonstračních videí, které jsou natočeny tak, aby co nejvíce usnadnily práci našeho detektoru a demonstrovali vysokou spolehlivost našeho softwaru v ideálních podmínkách. Tedy tmavě oblečení lidé na bílém pozadí, žádný jiný pohyb v obraze krom lidí a žádné zrcadlivé plochy, které by odrážely obraz člověka. Aby bylo vidět, jak dobře dokáže algoritmus pracovat za ideálních, ač ne příliš reálných podmínek. Toto není cílem méj práce, já chci, aby můj algoritmus mohl být nasazen v reálných podmínkách, kde se musíme potýkat s mnohými „nevýhodami“, kterými se málokdo zabývá.

Proto jsem také natočil videa, která by, co nejrealističtěji znázorňovaly scénu v běžném pozorovaném bytě. To ovšem přinášelo řadu již zmíněných „nevýhod“. Velkým problémem jsou zrcadlivé plochy. V mém videu je to velké sklo na pozadí, kde se objevuje odraz člověka a tím mate detektor, který jej zaznamenává a bere ho jako pohybující se objekt, i když tam žádný není. Dalším problémem jsou stejně barevné plochy na pozadí, jako je barva oblečení člověka. Na těchto plochách naopak značná část člověka splývá s pozadím a detektor ho velice špatně detekuje, detekuje jen malou část člověka nebo se může stát, že ho ztratí úplně. Ovšem já jsem chtěl nasimulovat všechny možné nepříznivé vlivy, se kterými se může můj software setkat, aby byl co nejvíce robustní.

### 5.1 Matlab Vision Toolbox

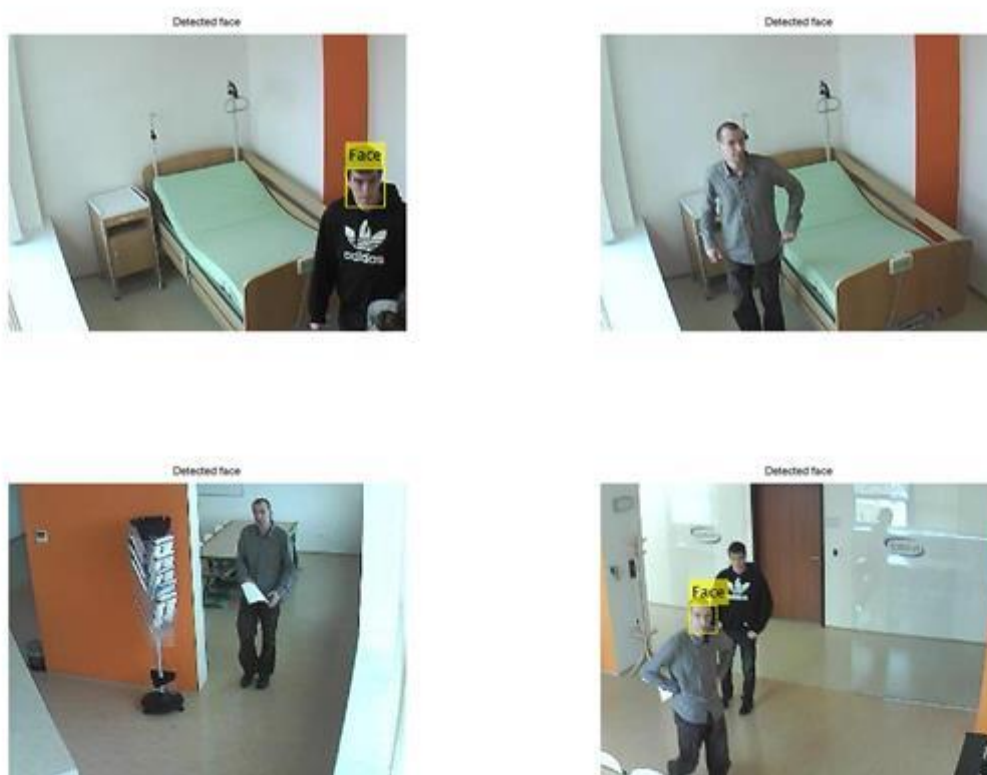
Nejprve jsem otestoval funkce, které nabízí samotný Matlab, abych určil, zda vůbec, případně jakým způsobem bych je mohl použít v mém algoritmu. Otestoval jsem 3 funkce, které jsou nadefinovány v Matlabu pod jeho vlastní vision knihovnou:

```
vision.CascadeObjectDetector(),  
vision.PeopleDetector(),  
vision.ForegroundDetector().
```

Po prvotním otestování v základním nastavení tyto funkce nedosahovaly požadované přesnosti. Proto jsem zkusil nastavovat různé jejich parametry, abych viděl jejich vliv na přesnost detekce v mém konkrétním prostředí a zjistil jsem, zda lze dosáhnout lepších výsledků a daly by se tyto funkce použít v mém algoritmu. Pro testování jsem použil pár snímků z kamer nainstalovaných ve vzorovém bytě.

*vision.CascadeObjectDetector():*

Tato funkce ve výchozím nastavení detekuje obličeje otočené směrem ke kameře. Na našich snímcích měla tato funkce úspěšnosti necelých 30%. Výsledky jsou vidět na obrázcích:



*Obr. 5. 1 Základní detekce obličejů*

Poté jsem zkusil detekci obličeje z profilu, protože člověk není vždy otočen čelně přímo ke kameře, ovšem toto nastavení nezaznamenalo žádnou úspěšnou detekci. Další nastavení byl Scale Factor, neboli hodnotu, jak může být zvětšený či zmenšený obličej na snímku oproti obličejům, na kterých je tento klasifikátor naučen. Aby funkce našla více obličejů a ne jen ty, které jsou stejně velké jako obličeje použité při učení klasifikátoru. Tento parametr jsem postupně nastavoval z výchozího nastavení 1,1 až na hodnotu 1,01. Což sice přineslo o pár případů více detekcí obličejů, ale také velké množství nesprávně detekovaných míst, jak je vidět na testovaných snímcích na obrázcích:



*Obr. 5. 2 Detekce obličejů se změněným Scale Factorem*

. Posledním parametrem, který jsem měnil, byl Merge Threshold, snížením z výchozí hodnoty 4 postupně až na hodnotu 1. Tato hodnota určuje, které objekty budou označeny za správné při výskytu více potenciálně správných objektů. Zajímalo mě, jaký bude poměr více správně detekovaných obličejů proti nárůstu nesprávně detekovaných obličejů. Objevilo se pár správně detekovaných obličejů navíc, ale mnohem více nesprávně detekovaných, jak je vidět na testovaných snímcích na obrázcích:





Obr. 5. 3 Detekce obličejů se změněným Merge Thresholdem

Ukázalo se, při změně parametrů dochází k velkému množství nesprávných detekcí a jen velice malému nárůstu správných detekcí a ve výchozím nastavení, nedochází k dostatečně spolehlivé detekci, proto tato funkce není vhodná pro moje zadání.

*vision.PeopleDetector():*

Tato funkce detekuje vzpřímené postavy. Ve výchozím nastavení dosahovala poměrně slušných výsledků, jak je vidět na testovaných snímcích na obrázkách:



Obr. 5. 4 Základní detekce vzpřímené postavy



Proto se dalo očekávat, že po změně nastavení budou výsledky ještě lepší. Jako první nastavení parametru, které jsem změnil, byla velikost obrázků použitých pro učení tohoto detektoru ze 128x64 na 96x48. Protože na testovacích snímcích se vyskytují i relativně malé postavy. Proto jsem touto změnou očekával více detekovaných postav. Tato změna opět přidala pár správně detekovaných postav navíc, ale velké množství nesprávně detekovaných, jak je vidět na testovaných snímcích na obrázkách:



*Obr. 5. 5 Detekce vzpřímené postavy se změněnou velikostí učících obrázků*

Další parametr byl Classification Threshold, který jsem změnil z výchozí hodnoty 1 postupně až na hodnotu 0,1. Tato hodnota určuje, které objekty budou označeny za správné při výskytu více potenciálně správných objektů. Opět mě zajímal, jaký bude poměr více správně detekovaných postav proti nárůstu nesprávně detekovaných postav. Opět se objevilo pár správně detekovaných postav navíc, ale i několik nesprávně detekovaných, jak je vidět na testovaných snímcích na obrázkách:



Obr. 5. 6 Detekce vzpřímené postavy se změnou velikostí Classification Thresholdu

V mých podmínkách se ukázalo nemožné nastavit parametry pro přesnou detekci, proto budu tuto metodu používat jen k zpřesnění mého algoritmu k určení, který objekt je člověk.

*vision.ForegroundDetector()*:

Tato funkce detekuje pohybující se objekty. Bohužel tento detektor potřebuje velice plynulou sekvenci obrázků, nejlépe pak rovnou video a absolutně statické pozadí. Jinak se objeví velké množství falešně detekovaných cílů. Takovéto podmínky jsem v mojí úloze nedokázal zajistit, a proto jej v mém algoritmu nepoužiji.

### 5.1.1 Shrnutí výsledků Matlab Vision Toolboxu

Závěrem jsem usoudil, že se tyto funkce pro můj algoritmus dají použít jen pro zpřesnění. Protože detekce obličeje nefungovala spolehlivě ani s různým nastavením. Zřejmě to je způsobeno špatným rozlišením kamery, které znemožňuje funkci rozpoznávat spolehlivě obličeje. Detekce vzpřímených osob také nefungovala dostatečně přesně pro můj algoritmus. V mém algoritmu si nemůžu dovolit, aby se stávalo, jako u této funkce, že některé osoby nebudou detekovány, to by mohlo mít vážné následky. Při

změně různých parametrů nastavení objevovalo velké množství falešných cílů a stejně nebyly nalezeny všechny osoby. Funkce detekce pohybujících se objektů fungovala dobře, ale jen kvalitním obraze a s absolutně statickým pozadím. Toto ovšem s levnými IP kamerami a lehce se měnícím pozadím (osvětlení, stíny) nemůžu zaručit a proto z nedostatku robustnosti ani tato metoda není vhodná. Proto jsem se rozhodl napsat si vlastní metodu na detekci pohybujících se objektů a tu dále rozvíjet pro potřeby mého algoritmu. Metody Matlabu použiji jen pro zpřesnění mého algoritmu například pokud Matlab detekuje vzpřímenou postavu, vím, že se jedná o člověka a můžu toho v mém algoritmu využít. Nebo pokud Matlab detekuje obličej, vím, že se nejedná jen o stín nebo odraz podobných parametrů jako člověk, ale člověk to skutečně je.

## 5.2 Algoritmus

Algoritmus budu vyvíjet za pomoci některých metod popsanych v kapitolách 2.3 a 2.4. Rozhodl jsem se pro metodu odečítání aktuálního snímku od pozadí a vytvoření siluety reprezentující pohybující se objekt. Tato metoda je snadná na implementaci a není tak výpočetně náročná jako např. metoda MVFI (Motion Vector Flow Instance). Při vytváření siluety se dají nastavit různé parametry (o kterých budu psát níže v popisu samotného algoritmu) a proto je tato metoda dost robustní, aby se dala použít pro celodenní sledování s proměnlivými světelnými podmínkami a lehce se měnící scénou. S výslednou siluetou se dá dále pracovat, aby vypadala tak, jak potřebuji a mohla tak sloužit k detekci pádu.

Samotný algoritmus se skládá z mnoha bloků, jak je vidět na diagramu (Diagram 1). Nejprve se inicializují všechny potřebné konstanty potřebné pro běh programu a jednotlivé funkce. Vyčte se jeden snímek z videa, protože kamery nemají konstantní počet snímku za vteřinu, občas se stává, že pár snímku po sobě jdoucích jsou stejné. Proto kontroluje, zda jsou dva po sobě jdoucí snímky rozdílné, pokud ano pokračuje dál, pokud ne vezme další snímek v pořadí. Detekuje osoby pomocí funkce z Matlab Vision Toolboxu, aby mohl jím detekované pohybující se objekty s jistotou označit za člověka. Spočítá průměrný rozdíl intenzity pixelů v po sobě jdoucích lišících se snímcích a podle nich nastaví konstanty pro aktualizaci pozadí a převodu na binární obrázek. Aktualizuje pozadí. Odečte aktuální snímek od pozadí, převede ho na binární obrázek, čímž získá siluetu. Z této siluety vybere ty části, který považuje za člověka a tyto části siluety upraví. Takto vzniklé siluety spáruje mezi jednotlivými snímky. Nyní detekuje pád. A pokud je na konci videa, vytvoří ze vzniklých snímků výsledné přehledové video, které slouží k prezentaci úspěšnosti algoritmu. Pokud ještě není na konci videa, pokračuje opět dále stejnými kroky. Jednotlivé důležité bloky si rozebereme podrobněji v následujících kapitolách:

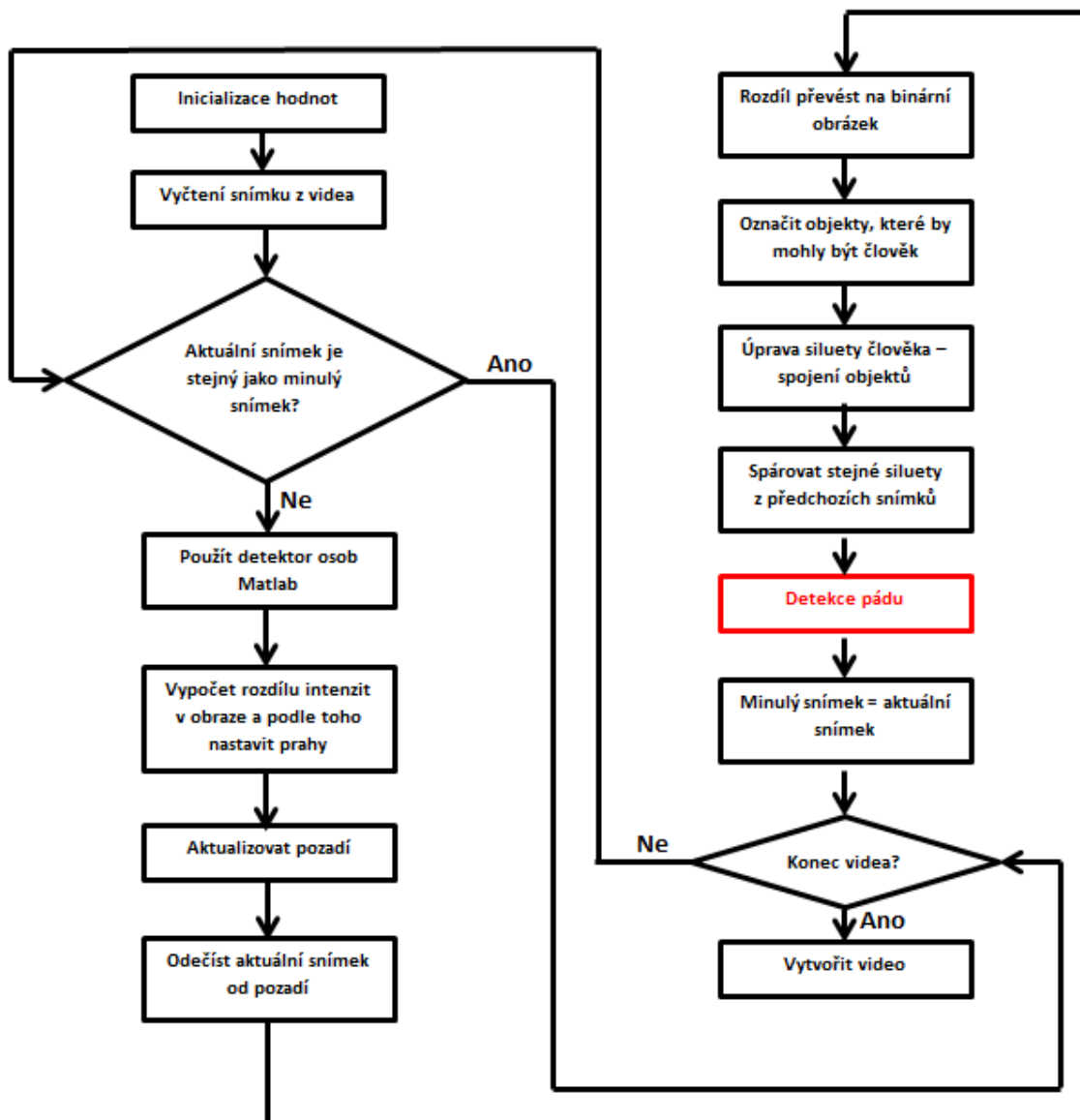


Diagram 1 Struktura algoritmu

### 5.3 Aktualizace pozadí

Nyní se podíváme na aktualizaci pozadí. Protože pozorovaná scéna se neustále mění, ať už změnou osvětlení během dne, či drobnou změnou interiéru sledovaného bytu. Jak už bylo uvedeno v kapitole 2.4, existují dva přístupy aktualizace pozadí. Aktualizace jen té části snímku, kde se nenachází silueta, jak je uvedeno ve vzorci 5.1:

$$P_{j+1} = M \cdot S + (1 - M)[(1 - \alpha) \cdot S + \alpha \cdot P_j] \quad (5.1)$$

Kde  $P_{j+1}$  je nově vzniklé pozadí po aktualizaci,  $M$  je binární obrázek, kde váha 1 je pro místa se siluetou a 0 pro zbytek snímku,  $P_j$  je staré pozadí,  $S$  je aktuální snímek a  $\alpha$  určuje rychlost aktualizace pozadí. Toto řešení mělo za následek, že chybně označené

objekty postupem času nemizely a znehodnocovaly tak výsledky algoritmu. Proto jsem zvolil aktualizaci celého pozadí i v místech, kde se nachází silueta člověka. Nejprve jsem vyzkoušel aktualizace pozadí průměrováním několika po sobě jdoucích snímků z kamery. Ovšem při malém počtu průměrovaných snímků docházelo k aktualizaci pozadí příliš rychle a chvíli nehybně stojící či ležící objekty se ztrácely a to nebylo žádoucí. Mohlo by se to vyřešit větším počtem průměrovaných snímků, což ale bylo příliš náročné na výpočetní čas a paměť. Proto jsem zvolil jako aktualizaci pozadí exponenciální zapomínání. Neboli že aktuální pozadí vždy nahradím určitým procentem aktuálního snímku, jak je uvedeno ve vzorci 5.2:

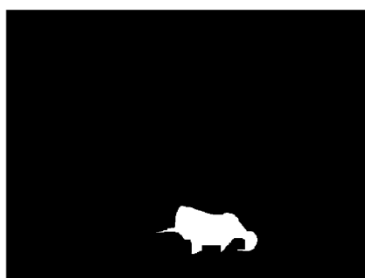
$$P_{j+1} = (1 - \alpha) \cdot S + \alpha \cdot P_j \quad (5.2)$$

Kde opět  $P_{j+1}$  je nově vzniklé pozadí po aktualizaci,  $P_j$  je staré pozadí,  $S$  je aktuální snímek a  $\alpha$  určuje rychlost aktualizace pozadí. Tuto konstantu je třeba měnit v závislosti na aktuálním dění před kamerou. Protože pokud je tam velký pohyb je potřeba rychle aktualizovat, aby za pohybujícím se člověkem nezůstávaly „duchové“. Naopak pokud na scéně není téměř žádný pohyb je potřeba aktualizovat, co nejméně, aby se nám neztrácely nepohybující se osoby. Proto jsem tuto konstantu volil 0,9-0,99 v závislosti na velikosti pohybu před kamerou, který jsem určil z průměrné změny barevnosti jednotlivých pixelů v po sobě jdoucích snímcích, velká změna znamená, že se objekt pohnul výrazně a proto se práh aktualizace nastaví na nízkou hodnotu (0,9), aby se snímek rychle aktualizoval. Naopak malá změna signalizuje, že se objekt téměř nepohnul a proto je práh aktualizace vysoký (0,99), aby se téměř neaktualizovalo, aby nedošlo ke ztrátě ležící či nepohybující se osoby. Pro znázornění, jaký vliv má tato konstanta na tvar siluety, jsou na obrázku 5.7 znázorněny siluety pro různé hodnoty  $\alpha$  ve snímcích, kde není téměř žádný pohyb, když člověk leží nehybně po pádu (vlevo) a kde je velký pohyb, když člověk padá (vpravo).

Reálný  
snímek



$\alpha = 0,90$



$\alpha = 0,93$



$\alpha = 0,95$



$\alpha = 0,99$



*Obr. 5. 7 Ukázka siluet pro různé hodnoty  $\alpha$  pro malý pohyb před kamerou (vlevo) a pro velký pohyb před kamerou (vpravo)*

Přesné hodnoty konstanty  $\alpha$  použité v mém algoritmu jsou znázorněny v tabulce Tab. 2:

Změna $\alpha$	0-10	10-15	15-20	>20
	0,99	0,95	0,93	0,90

*Tab. 2 Konstanty  $\alpha$*

Protože aktualizují celý snímek, zůstává na pozadí i částečný otisk pohybujícího se objektu, který v dalších snímcích postupně mizí. Jeho barevná intenzita se příliš neliší od okolí, proto nebrání v dobré detekci pohybujícího se objektu v dalším snímku. Ukázku pozadí i s částečným pozůstatkem pohybující se osoby z obrázku Obr. 5.9, můžeme vidět na obrázku Obr. 5.8.



*Obr. 5. 8 Ukázka pozadí s částečným otiskem pohybujícího se objektu*



*Obr. 5. 9 Skutečný snímek, kde se nachází objekt, jehož „duch“ je zachycen na obrázku Obr. 8*

## **5.4 Vytvoření siluety**

Jako první je potřeba v algoritmu vytvořit siluetu odečtením aktuálního snímku od pozadí. Nejprve si pozadí i aktuální snímek vyfiltruji Gaussovým filtrem, abych odstranil šum, který by mohl dávat vysoké nebo naopak nízké rozdílové hodnoty, které by poté kazily celkovou informaci o snímku. Poté je od sebe odečtu ve všech třech jednotlivých barvách, protože se hodnoty pro jednotlivé barvy liší, vyberu tu barvu, kde je rozdíl nejvyšší, aby byla výsledná silueta co nejpřesnější. Výsledek odečtení snímků v ideálních podmínkách můžeme vidět na obrázku Obr. 5.10 a v reálných podmínkách na obrázku Obr. 11.





*Obr. 5. 10 Odečtení aktuálního snímku od pozadí v ideálních podmínkách*



*Obr. 5. 11 Odečtení aktuálního snímku od pozadí v reálných podmínkách*

Poté si tento rozdílový obrázek převedu na binární obrázek pomocí funkce *im2bw()*. Kde pouze nastavím práh  $p$  pro tento převod. Tato hodnota také, stejně jako konstanta  $\alpha$  u aktualizace pozadí, nemůže být konstantní, ale musí se měnit pro každý snímek. Určuji jí z průměrné intenzity pixelů v obraze. Opět pro znázornění, jaký vliv má tato konstanta na tvar siluety, jsou na obrázku 5.12 znázorněny siluety pro různé hodnoty  $p$  ve snímcích, kde není téměř žádný pohyb, když člověk leží nehybně po pádu (vlevo) a kde je velký pohyb, když člověk chodí (vpravo).

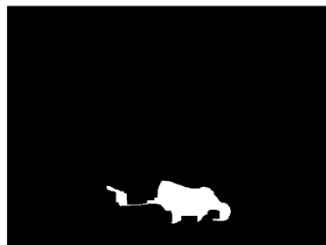
Reálný  
snímek



$p = 0,12$



$p = 0,15$



$p = 0,18$



$p = 0,22$



$p = 0,25$



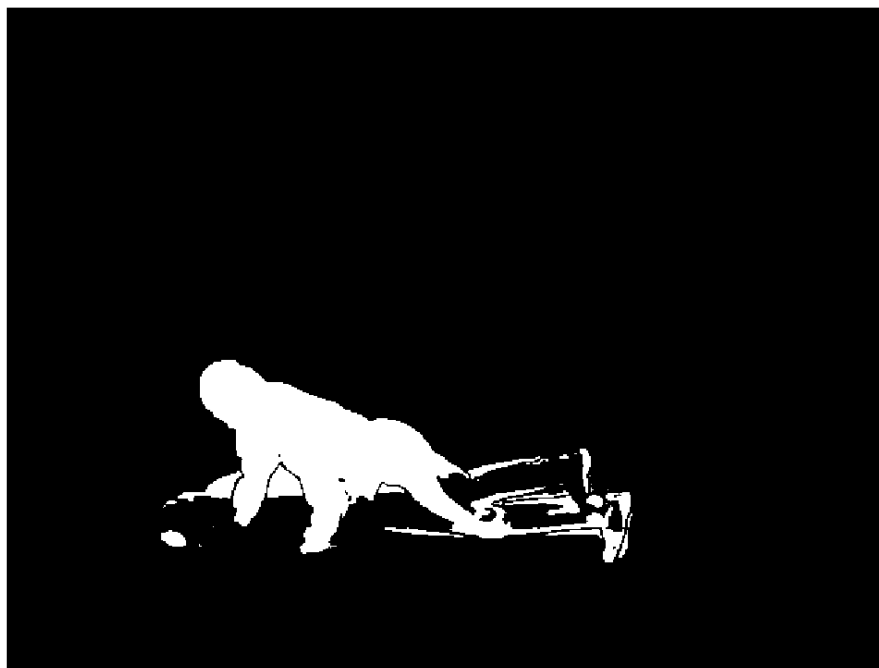
Obr. 5. 12 Ukázka siluet pro různé hodnoty  $p$  pro malý pohyb před kamerou (vlevo) a pro velký pohyb před kamerou (vpravo)

Přesné hodnoty tohoto prahu  $p$  použitého v mém algoritmu jsou znázorněny v tabulce Tab. 3:

Změna	0-5	5-10	10-15	15-20	>20
$p$	0,12	0,15	0,18	0,22	0,25

*Tab. 3 Konstanty  $p$*

Důvod je takový, že pokud je před kamerou hodně pohybu a aktualizace pozadí neaktualizuje dostatečně rychle, vznikají v obraze „duchové“ člověka, jako jsou vidět na obrázcích Obr. 5.10 a Obr. 5.11, kteří ale nejsou jasově tak výrazní a právě vysokým prahem u převodu na binární obrázek se dají z výsledné siluety vyfiltrovat. Dále pokud jsou v obraze stíny nebo odrazky lidí třeba ve skle, také nejsou jasově tolik výrazné a dají se takto opět vyfiltrovat. Binární obrázek siluety z obrázků Obr. 5.10 a Obr. 5.11 můžeme vidět na obrázcích Obr. 5.13 a Obr. 5.14.



*Obr. 5. 13 Binární obrázek siluety v ideálních podmínkách*



*Obr. 5. 14 Binární obrázek siluety v reálných podmínkách*

## **5.5 Úprava siluety**

Po převedení na binární obrázek nevznikne celistvá silueta člověka, jak je vidět na obrázcích Obr. 5.13 a Obr. 5.14, ale více menších objektů, ze kterých se musí člověk pospojovat a upravit okraje. Jak tento problém řeším, je znázorněno na diagramu (Diagram2) a podrobně popsáno níže. V následujících obrázcích se vždy nacházejí dva stejné snímky, aby bylo vidět, jak úprava siluety funguje a jaké jsou její výsledky.

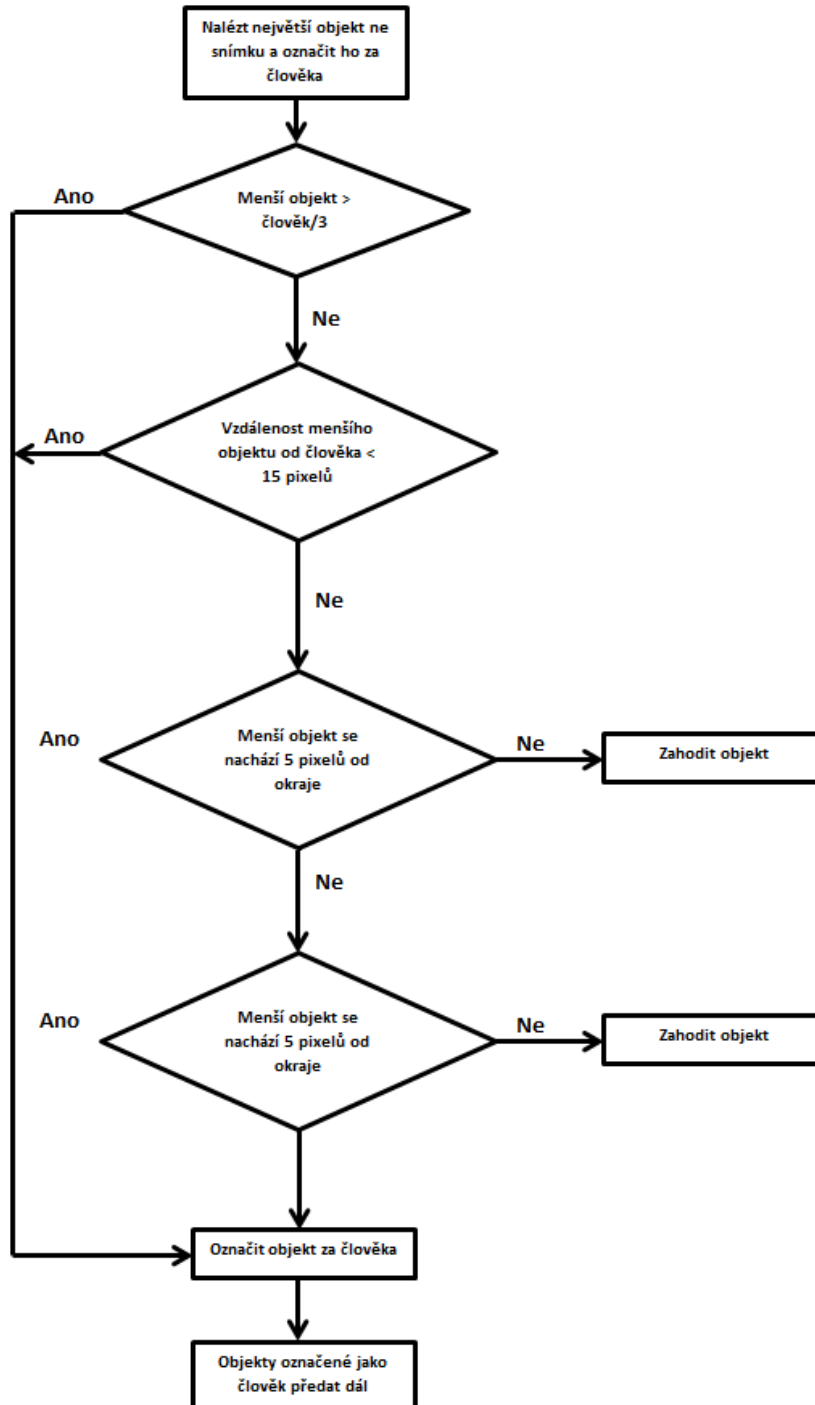
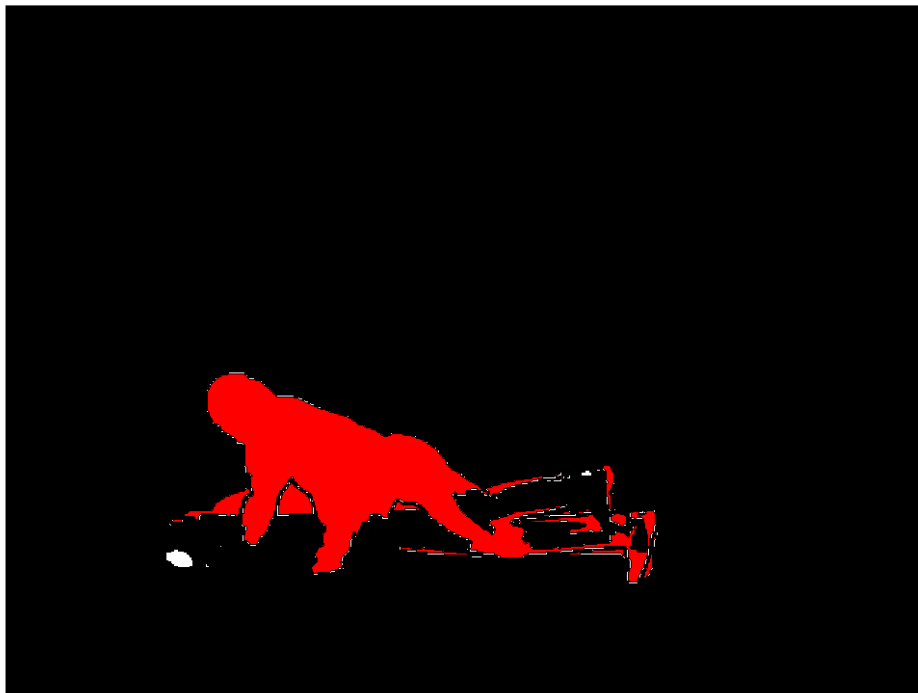


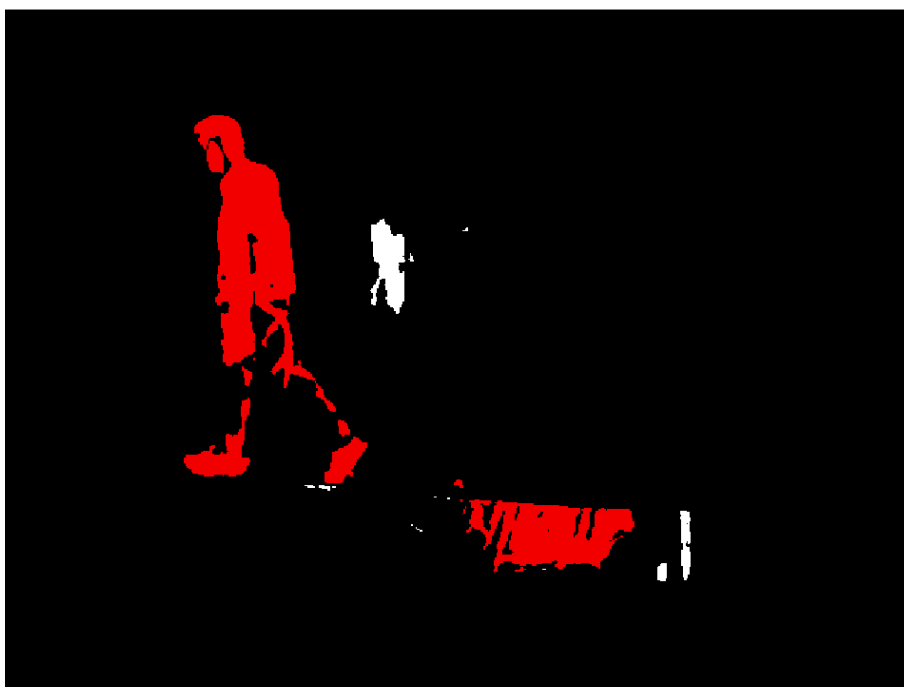
Diagram 2 Označení člověka

Můj algoritmus, počítá s tím, že největší objekt v obraze a objekty podobné velikosti (v mém případě stačí 30% velikosti největšího objektu) jsou základem člověka. Tyto velké objekty jsou základem člověka a nyní je potřeba k nim připojit menší objekty, které často představují části těla, které byly převodem na binární obrázek s příliš velkým prahem odděleny od zbytku člověka. V mém případě to jsou objekty vzdálené maximálně 15 pixelů od velkého objektu, toto číslo je zvolené takovéto, protože v dalším kroku provádím spojování těchto částí pomocí dilatace s velikostí jádra 15 a proto by se objekty

vzdálenější než 15 pixelu nepřipojily a proto nemá cenu je uvažovat. Tento výběr malých objektů provádíme dvakrát, aby se připojily všechny objekty. Jako poslední jsou za člověka považovány i malé objekty na kraji snímku. Zde se může jednat o člověka vstupujícího do snímku, který je zatím reprezentující malým objektem a v průběhu času poroste a je potřeba ho sledovat již od samého začátku výskytu ve snímku. Objekty označené za člověka z obrázků Obr. 5.13 a Obr. 5.14 jsou v obrázcích Obr. 5.15 a Obr. 5.16 znázorněny červeně.

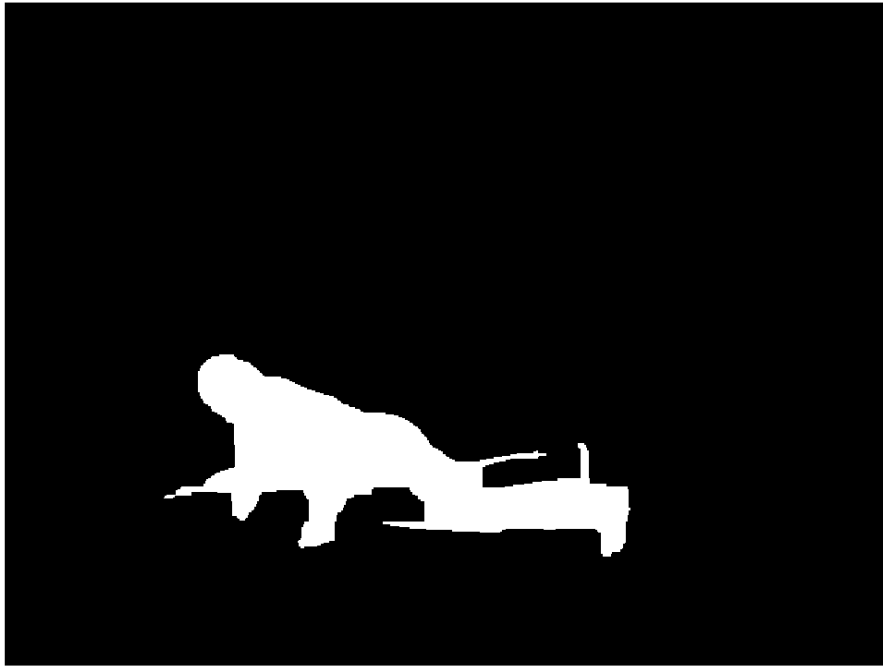


*Obr. 5. 15 Červeně označené části člověka v ideálních podmínkách*



*Obr. 5. 16 Červeně označené části člověka v reálných podmínkách*

Nyní mám vybrané objekty, které považuji za člověka a je zapotřebí z těchto objektů sestavit celistvou a uhlazenou siluetu pomocí morfologických operací, které jsme si popsali v kapitole 4.1.2. Nejprve provedu dilataci obrazu funkcí *imdilate()*, která nafoukne a spojí objekty podle velikosti jádra (v mém případě 15 pixelů). Poté provedu erozi obrazu funkcí *imerode()*, která mi objekty vrátí na původní velikost (v případě že použiji stejnou velikost jádra jako u dilatace), ovšem spojené objekty už nerozpojí a snaží se i zarovnat hrany, aby siluety nebyly příliš „okousané“. Poté už jen vyplním díry v siluetě, které vznikly při spojování pomocí dilatace a eroze, funkcí *imfill()*. Výsledný obrázek můžeme vidět na obrázcích Obr. 5.17 a Obr. 5.18.



*Obr. 5. 17 Celistvá silueta po aplikaci morfologických operací v ideálních podmínkách*



*Obr. 5. 18 Celistvá silueta po aplikaci morfologických operací v reálných podmínkách*



## 5.6 Párování objektů mezi snímky

Nyní je potřeba rozpoznávat stejné objekty mezi jednotlivými snímky. Abych věděl, jak se který objekt chová, zda se opravdu jedná o člověka a také z jeho vlastností později mohl detekovat pád, či jiné události. Algoritmus tohoto párování se znázorněn na diagramu (Diagram 3) a detailně popsán níže.

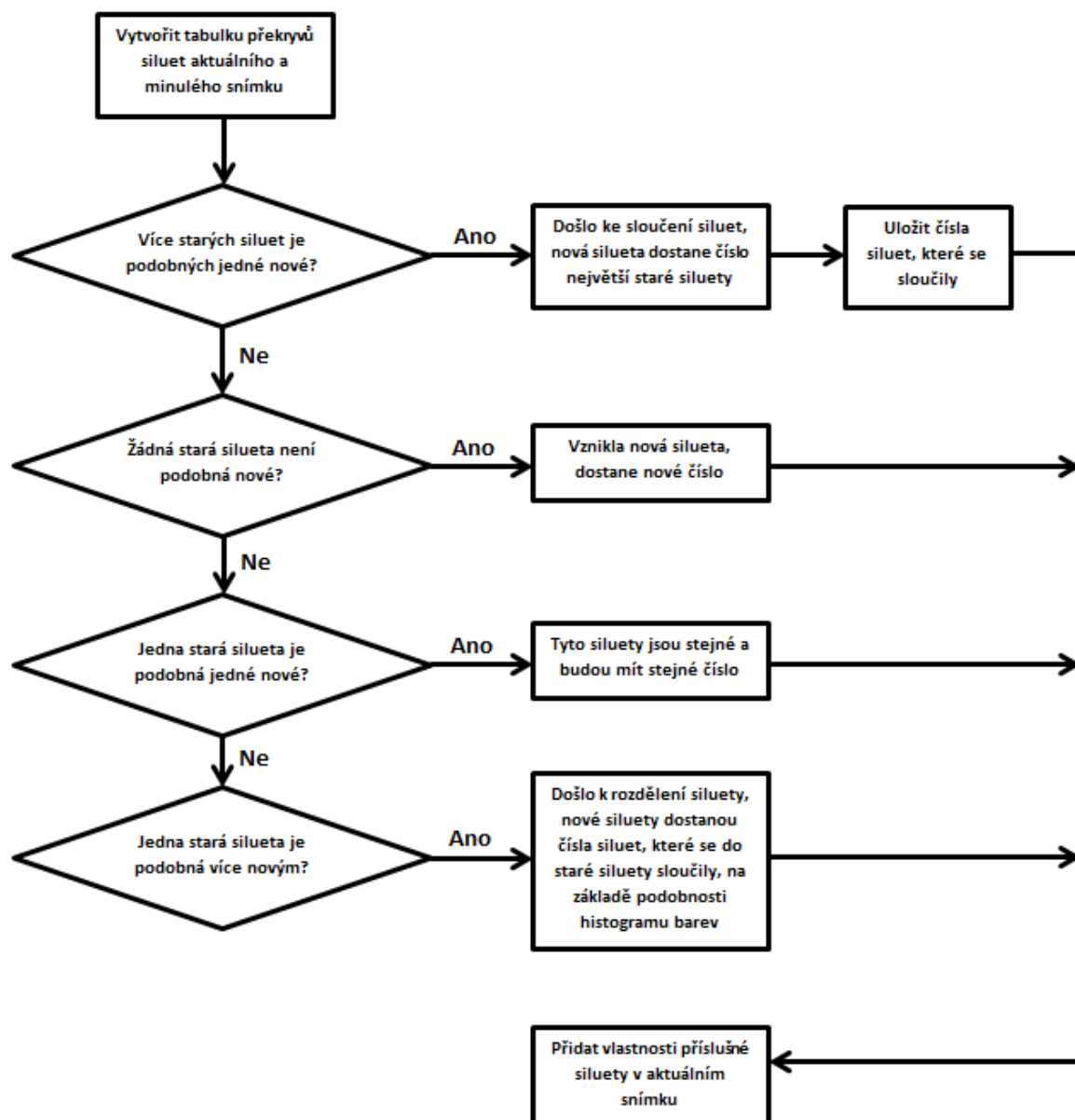


Diagram 3 Párování objektů mezi snímky

Na prvním snímku videa si všechny siluety označené za člověka uložím do tabulky, abych mohl později porovnávat jejich vlastnosti (velikost, orientaci, těžiště a barvu) a tím zjišťovat jejich podobnost nebo detekovat pád. A každé siluete přiřadím

jedno identifikační číslo, které bude mít po celou dobu výskytu ve snímcích. Vždy s příchodem nového snímku se vytvoří tabulka, ve které se porovnává nový snímek s tím předchozím. Zjišťuje se, které siluety jsou na obou snímcích stejné. Porovnávají se jednotlivé siluety z nového snímku s každou siluetou ze starého snímku. Zjišťuje se, které se vzájemně překrývají. Ty, které se překrývají z více jak 10%, jsou označeny za stejné. Tabulka je vytvořena tak, že na jednotlivých řádcích jsou siluety z nového snímku a ve sloupečcích siluety ze starého snímku. Pokud se silueta na novém snímku nepřekrývá s žádnou siluetou ze starého snímku, obsahuje její řádek samé nuly, jak je vidět v tabulce 5.3 na prvním řádku. Pokud jsou si ovšem silueta z nového a starého snímku překrývá, jsou na jejím řádku jedničky ve sloupečcích siluet, se kterými se překrývá, jak je vidět v tabulce 5.3 na druhém řádku.

		Staré siluety		
		1	2	3
Nové siluety	1	0	0	0
	2	0	1	0
	3	0	1	1

Tabulka 5.3

Pokud je celý řádek nulový, není s identifikací siluety problém. Silueta se ve snímcích objevila poprvé a dostane nové identifikační číslo a její vlastnosti se uloží do tabulky na příslušný řádek, podle identifikačního čísla a sloupec podle času kdy byl snímek pořízen. Pokud je na řádku pouze jedna jednička, znamená to, že silueta v novém snímku je ta samá, co byla v minulém snímku. Proto se nové siluety přiřadí stejné identifikační číslo, které měla silueta na minulém snímku a její vlastnosti se zapíší do tabulky na stejný řádek jako u siluety v minulém snímku ale o sloupeček dále. Pokud se na řádku nachází více jedniček, znamená to, že jedné nové siluety je podobno více starých siluet. K této situaci dochází, pokud se siluety spojily dohromady, například pokud se dvě jdoucí osoby proti sobě nacházejí v bodě, kdy jsou vedle sebe a vzájemně se překrývají. Tato situace je znázorněna na obrázcích Obr. 5.19a-d).



a) *Skutečný snímek dvou osob jdoucích proti sobě*



b) *Skutečný snímek dvou osob, které se překrývají*



c) *Binární obrázek dvou siluet jdoucích proti sobě*



d) *Binární obrázek dvou siluet, které se spojily do jedné*

*Obr. 5. 19 Skutečný a binární snímek sloučení dvou silueta*

V takové situaci, se nové siluety přiřadí identifikační číslo té největší ze starých jí podobných siluet, jak je vidět na obrázku Obr. 5.19c) silueta 125 je větší a proto spojená silueta na obrázku Obr. 5.19d) má také číslo 125. Opět se její vlastnosti uloží do tabulky. Dále se musí do zvláštní tabulky uložit všechny identifikační čísla starých siluet, které se spojily do nové siluety. Aby se vědělo, které siluety se sloučily do které, jsou vždy čísla všech sloučených siluet uloženy. Toto bude použito později, až dojde k opětovnému rozdělení takto sloučených siluet a bude potřeba vědět, které siluety se sloučily, aby se jim opět mohly přidělit jejich původní identifikační čísla, aby nedocházelo ke ztrátě identifikace siluety jen kvůli vzájemnému překryvu či sloučení. Problém s identifikací siluety nastává, pokud se v jednom sloupečku nachází více jak jedna jednička. Jak je to vidět v tabulce 5.3 ve sloupečku 2. To znamená, že dvě nové siluety jsou podobné jedné staré. Tedy že došlo k rozdělení jedné siluety na dvě či více. Například pokud se dvě osoby stojící blízko sebe najednou vzdálili. Tato situace je znázorněna na obrázcích Obr. 5.20a-d).



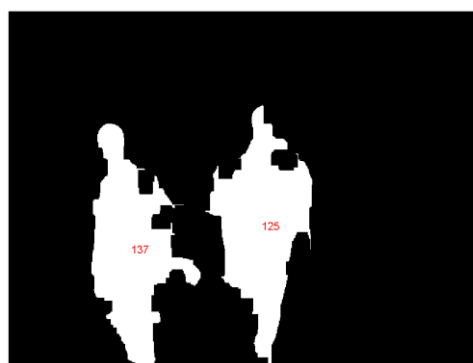
a) *Skutečný snímek dvou osob, které se spojily do jedné*



b) *Skutečný snímek dvou osob, které se opět rozdělily*



c) *Binární obrázek dvou siluet, které se spojily do jedné*



d) *Binární obrázek dvou siluet, které se opět rozdělily*

*Obr. 5. 20 Skutečný a binární snímek rozdělení dvou silueta*

Nyní je potřeba každé nové siluety vrátit původní identifikační číslo, které měly před spojením do jedné. Proto jsem si, jak bylo zmíněno výše, ukládal identifikační čísla všech siluet, které se spojily. Víím totiž, která silueta se nyní rozdělila a z tabulky, kam jsem si ukládal všechny siluety, které se spojily, víím, které siluety se do ní spojily. Proto už jen nyní musím, podle kritéria opět určit, která silueta je která po rozdělení. V mém algoritmu jako kritérium používám histogram jednotlivých barev pixelů v siluety. Vždy porovnám histogram barev nové siluety, se všemi histogramy barev sloučených siluet a vyberu tu nejvíce podobnou a takto to udělám u všech rozdělení vzniklých siluet a vrátím jim jejich původní identifikační čísla. Jak je vidět na obrázku Obr. 5.20d), kde siluety opět dostaly svoje čísla 137 a 125, jak tomu bylo před sloučením na obrázku Obr. 5.19c). Opět zapíši jejich vlastnosti do tabulky na příslušný řádek. Díky tomuto algoritmu nedochází ke ztrátě identifikace siluety při spojení a opětovném rozdělení.

## 5.7 Detekce pádu

Nyní už mám objekty spárované mezi snímky a můžu vyhodnocovat jejich vlastnosti a tím pádem i hlídat zda nedošlo k pádu nebo jiné neočekávané situaci. V mém algoritmu pád detekuji z pohybu těžiště objektu, vývojový diagram této detekce je znázorněn na diagramu (Diagram 4) a detailně popsán níže.

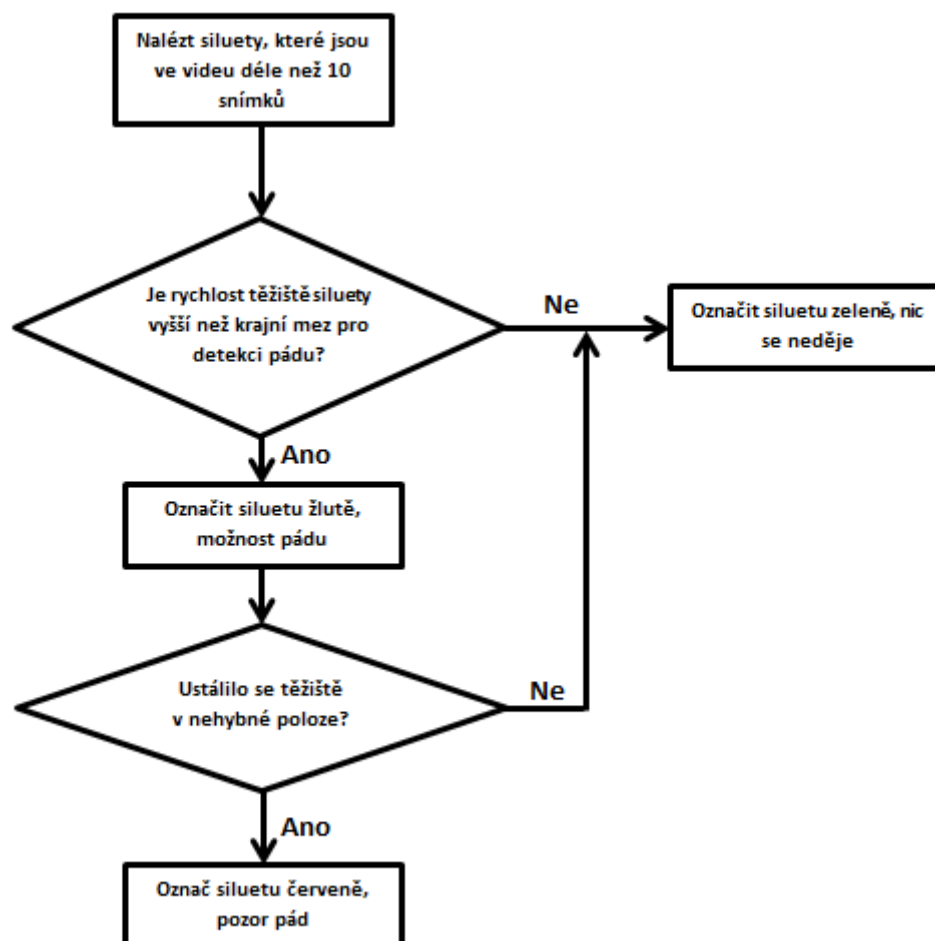
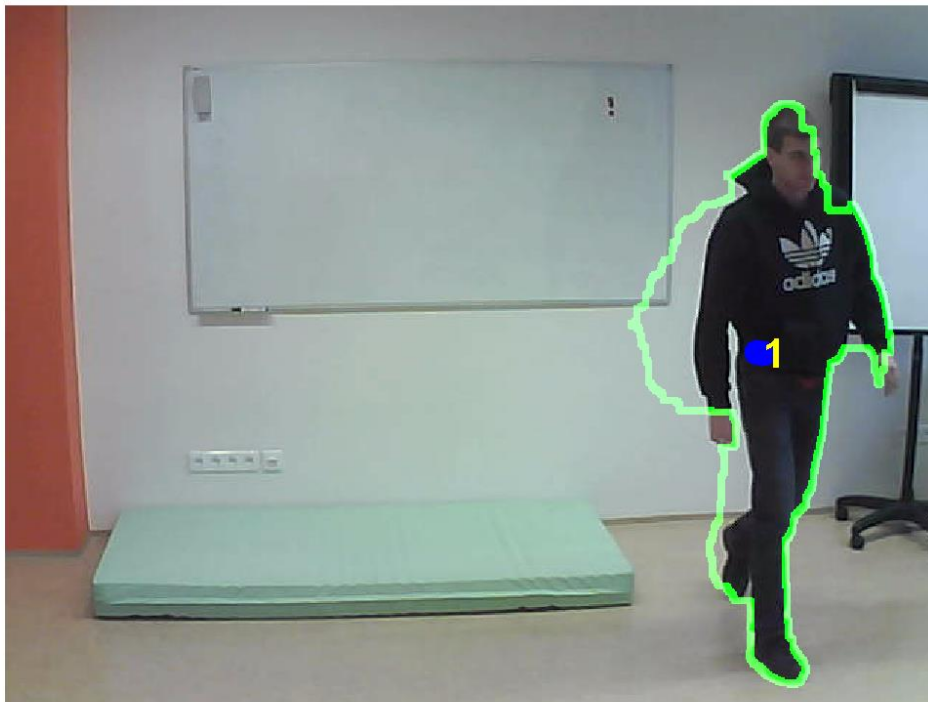


Diagram 4 Detekce pádu

Protože mám objekty spárované mezi snímky, můžu snadno zjistit, kde se nacházelo těžiště daného objektu v minulém snímku a tedy i o kolik se pohnulo. Bohužel, jak bylo uvedeno v dřívější kapitole, levné IP kamery nedokáží zajistit konstantní snímkovou frekvenci. Proto pro přesnější měření pohybu objektů nepoužívám vzdálenost těžišť, ale rychlost, tedy vzdálenost těžišť dělenou dobou, která uplynula mezi dvěma vzájemně se lišícími snímky. Pro rychlost už můžu určit mezní konstantu, která mi určí, zda se jedná o pád či nikoli i s proměnlivou snímkovou frekvencí. Ovšem nestačí jen samotná rychlost těžišť, důležitý je i směr pohybu. Pokud má být detekovaný pád správně, vždy se musí jednat o pohyb směrem dolů. Kdyby tam toto omezení nebylo, pád

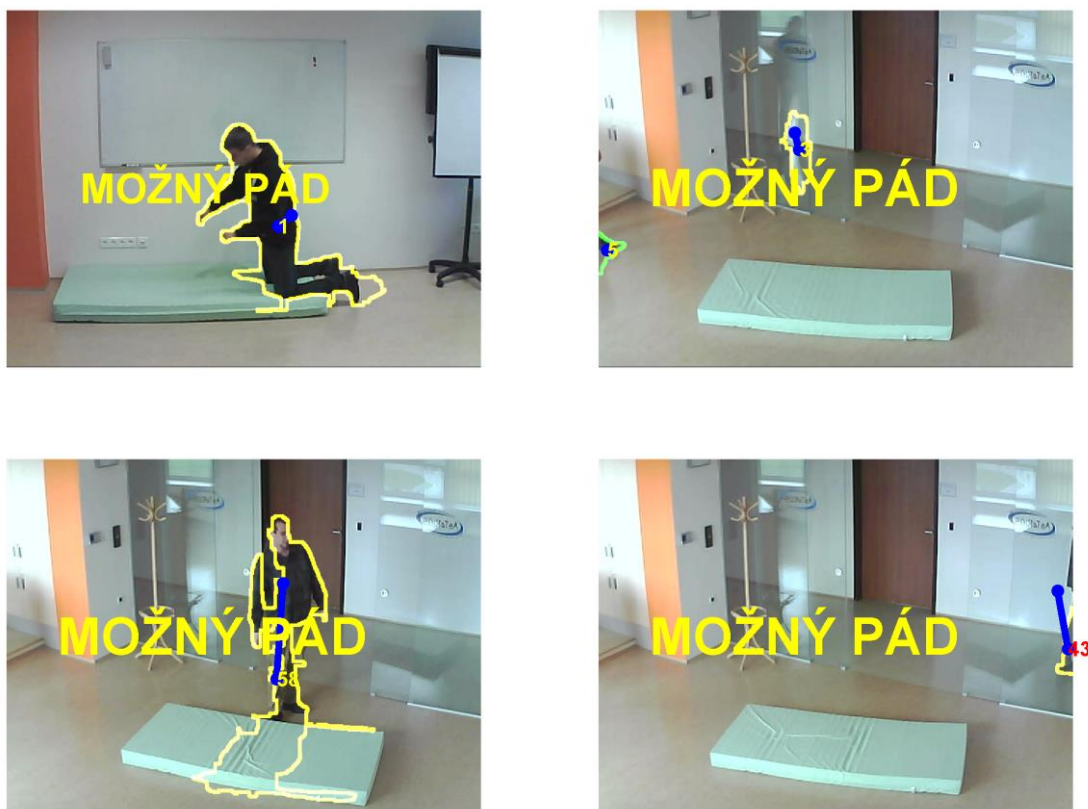
by vyvolal i rychlý vodorovný pohyb například u běhu nebo rychlý pohyb směrem nahoru například u výskoku.

Samotná detekce pádu je v mém algoritmu rozdělena na tři stupně. První stupeň, kdy je rychlost těžišť pod mezní konstantou nebo pohyb nesměruje dolů, znamená, že je vše v pořádku, nic podezřelého se neděje a obrys objektu je znázorněn zeleně. Tato situace je znázorněna na obrázku Obr. 5.21.



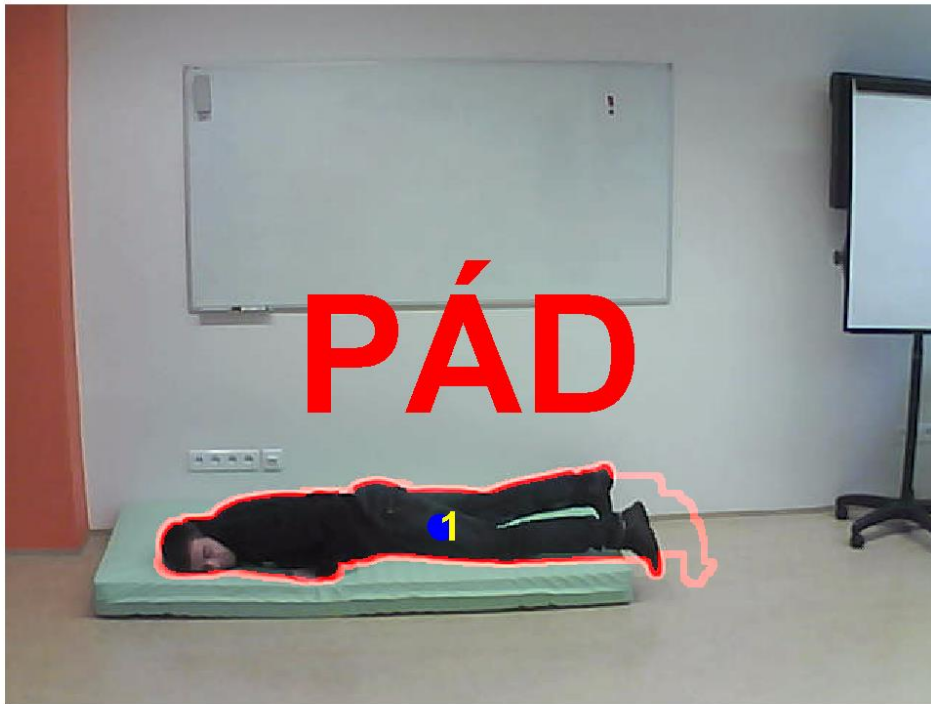
*Obr. 5. 21 Žádný pád*

Druhý stupeň, kdy je rychlost těžišť nad mezní konstantou a pohyb směřuje dolů, znamená, že by se mohlo jednat o pád. Proto je obrys objektu znázorněn žlutě a je spuštěn alarm upozorňující na možnost pádu. Tato situace je znázorněna na obrázku Obr. 5.22. Dále je na obrázku modrou úsečnou znázorněn pohyb těžiště mezi dvěma po sobě jdoucími snímky.



Obr. 5. 22 Znárodné příklady možného pádu i s pohybem těžiště znázorněného modrou úsečkou

Ovšem řada těchto alarmů může být falešných jak je vidět na obrázku Obr. 22, ať už pohybem na odrazivé ploše (vpravo nahoře), spojením dvou objektů a tím pádem velký posun těžiště (vlevo dole), většina objektu zmizela (vpravo dole), chybou v párování objektů nebo se člověk mohl jen pro něco shýbnout k zemi a další podobné případy. Ve výsledné aplikaci není žádoucí, aby se tento alarm vždy spouštěl a vyžadoval si pozornost operátora u monitoru. Protože se počítá, že bude připadat několik desítek detekčních systémů na jednoho operátora, který se jen v případě pádového alarmu podívá na obraz z kamery a vyhodnotí situaci. Proto i kdyby se tento falešný alarm objevil jednou za hodinu, znamenalo by to, že operátor by musel zhruba každou minutu kontrolovat nějaký obraz z kamery a to by bylo velice nepraktické. Proto zde mám třetí stupeň, který už je kritický a znamená, že opravdu došlo k pádu a hrozí nebezpečí. A je potřebná pozornost operátora u monitoru, aby vyhodnotil situaci. Tato situace je znázorněna na obrázku Obr. 5.23.



*Obr. 5. 23 Pád*

V tomto případě je obrys objektu znázorněn červeně a je spuštěn pádový alarm, který upozorní operátora u monitoru, že je potřeba se podívat na obraz z kamery a vyhodnotit situaci. K tomuto alarmu dochází pouze v případě, že byl vyvolán alarm možného pádu a poté se těžiště daného objektu v krátké době ustálilo na jednom místě a zůstalo nehybné. To v praxi znamená, že člověk upadl, zůstává nehybně ležet a hrozí nebezpečí. Operátor u monitoru je upozorněn a záleží už na něm, jak situaci vyhodnotí.

## **5.7 Přehledové video**

Aby se výsledky algoritmu dali dobře vyhodnocovat, vytvářím ze všech zpracovávaných snímků výsledné video. V tomto videu jsou pozorovaní lidé znázorněni obrysem siluety, který má barvu podle toho, zda došlo k pádu či nikoliv, v již barevném snímku. Dále má každá osoba svoje identifikační číslo, aby bylo dobře pozorovatelné, zda algoritmus na identifikaci siluet mezi snímky funguje správně. A nakonec je u každé osoby znázorněno těžiště v současném a minulém snímku, aby se podle jeho pohybu dal detekovat pád. Protože pád je velice rychlý pohyb, těžiště budou mezi snímky daleko od sebe. Tyto přehledová videa můžete nalézt v elektronické příloze ve složce Videá.



## 6 Závěr

Nejprve jsem provedl studii stavů a událostí, které je možné detekovat ve vnitřním prostředí na základě záznamu z kamerového dohledu. Z materiálů, které jsem shromáždil, jsem zjistil, které události je možné detekovat a z nich byla pro mě nejdůležitější detekce pádu. Protože to je nyní velice žádoucí téma, hlavně pro dohled starších a nemocných lidí, kteří potřebují včas přivolat pomoc v případě pádu, aby nedošlo k vážnějším následkům.

Poté jsem se pustil do vývoje vlastního softwaru. Nejprve jsem otestoval funkce v Matlabu pro detekci osob a obličejů. Tyto funkce nedosahovaly příliš dobrých výsledků ať už tím, že nemám dostatečně kvalitní videa nebo že jejich klasifikátory nejsou naučeny na mých datech. Proto jsem se rozhodl tyto funkce použít jen pro zpřesnění mého vlastního algoritmu.

Algoritmus implementovaného softwaru pracuje na základě odečítání aktuálního snímku od pozadí, které je ovšem potřeba v průběhu aktualizovat, kvůli změnám v pozorované scéně. Tímto vzniká silueta, která se dále zpracovává, aby vznikla celistvá silueta člověka a z jejích vlastností se mohl detekovat pád. Samotný pád se detekuje z rychlosti pohybu těžiště siluety mezi jednotlivými snímky. Tato detekce probíhá ve dvou stupních. Nejprve pokud je na základě pohybu těžišť detekován pád, je označen pouze jako možný pád. Poté je spuštěn druhý stupeň detekce, který zkoumá další vlastnosti možného pádu, jestli upadlá osoba zůstává nehybně ležet a nezvedá se, je spuštěn pádový poplach nebo jestli se dále pohybuje a tedy není potřeba pomoci.

Nakonec jsem můj software otestoval na demonstračních videích, které jsem nahrál v Centru asistivních technologií. Nejprve jsem můj software testoval na videích v ideálních podmínkách, kde dosahoval téměř bezchybných výsledků v detekci osob i případných pádů. Já jsem ovšem chtěl software, který bude možné nasadit i v reálných podmínkách to vyžaduje vypořádat se s obtížnější detekcí osob a pádů, protože v reálném videu je velké množství rušivých elementů. Software na videích v reálných podmínkách dosahoval také velice dobrých výsledků, detekoval všechny nastalé pády, ale také občas detekoval falešné pády způsobené špatně označeným objektem. Za určitých situací ztratil software objekt úplně, což bylo způsobeno málo kontrastní barvou pohybujícího se objektu a pozadí.

Celkově, ale můj software dosáhl uspokojivých výsledků i na videích v reálných podmínkách, protože detekoval všechny pády.

# Literatura

- [1] P. Manoranjan, H. M. E. Shah, Ch. Subrata, **Human detection in surveillance videos and its applications**, *EURASIP Journal on Advances in Signal Processing*, 2013
- [2] C. Stauffer, W. Grimson, **Adaptive background mixture models for real-time tracking**, *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 1999) (IEEE, Piscataway, 1999)*, pp. 246–252
- [3] A Elgammal, D Harwood, L Davis, **Non-parametric model for background subtraction**, in *6th European Conference on Computer Vision - Part II (ECCV '00) (Springer, London, 2000)*, pp. 751–767
- [4] F-C Cheng, S-C Huang, S-J Ruan, **Scene analysis for object detection in advanced surveillance systems using Laplacian distribution model**. *Syst. Man Cybern. Part C Appl. Rev. IEEE Trans.* 41(5), 589–598 (2011)
- [5] T. Ko, S. Soatto, D. Estrin, **Warping background subtraction**, *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010) (IEEE, Piscataway, 2010)*, pp. 1331–1338
- [6] S. Chen, J. Zhang, Y. Li, J. Zhang, **A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction**. *IEEE Trans. Ind. Inform.* 8(1), 118–127 (2012)
- [7] J.L. Barren, D.J. Fleet, S.S. Beauchemin, T.A. Burkitt, **Performance of optical flow techniques**, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '92) (IEEE, Piscataway, 1992)*, pp. 236–242
- [8] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, **Behavior recognition via sparse spatio-temporal features**, *2nd IEEE Joint International Workshop Visual Surveillance and Performance Evaluation of Tracking Surveillance (IEEE, Piscataway, 2005)*, pp. 65–72
- [9] Wann-Yun Shieh, Ju-Chin Huang - **Falling-incident detection and throughput enhancement in a multi-camera video-surveillance system**, *Medical Engineering & Physics, Volume 34, Issue 7, September 2012, Pages 954-963*
- [10] Laila Alhimale, Hussein Zedan, Ali Al-Bayatti - **The implementation of an intelligent and video-based fall detection system using a neural network**, *Applied Soft Computing, Volume 18, May 2014, Pages 59-69*
- [11] Chien-Liang Liu, Chia-Hoang Lee, Ping-Min Lin - **A fall detection system using k-nearest neighbor classifier**, *Expert Systems with Applications, Volume 37, Issue 10, October 2010, Pages 7174-7181*

[12] Mirmahboub, B (Mirmahboub, Behzad) ; Samavi, S (Samavi, Shadrokh) ; Karimi, N (Karimi, Nader) ; Shirani, S (Shirani, Shahram) - **Automatic Monocular System for Human Fall Detection Based on Variations in Silhouette Area**, *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING* Volume: 60 Issue: 2  
Published: FEB 2013  
Pages: 427-436

[13] David Nicholas Olivieri, Iván Gómez Conde, Xosé Antón Vila Sobrino - **Eigenspace-based fall detection and activity recognition from motion templates and machine learning**, *Expert Systems with Applications*, Volume 39, Issue 5, April 2012, Pages 5935-5945

[14] Hlaváč V., Sedláček M. - **Zpracování signálů a obrazů**, skriptum, *Vydavatelství ČVUT, 2009*

[15] Ing. Karel Horák, Ph.D., **Počítačové vidění**, dostupné online:  
[http://midas.uamt.feec.vutbr.cz/ZVS/Exercise10/content\\_cz.php](http://midas.uamt.feec.vutbr.cz/ZVS/Exercise10/content_cz.php) [citováno 15.12.2014]

[16] Fisher, R. B., et al. - **Dictionary of Computer Vision and Image Processing**, *New York: Wiley, 2005*

[17] Šonka M., Hlaváč V., Boyle R. - **Image Processing, Analysis and Machine Vision**, *Thomson, 3rd edition, ISBN 978-0-495-08252, 2007*

[18] Hošek Lukáš, **Filtrace šumu z digitálních fotografií**, *Bakalářská práce, Praha, Univerzita Karlova: Matematicko-fyzikální fakulta, 2007*

[19] **NetRex** dostupné online: <http://www.netrex.cz/cz/podpora/inteligentni-IP-kamerovy-sytem/> [citováno 15.12.2014]

[20] Tao J., Turjo M., Wong M-F., Wang M., Tan Y-P. **Fall incidents detection for intelligent video surveillance**. *Proceedings of the Fifth International Conference on Information, Communications and Signal Processing. 2005. p. 1590–4*

## Obsah CD

./Diplomová práce

./Videa

./Software

Text diplomové práce ve formátu PDF

Přehledová videa pro ukázkou funkčnosti softwaru

Jednotlivé M-fily našeho softwaru