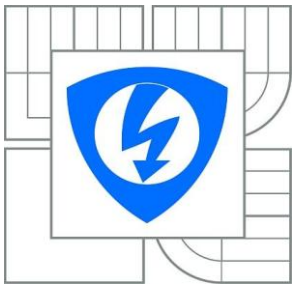
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SLEDOVÁNÍ OBJEKTU VE VIDEOSEKVENCÍCH

OBJECT TRACKING IN VIDEO SEQUENCES

DIPLOMOVÁ PRÁCE

MASTER´S THESIS

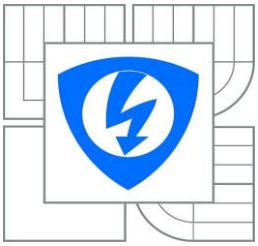
AUTOR PRÁCE — Bc. ZDENĚK LIBIŠ

AUTHOR

VEDOUCÍ PRÁCE — Ing. PETR ČÍKA, Ph.D.

SUPERVISOR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Zdeněk Libiš ***ID:*** 73012
Ročník: 2 ***Akademický rok:*** 2010/2011

NÁZEV TÉMATU:

Sledování objektu ve videosekvencích

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi rozpoznávání objektů ve video sekvenci. Vytvořte si video sekvenci, ve které si označte jeden objekt. Navrhněte algoritmus, který bude schopen automaticky sledovat pohyb vybraného objektu ve videu, určit směr jeho pohybu a jeho rychlost. Seznamte se s prostředím RapidMiner a vytvářením nových struktur. Implementujte váš návrh do tohoto prostředí a zhodnoťte jeho činnost.

DOPORUČENÁ LITERATURA:

[1] Bovik, Al (ed.). Handbook of Image and Video Processing. San Diego: Academic Press, 2000. ISBN 0121197905
[2] Report the Future [online]. 2010 [cit. 2010-10-11]. Rapid-i . Dostupné z WWW: <http://rapid-i.com/content/view/181/190/>.

Termín zadání: 7.2.2011 ***Termín odevzdání:*** 26.5.2011

Vedoucí práce: Ing. Petr Číka, Ph.D.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

Anotace

Práce se zabývá sledováním objektu ve video sekvenci. Zaměřuje se na studium pohybu jednoho objektu uvnitř jinak statické scény. Pohyb je určen směrem a rychlostí. Pro jejich určení jsou v rámci práce vytvořeny 3 operátory do prostředí RapidMiner. Operátor AccumulativeDifferenceImage zachycuje trajektorii pohybu scénou pomocí techniky kumulovaného rozdílového snímku. Operátor OpticalFlow je navržen pro popis typu pohybu a nalezení míry změny polohy. Pro určení rychlosti slouží operátor SpeedMeasuring, který na základě vstupních binárních masek vypočítá rychlost objektu v m/s. V teoretické části práce jsou popsány typy segmentačních technik, základní druhy vyhledávacích algoritmů, vlastnosti video sekvencí a problematika záznamu pohybu scény. V praktické části jsou vysvětleny implementace jednotlivých operátorů, popsány testovací video sekvence a zobrazeny výsledky jejich testů pro jednotlivé operátory.

Klíčová slova

Video sekvence, pohyb objektu, segmentace obrazu, měření rychlosti, kumulovaný rozdílový snímek.

Abstract

This thesis deals with object tracking in video sequence. It focuses on studying of one object´s motion in static background. Motion is defined by its direction and its speed. It was created 3 operators in RapidMiner to determine it. The operator called AccumulativeDifferenceImage searches a trajectory of motion by technique of accumulative difference image. Operator called OpticalFlow is created to describe type of motion and to find size of location´s transition. The operator called SpeedMeasuing is used for determining of speed, it calculates speed of object by using input´s binary masks in meters by second. In theoretical part of thesis are described the types of segmentation´s methods, basic types of block matching algorithms, attributes of video sequences and problem of recording of motions. In practical part are described implementations of every operator, the testing video sequences and showed results of tests for every operator.

Keywords

Video sequence, object´s motion, segmentation of picture, speed measuring, accumulative difference image.

Bibliografická citace mé práce:

LIBIŠ, Z. *Sledování objektu ve video sekvencích.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 61 s. Vedoucí diplomové práce Ing. Petr Číka, Ph.D.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Sledování objektu ve video sekvenci jsem vypracoval samostatně pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení §11 a následujících autorského zákona č.121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení §152 trestního zákona č.140/1961 Sb.

V Brně dne ....

podpis autora

Děkuji vedoucímu diplomové práce Ing. Petru Číkovi, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne .....

.....

podpis autora

Obsah

Seznam obrázků

Úvod

Počítačové vidění je vědní disciplínou, zabývající se napodobením lidského zraku strojovými technikami. Ukazuje se, že samotné snímaní reálného světa na vysoké úrovni, ať už ve statické podobě (fotografie) nebo v dynamické (videosekvence) nečiní dnešní technice problém. Obtížným se pro stroje jeví druhá část lidského obrazového vnímaní, kterým je zpracování přijatého vjemu a jeho interpretace. Člověk se tuto dovednost naučí v několika prvních letech svého života a poté již na základě předchozích zkušeností analyzuje i nové obrazové informace. Implementovat toto chování do umělé podoby se prozatím jeví jako jeden z nejtěžších úkolů. Rozmanitost objektů reálného světa se ukazuje pro dnešní počítačovou techniku jako nezvládnutelná.

Tato práce je součástí projektu zabývajícího se zpracováním, segmentací, reprezentací a následnou interpretací obrazových dat. Cílem projektu je rozpoznávání objektů obrazu a jejich popis. Dokončený projekt by měl být systémem schopným vyhledat a popsat reálné objekty uvnitř cizího obrazu. Projekt je značně rozsáhlý, proto je rozdělen do několika dílčích kroků mezi větší počet lidí. Každá část řeší určitý úsek samostatně, nicméně ve vzájemné kooperaci s ostatními. V konečné fázi budou jednotlivé části propojeny a celý systém otestován.

Celý projekt je zpracováván ve vývojovém prostředí RapidMiner v jazyku JAVA. Přestože toto prostředí není původně určeno pro práci s obrazovými daty, poskytuje svou operátorovou strukturou a efektivním zpracováním velkoobjemových datových struktur dobré podmínky pro variabilní a rychlé zpracování obrazu.

Tato diplomová práce se zabývá analýzou pohybu objektů uvnitř statického prostředí. Cílem je na základě rozboru pořízeného videozáznamu scény pomocí některé segmentační techniky označit pohybující se objekt. Poté jeho pohyb zkoumat z hlediska směru a rychlosti. Pro určení směru je sledována trajektorie pohybu objektu scénou. Rychlost je popsána svou velikostí v horizontálním a vertikálním směrem. Teoretická část práce se věnuje především vlastnostem video sekvencí, segmentačním technikám používaných ve statických snímcích a sekvencích snímků, vyhledávacím algoritmům a problematice záznamu pohybu. V praktické části je vysvětlena implementace operátorů vytvořených pro určení směru a rychlosti pohybu objektu. Dále jsou představeny testovací video sekvence a zhodnoceny výsledky výstupů operátorů.

1 Video sekvence

1.1 Vlastnosti video sekvence

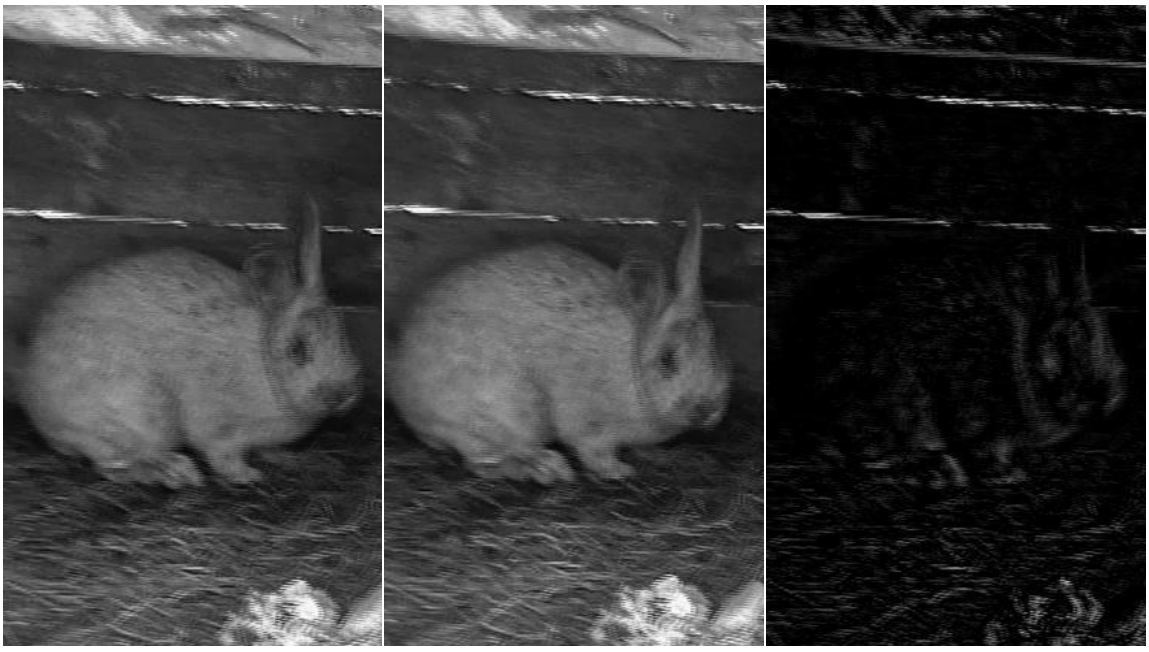
Záznam a následná reprodukce dynamických obrazových dat byly dlouhou dobu velkým problémem. Zatímco fotografie, jako statický záznam scény, se postupně vyvíjela a zachycovala události se stále větší mírou reálnosti, snímání pohybu scény po určitou dobu a následné přehrávání se jevilo jako nesplnitelné. Již od vzniku prvních přístrojů pro přehrávání „pohyblivých“ obrázků se využívalo nedokonalosti lidského zraku, tzv. setrvačnosti zrakového vjemu. Po průchodu světla sítnicí dochází k podráždění nervových zakončení. Látky, které způsobují toto podráždění, se určitou dobu obnovují (1/3 až 1/7 s). Při rychlých změnách jasu dopadajícího světla tedy nezaniká předchozí vjem a dochází ke sloučení s předchozím vjemem. Pokud je frekvence impulsů dopadající na oko nad určitou hranicí, lidský mozek vnímá střední hodnotu jasu. Tudíž pro vznik vjemu pohyblivé scény, stačí lidskému zraku vystavovat statické snímky zachycující scénu po určitou dobu v rychlém sledu [1]. Bylo zjištěno, že minimální hodnota snímkové frekvence pro vnímání plynulého pohybu je 25 – 30 snímků za sekundu [2]. Tato hodnota se dnes běžně využívá ve filmové technice. Na video sekvenci lze tedy pohlížet jako na rychlý sled statických snímků.

1.2 Vektory pohybu

Typickou vlastností dvou po sobě následujících snímků je jejich vzájemná podobnost, viz obr. 1-1. Tedy liší se jen v několika bodech snímku. V časové oblasti nastává velká redundance přenášených dat. Pro její kompresi se využívá predikce a kompenzace pohybu.

V datovém toku existují 3 základní typy snímků: I, P, B. Snímky I se kódují stejně jako statické obrazy podle standardu JPEG (viz [3] či norma ISO č. 10918) a nepodléhají časové kompresi. Pro snímky P, B jsou hledány vektory pohybu. V případě standardů MPEG 1(ISO 11172), MPEG 2(ISO 13818) spočívá hledání vektorů pohybu ve vyhledávání každého makrobloku snímku v předcházejícím snímku. Pro makroblok je tedy vždy uložena informace o poloze jako rozdíl polohy původního a nalezeného makrobloku. Makroblokem se označuje část jednoho snímku o velikosti 16x16 pixelů, viz [2], [3]. Pokud není makroblok nalezen s jistou dovolenou odchylkou v určité blízké poloze své původní polohy, je makroblok kódován jako typ I.

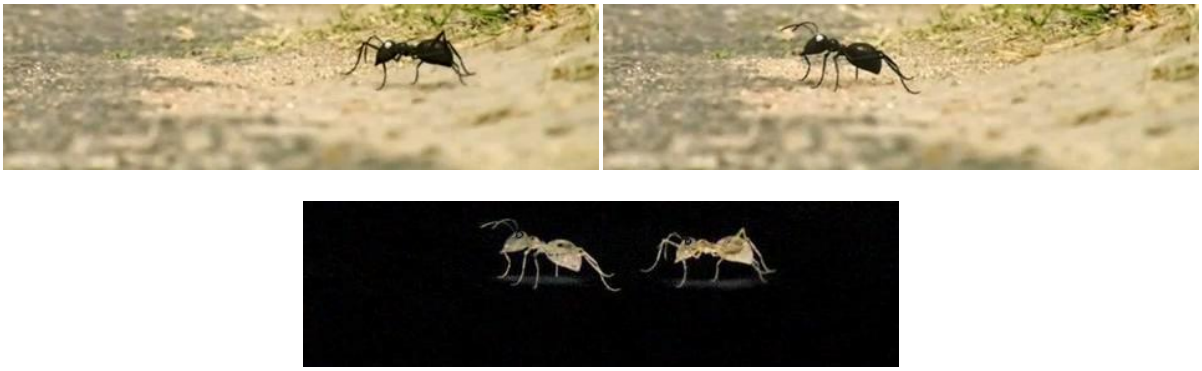
Jiná metoda byla zvolena ve standardu MPEG4Visual (ISO 14496 – Part 2), kde je obrazová scéna rozdělena na několik objektových rovin, přičemž každá tato rovina tvoří nějaký logický celek. Sekvence takové roviny tvoří objekt videa, který je určen několika parametry (textura, tvar, pohyb). S každou rovinou se při kompresi pracuje odděleně [2] [3]. Obou předchozích vlastností lze efektivně využít pouze v plynulé scéně, tedy bez střihů nebo změn polohy kamery. Pro naši další práci nás bude zajímat přístup použitý ve standardu MPEG4Visual, tedy považovat obraz za množinu objektů.



Obr. 1-1: Vlevo a uprostřed dva následující snímky video sekvence, vpravo jejich rozdílový snímek

1.3 Pohyb scény

Pohyb scény je způsoben dvěma základními faktory: pohybem objektu uvnitř scény při statickém umístění kamery (snímače) nebo pohybem kamery při statické scéně. Třetí možností je kombinace obou těchto způsobů snímání. Představme si případ jednoho pohybujícího se objektu uvnitř jinak statické scény, viz obr. 1-2. Horní obrázky jsou snímky video sekvence navzájem časově vzdáleny o hodnotu 1 s. Dolní snímek vznikl jako součet rozdílu prvého a druhého a rozdílu druhého a prvého. Z něj je patrný přesun objektu (mravence) na novou pozici a statika pozadí. Vektory pohybu budou nenulové pouze pro tento objekt. Zbytek obrazu tvoří pozadí, které se nepohybuje, vektory pohybu jsou tedy nulové.



Obr. 1-2: Snímky video sekvence, vlevo v čase *t*=0s, vpravo v čase *t*=1s, dole součet rozdílů obou snímků

V druhém případě - pohyb kamery vůči scéně - bude rozdílový snímek vypadat jinak. Všechny body dvou po sobě následujících snímků budou posunuty v opačném směru pohybu kamery. Vektory pohybu budou mít pro všechny body stejnou velikost a směr. Lze je tedy nahradit jediným vektorem pro celý snímek. Tento vektor bude mít opačný směr, než má vektor pohybu kamery.

1.4 Optický tok

Soubor vektorů pro všechny body obrazu lze označit jako optický tok. Pro zachycení změn v určité lokální oblasti obrazu je možné vytvořit parametrický model optického toku pro tuto oblast. Na základě parametrizovaných funkcí souřadnic dokáže model i s malým počtem parametrů přesně popsat změny uvnitř sledovaného regionu.

Pro malý region může postačovat jednoduchý parametrický model

$$u(x,y) = a_0 + a_1 x + a_2 y \tag{1.1}$$

$$v(x,y) = a_3 + a_4 x + a_5 y, \tag{1.2}$$

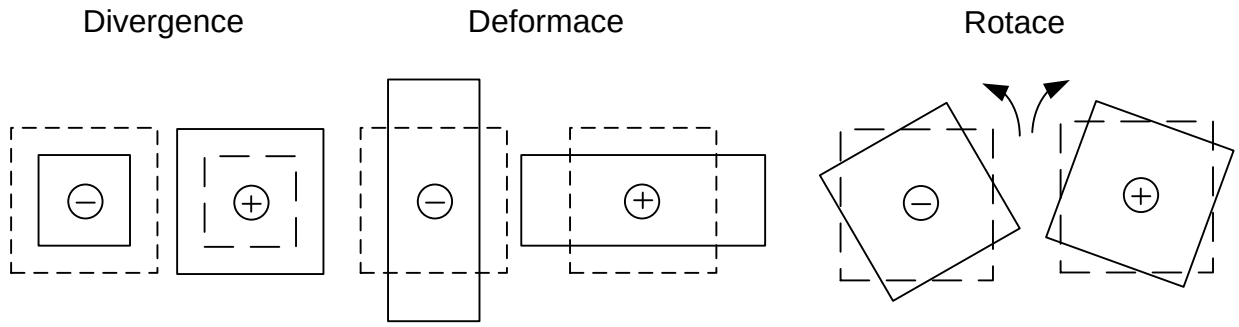
kde $a_0 - a_5$ jsou konstanty, $u(x,y)$a $v(x,y)$ jsou horizontální a vertikální složky optického toku v bodě o souřadnicích $[x,y]$. Parametry $a_0 - a_5$ charakterizují změnu regionu dle obr. 1-3. Pro translaci oblasti jsou důležité parametry a_0 (horizontální směr) a a_3 (vertikální směr). Další typy pohybu, jako rotaci, divergenci a deformaci je možné dodefinovat takto:

$$divergence = a_1 + a_5, \tag{1.3}$$

$$deformace = a_1 - a_5, \tag{1.4}$$

$$rotace = a_4 - a_2. \tag{1.5}$$

Tento model je použitelný pouze v případě, že se nemění rovina pohledu na vybraný objekt. V opačné situaci by bylo nutné použít planární model, který již počítá s možností perspektivní projekce [4].



Obr. 1-3: Interpretace znamének parametrů $a_0 - a_5$

2 Obraz jako soubor objektů

2.1 Vnímaní obrazu

Člověk na základě svých zkušeností je schopen velice rychle rozpoznat a popsat objekty (pozadí, předměty, osoby, písmo, atd.) v každém běžně pořízeném snímku. Bez uvažování abstraktních obrazů, např. děl kubistických mistrů. Pro počítač není však tato schopnost běžná. Počítač, v případě digitalizovaného snímku, považuje obraz za množinu pixelů. Pixely mají různou hodnotu jasu, barvonosných složek atd. Nicméně počítač není schopen určit, které pixely tvoří jeden objekt a které jiný objekt. Proces, který se snaží dát schopnost počítači rozdělovat obraz na objekty dle jejich podoby z reálného světa, se nazývá segmentace obrazu [5].

2.2 Segmentace obrazu

2.2.1 Předzpracování

Segmentaci obrazu musí nutně předcházet další procesy. Jsou jimi digitalizace obrazového vjemu a předzpracování obrazu. Digitalizace znamená převádění analogového signálu obrazu do diskrétního. Převod se může odehrávat přímo ve snímacím zařízení (digitální fotoaparát, digitální kamera) nebo až v počítači. Diskrétní hodnoty signálu jsou uloženy v počítači a tvoří matici pixelů určených svou pozicí, hodnotou jasu, či barvonosných složek. Digitalizace se skládá ze dvou základních postupů a to vzorkování a kvantování, viz [6]. Předzpracování obrazu je proces, při kterém dochází k určitým úpravám uloženého obrazu. Jedná se především o odstranění nežádoucích jevů vzniklých při pořízení snímku a jeho digitalizaci. Patří sem tedy potlačení šumu a zkreslení pomocí filtrace obrazu. Někdy se může předzpracování také zabývat zvýrazněním určitých rysů obrazu, jež budou použity dále při segmentaci (např. zvýraznění hran pro hranové detektory, viz část 2.2.3) [7].

2.2.2 Cíle a metody segmentace

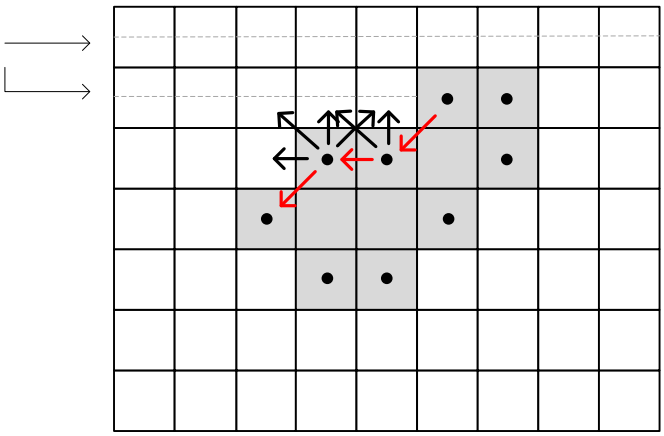
V segmentovaném obraze by měl být každý objekt oddělenou oblastí. Tato oblast je jednoznačně definována svou hranicí a oblastí, jež tato hranice obepíná. Z principu této definice vychází základní segmentační metody – metoda detekce hran a metoda narůstáním oblastí. Další metody kombinují tyto dva přístupy – hybridní metody. Některé další metody využívají globálních statistických informací o obraze – statistické metody. Další skupinu metod tvoří metody, které pracují na základě předdefinovaných předloh a jejich vyhledávání v obraze – znalostní metody [7].

2.2.3 Metody detekce hran

Za hranu se považují sousední body, jež mají významný rozdíl v hodnotě jasu. Spolehlivá a jednoznačná detekce hran v reálném obraze je velmi složitý a doposud nevyřešený problém. Velké obtíže činí hranovým detektorům kvantizační šum, rozostření obrazu atd. Hrany pak mohou být detekovány na pozicích, kde se nenachází nebo naopak skutečné hrany detekovány nejsou. Proto je nezbytné obraz před detekcí hran předpřipravit filtrací. Samotná detekce hran pro segmentaci obrazu nestačí. Je třeba následného propojení hranic do řetězců, které více odpovídají skutečnému průběhu. Na obr. 2-2 je zobrazen originální snímek a jeho podoba po detekci hran a následném pospojování hran. Z výsledku je patrné, že dolní část snímku (vodní hladina) se jeví detektoru hran jako šum.

Metoda sledování hranice

Pro binární obraz se jako základní metoda pro detekci hran může využít sledování hranice. Implementace spočívá v procházení obrazu pixel po pixelu a nalezení sousedních pixelů s odlišnou jasovou složkou. Jakmile je tato dvojice nalezena, upouští se od pixelového procházení celého zbylého obrazu. Pokračuje se určením vnější a vnitřní hranice, které spočívá na začátku v prohledání ve 4- nebo 8-okolí nalezené dvojice pixelů a poté postupně v okolí dalších nalezených bodů, viz obr. 2-1. Takto se postupně projde celá hranice objektu, vyhledávání končí při opětovném nalezení původních dvou bodů. Metoda v základní podobě bude dávat spolehlivé výsledky pouze v případech, kdy je celý objekt uvnitř obrazu a neobsahuje vnitřní duté oblasti.



Obr. 2-1: Postup při detekci hran sledováním hranice

Další metody

Základní metody detekce hran lze rozdělit na postupy využívající první derivaci (např. Cannyho operátor) a využívající druhou derivaci. Existují také hranové detektory, které se pokouší vytvořit parametrický model hrany z funkce obrazu. Tyto metody jsou sice výpočetně náročné, nicméně poskytují přesnější popis hran. Patří mezi ně Houghovy transformace, aktivní kontury, Level-sets a metoda pracující ve 3D Isosurfaces [7].

Obr. 2-2: Originál (vlevo) a obraz vytvořený detektorem hran (vpravo)

2.2.4 Metody narůstáním oblastí

Tyto metody se zaměřují na detekci oblastí namísto hranic mezi oblastmi (viz část 2.2.3). Jsou efektivnější pro zašumělý obraz, pro který je detekování hran velice obtížné. Stěžejní kritérium

pro segmentaci oblastí je homogenita jednotlivých ploch. Tímto kritériem nemusí být jen jas či barva, ale také textura, tvar nebo model. Existují dva základní postupy – metoda spojování oblastí a metoda štěpení oblastí. Při použití metody spojování oblastí je obraz rozdělen na menší části (možno až na úrovně pixelů). Jednotlivé sousední části jsou poté testovány, zda splňují společné kritérium (např. úroveň jasu). Pokud ano, jsou spojeny do větší části atd. obr. 2-3 zobrazuje originál snímku a výsledek metody statistického spojování oblastí pro barevné snímky.

Metoda štěpení oblastí nabízí opačný přístup. Celý obraz je chápán jako jediná oblast. Poté je postupně dělen podle zvolených kritérií homogenity stejných jako při metodě spojování oblastí. Kombinace těchto dvou základních přístupů tvoří metodu split and merge (štěpení a spojování oblastí). Metoda využívá stromové reprezentace obrazu. Obraz je cyklicky dělen na 4 kvadranty, jestliže je v něm zachycena nehomogenita. V případě, že jsou sousední oblastí dle měřených kritérií homogenní, jsou spojeny v jednu oblast [7].

Obr. 2-3[1]: Originální snímek (vlevo) a výsledek metody spojování oblastí

2.2.5 Statistické metody

Tyto metody využívají pro segmentaci některé globální statistiky obrazu. Nejčastěji je to histogram některé vlastnosti. Postup, který využívá histogramu jasu, se nazývá prahování. To lze považovat za nejjednodušší segmentační metodu. Tato metoda se hodí pouze pro určitý typ obrazů. Využívá konstantní odrazivosti či pohltivosti povrchu objektů nebo oblastí. Často lze tedy rozlišit snadno objekt od pozadí. Typickými obrazy, jež lze touto metodou segmentovat, jsou binární, či textové obrazy. Při použití této metody je určena často dle histogramu hodnota prahu a jednotlivé pixely jsou porovnávány s touto hodnotou [5]. Podle vzájemného vztahu jsou děleny do oblastí. Klíčovou problematikou této metody je optimální volba prahu. Jen málokdy je možné prahovat celý obraz dle jediné hodnoty prahu. Obraz je tedy nutné rozdělit do oblastí a v každé z nich určit lokální práh (tzv. adaptivní prahování).

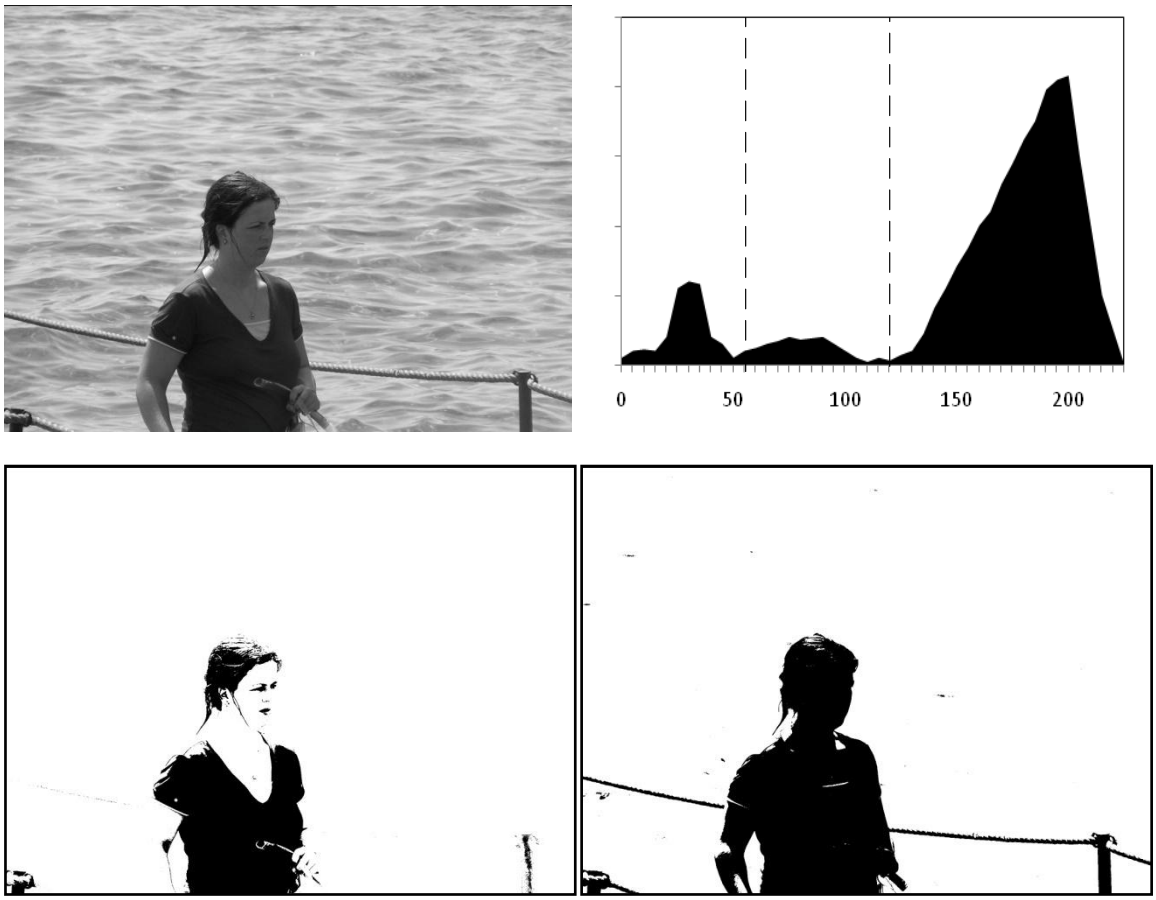
Pro hodnoty jasu pixelu $f(x,y)$ výsledného segmentovaného obrazu z obrazu původního $g(x,y)$ platí

$$f(x,y) = \begin{cases} 255 \text{ pro } g(x,y) \geq T \\ 0 \text{ pro } g(x,y) < T \end{cases}, \tag{2.1}$$

[1] Obrázek byl vytvořen pomocí java appletu na adrese http://www.sonycsl.co.jp/person/nielsen/SRM/

kde *T* je hodnota prahu.

Na obr. 2-4 je zobrazen histogram jasové složky původního obrazu s vyznačenými dvěma hodnotami prahu (55, 120), pro které byl obraz segmentován. Pod histogramem pak výsledné segmentované obrazy. Z porovnání výsledků je zřejmé, že pro různé hodnoty prahu dochází k segmentaci různých objektů. K dalším statistickým metodám patří shluková analýza (clustering), využití Kohonenových map nebo Markovských náhodných polí (angl. Markov Random Fields) [7].

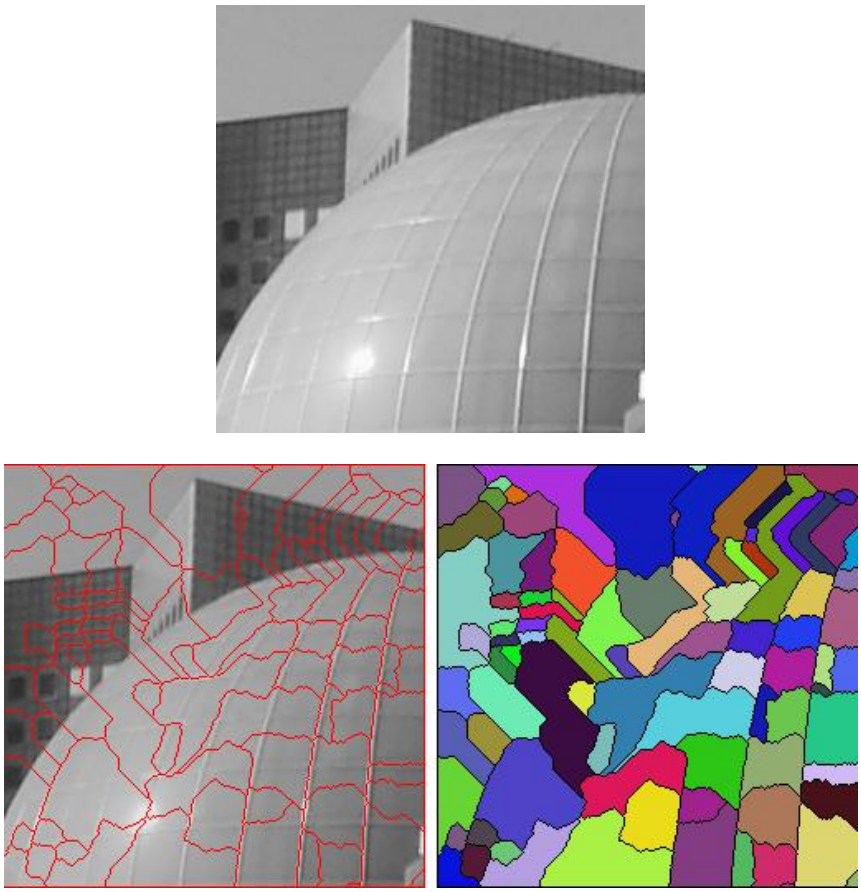


Obr. 2-4: Originální snímek (nahoře vlevo), jeho histogram (nahoře vpravo) a výsledné snímky metodou prahování pro práh o velikosti 55 (vlevo) a 120 (vpravo)

2.2.6 Hybridní metody

Do této kategorie lze zařadit metody, jež využívají některé postupy z předchozích skupin. Jednou z takových metod je watershed transform (viz obr. 2-5), která má nejblíže ke skupině metod narůstáním oblastí. Obraz je zde brán jako terén, jenž je postupně zaplavován z počátečních bodů (lokální minima obrazu), přičemž v místech, kde by se slévala voda ze dvou různých počátečních bodů, jsou vybudovány tzv. hráze. Proces zaplavování je zastaven při dosažení maxima obrazu.

Tato metoda je široce používána v lékařských aplikacích. V případě zašumělých obrázků produkuje velký počet regionů, které je třeba následně zpracovat, viz [8]. Odlišný přístup k segmentaci obrazu nabízí další hybridní metoda – použití neuronových sítí, viz [7], [9].



Obr. 2-5[2]: Originální snímek (nahoře) a segmentované snímky metodou watershed transform (vlevo) a v barevném rozlišení oblastí (vpravo)

2.2.7 Znalostní metody

Tato skupina zahrnuje postupy, které jsou založeny na předchozím vytvoření předloh nebo šablon obrazců, jejichž výskyt je v obraze očekáván. Na základě nalezené shody mezi šablonou a objektem v obrazu je poté segment lokalizován. Předpokládají tedy předchozí znalosti, apriorní informace. Tyto metody jsou velmi efektivní v obrazech, kde je struktura objektů velmi podobná. Naopak problémem je velká variabilita objektů (různá velikost, natočení, odstín či odchylka tvaru).

V lékařských aplikacích bylo v posledních letech dosaženo dobrých výsledků metodou Active Appearance Models. Tato metoda se také uplatňuje při detekci lidského obličeje a rozpoznání výrazu. Její hlavní nevýhodou je nutnost vytvoření rozsáhlé galerie trenovacích vzorů a jejich manuální anotace, což bývá časově náročné [7].

2.3 Segmentační metody v sekvenci snímků

Video sekvence přináší oproti statickému snímku třetí rozměr, kterým je časová informace. S měnícím se časem dochází ve snímcích ke změnám obrazové informace. Tato dynamika obrazových dat může být vyvolána různými aspekty, mezi které patří změna osvětlení, nežádoucí šum, ale především pohyb snímaných objektů scény (příp. pohyb kamery – relativnost pohybu). Většina aplikací analýzy video sekvence se zaobírá právě pohybem objektů.

[2] Obrázek byl vytvořen pomocí java appletu na adrese
http://bigwww.epfl.ch/demo/jwatershed/start.php

Její cíle lze rozdělit do 3 základních oblastí: detekce pohybu, lokalizace s následným popisem pohybujících se objektů a stanovení vlastností objektů [10].

Prvním krokem při analýze pohybu je jeho detekce. Pro ni lze využít rozdílových metod, které zaznamenají jakoukoliv změnu v porovnávaných snímcích, aniž by rozlišovaly, zda se jedná skutečně o pohyb těles, či jen šum, viz 2.3.1. Dalším stupněm je rozpoznávání objektů, hledání stejného objektu v následujících snímcích a následné určení směru pohybu s případnou predikcí trajektorie. Určení fyzikálních vlastností objektu se provádí za pomoci segmentačních technik používaných pro statické snímky, viz 2.2 [10].

2.3.1 Rozdílové metody

Tyto jednoduché metody detekce pohybu jsou založeny na výpočtu rozdílů obrazů v různých časových intervalech. Jestliže je rozdíl jasových hodnot na určité pozici větší než je velikost prahu ε, pak je na této pozici považován pohyb a je označena ve výsledném binárním obraze bílou barvou. Pro rozdílový snímek $d(x,y)$ obrazů $f_1(x,y)$ a $f_2(x,y)$ platí

$$d(x,y) = \begin{cases} 0 & |f_1(x,y) - f_2(x,y)| < \epsilon \\ 1 & |f_1(x,y) - f_2(x,y)| \geq \epsilon \end{cases}, \tag{2.2}$$

kde ε udává hodnotu prahu.

Na obr. 1-1 je ukázka rozdílového snímku dvou obrazů. Z něj jsou patrné výhody i nevýhody metody. Tedy rychlá detekce pohybujících se částí, nicméně žádná informace o směru pohybu nebo její velikosti.

Rozdílové snímky lze vytvářet ve dvou základních variantách:

- jednosměrný,
- obousměrný.

Jednosměrný rozdílový snímek

Jednosměrný snímek je tvořen jen kladnými hodnotami rozdílů mezi prvním a následujícím obrazem. Z definičního vztahu (2.2) vymizí absolutní hodnoty, pro snímky $f_1(x,y)$ a $f_2(x,y)$ a pak bude pro rozdílový snímek platit

$$d(x,y) = \begin{cases} 0 & f_1(x,y) - f_2(x,y) < \epsilon \\ 1 & f_1(x,y) - f_2(x,y) \geq \epsilon \end{cases}, \tag{2.3}$$

kde hodnota ε udává opět práh.

Nevýhodou této metody je, že část pohybujícího se objektu není v rozdílových snímcích zachycena, poněvadž si je dosti společná pro oba snímky a je odečtena [10].

Obousměrný rozdílový snímek

Obousměrný rozdílový snímek je počítán přímo dle vztahu (2.2). Obsahuje lepší informaci o podobě objektu oproti jednosměrnému.

Kumulovaný rozdílový snímek

Nevýhodou popsaných rozdílových snímků je, že neobsahují informaci o směru pohybu tělesa. Určení směru pohybu umožňuje modifikovaná metoda, kumulovaný rozdílový snímek. První ze snímků je označen jako referenční a výpočet rozdílového snímku probíhá pro každý obraz

sekvence vůči němu. Jednotlivé rozdíly jsou přičítány do výsledného kumulovaného snímku, viz také [11]. Příspěvek každého snímku může být také různé váhy w_i. Matematicky je pak kumulovaný rozdílový snímek d_{kum} pro sekvenci n snímků definován

$$d_{kum}(x,y) = \sum_{i=1}^{n} w_i \cdot |f_1(x,y) - f_2(x,y)| \text{ [10].} \tag{2.4}$$

Ukázky kumulovaných rozdílových snímků se nachází v kapitole 7.1.

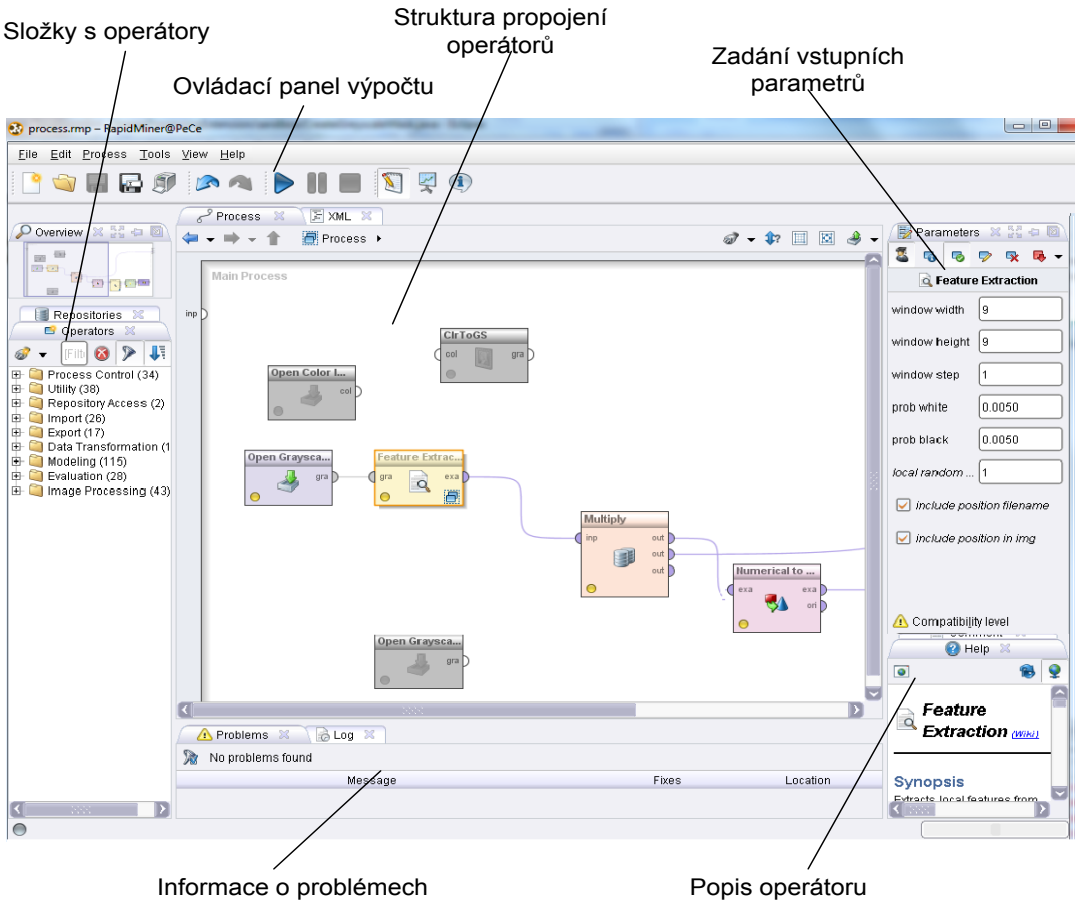
3 RapidMiner

RapidMiner je robustním nástrojem využívaným především pro data mining, prediktivní analýzu a business intelligence. RapidMiner je software nabízený jako opensource. RapidMiner existuje dle způsobu používání ve 3 verzích:

- Community Edition – bezplatná verze, určená pro vývojáře, kteří se chtějí podílet na vývoji otevřené distribuce Rapidmineru.
- Enterprise Edition – verze, která je pro profesionální použití. Kombinuje výhody Community Edition se specifickými požadavky konkrétního uživatele. Tato verze zahrnuje mj. automatický update, doplňkové operátory nebo systém podpory.
- Developer Edition – verze, která je určena pro vývojáře, kteří chtějí RapidMiner používat jako knihovnu ve svém komerčním software.

Celý proces výpočtu je rozdělen do struktury operátorů, kdy každý z nich zpracovává dílčí proces výpočtu. Operátory obsahují vstupní a výstupní porty, přes které putují data mezi nimi. Systém je tedy velmi pružný pro změny a úpravy při výpočetním procesu [12]. Popis prostředí RapidMiner je na obr. 3-1.

V naší práci budeme používat RapidMiner ve verzi 5.0. Pracujeme s knihovnou Image Processing Extension, která je rozšířením nástroje RapiMiner o podporu zpracování obrazů. Knihovna obsahuje pokročilé postupy v oblastech data miningu, zpracování obrazů, segmentace nebo sémantického porozumění obrazu.



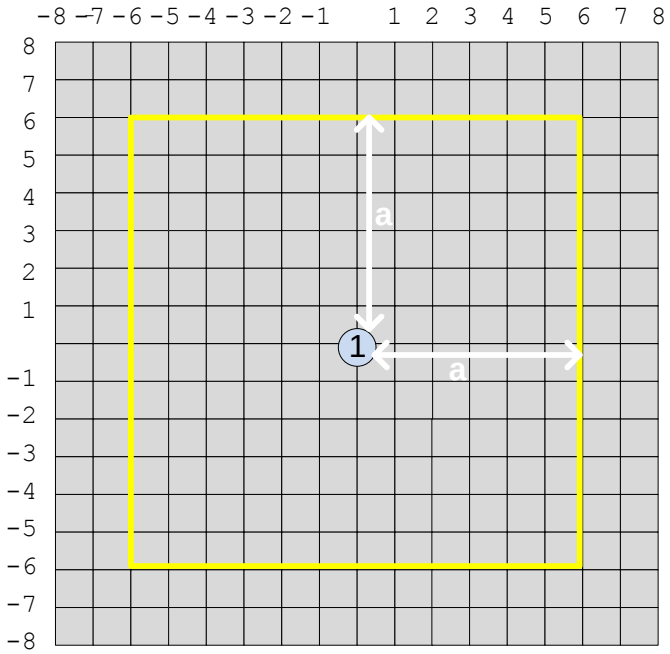
Obr. 3-1: Popis prostředí RapidMiner

4 Vyhledávací algoritmy

Z důvodu velké redundance obrazových dat v časové oblasti, tedy velké podobnosti po sobě jdoucích snímků (viz kapitola 1.1), byly vytvořeny komprimační techniky využívající této vlastnosti video sekvencí. V těchto postupech jsou přenášeny v originální podobě jednotlivé bloky či makrobloky (viz 1.2), nověji objekty videa pouze typu I. Pro ostatní typy (B, P) je přenášena pouze informace o změně polohy oproti stejnému bloku typu I. Základní problematikou je tedy nalezení odpovídajících si bloků, makrobloků, či objektů videa. Pro vyhledávání nejpodobnější bloků vzniklo několik základních algoritmů. Každá z nich vykazuje různou úspěšnost nalezení. Postupy zajišťující nejlepší výsledky většinou vykazují větší výpočetní náročnost. Pro porovnání podobnosti bloků používají vyhledávací algoritmy některé vyhledávací kritérium, viz kap. 5 [13].

4.1 Úplné vyhledávání - Full Search

Základní vyhledávací metodou je technika Full Search. Ta spočívá v porovnávání bloků v určitém vyhledávacím okně v okolí referenčního bloku. Volba velikosti vyhledávacího okna je klíčovou problematikou tohoto algoritmu. Volba menší velikosti zvyšuje riziko nenalezení shodného bloku pro rychle se pohybující blok. Naopak větší velikost zvyšuje značně výpočetní náročnost algoritmu. Pro vyhledávací okno o velikosti a bude existovat $(2 \cdot a + 1)^2$ prohledávaných bloků [13]. Na obr. 4-1 je vyznačeno vyhledávací okno o velikosti 5 pixelů.



Obr. 4-1: Vyhledávací okno v technice Full Search

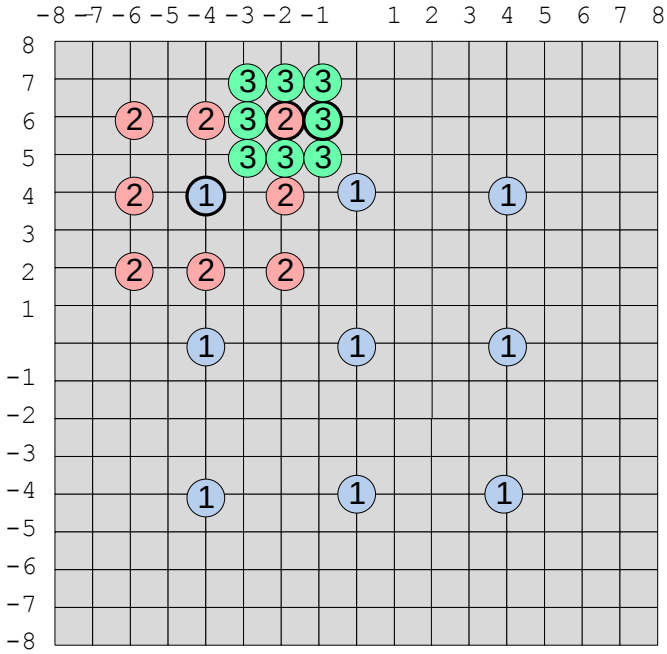
4.2 Suboptimální vyhledávaní

Jak již bylo řečeno, technika Full Search vykazuje velkou výpočetní náročnost, a proto byly vyvinuty postupy rychlejší a méně náročné. Tyto metody se zaměřují spíše na hledání lokálního minima rozdílu oproti globálnímu. To vede k menší míře přesnosti vyhledávání (menší míře shody výsledného bloku) než u techniky Full Search [14]. U každé metody je postup vysvětlen na schématu, na kterém jsou jednotlivé prohledávané pozice označeny číslem kroku a barevně pro každý krok odlišeny.

4.2.1 N-krokové vyhledávání (N-Step-Search)

Tento algoritmus vyhledává bloky v N krocích. Nejčastěji se používá ve verzi tříkrokové (označení Three-Step-Search). Velikost vyhledávacího okna je určena vztahem $\pm(2^N - 1)$ pixelů. Prohledávání probíhá jako porovnávání osmi pozic ve vzdálenosti $\pm 2^{N-1}$ od počáteční pozice. Pozice s nejvhodnější velikostí porovnávacího kritéria se stává počátkem pro další vyhledávací krok. V tom je snížena velikost vyhledávací oblasti na polovinu oproti předchozímu kroku. Následuje opět porovnávání osmi pozic a hledání té nejvhodnější. Takto se postupně snižuje vyhledávací oblast až do velikosti 1 pixelu. Pozice nalezená v tomto kroku se již bere jako výsledná [14].

Existuje více variant tohoto algoritmu, které se liší pouze způsobem zmenšování vyhledávací oblasti. Výše v textu i na obr. 4-2 je popsána varianta s půlením velikosti oblasti v každém kroku. V původní variantě autora tohoto algoritmu – Koga – se vyhledávací oblast v každém kroku snižovala o 1 [14].



Obr. 4-2: Postup vyhledávání metodou N-Step Search

4.2.2 Dvourozměrné logaritmické vyhledávání (Two Dimensioal Logaritmic Search, 2-D Logaritmic Search)

Tento algoritmus je také vícekrokový. Spočívá ve zmenšování vyhledávací oblasti v každém kroku, dokud není oblast triviálně malá. Lze jej rozdělit do 3 základních kroků. Jeho postup na obr. 4-3.

1. fáze – Na začátku je třeba určit maximální vzdálenost od původní polohy, do které bude algoritmus vyhledávat. Pro vzdálenost d bude velikost vyhledávacího kroku s rovna $s = 2^{(\log_2 d)-1}$. Původní pozice a dále pozice ve vzdálenosti $\pm s$ ve vertikální i horizontální ose vytváří 5 pozic, mezi kterými bude probíhat porovnávací proces. Pozice tvoří tvar kříže „+“.

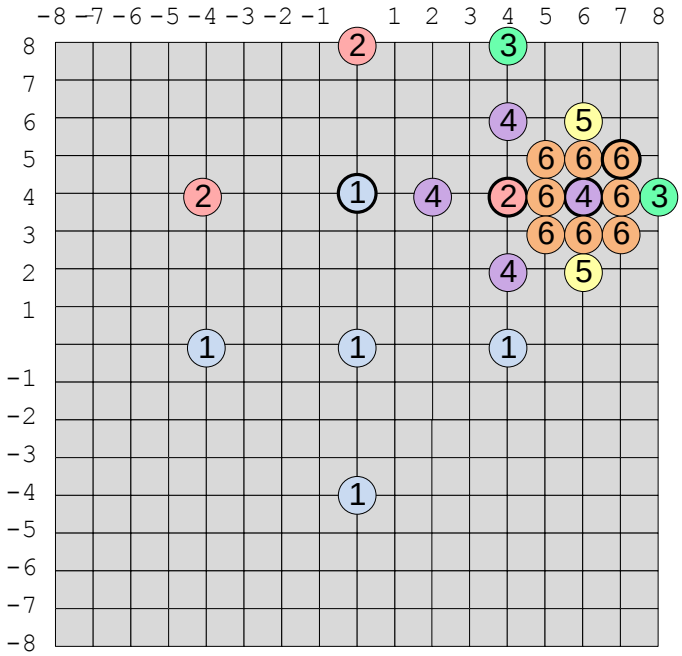
2. fáze – Pokud je nejlepší pozice (nejvhodnější dle vyhledávacího kritéria) nalezena v jiné poloze než té středové, stává se tato pozice novým středem a probíhá další kolo se stejnou

velikostí vyhledávacího kroku. Jestliže je nejvhodnější středová pozice, je snížen vyhledávací krok na polovinu a opět probíhá vyhledávací proces na 5 pozicích.

3. fáze – Jakmile je velikost vyhledávacího kroku *s* rovna 1, jsou prohledávány všechny okolní pozice kolem středové, tedy 9 pozic. Nejlepší z těchto pozic je výsledkem celého algoritmu.

Z popisu je jasné, že může nastat situace, kdy se vyhledávací pozice objeví mimo původně volenou maximální vzdálenost *d*. Aby algoritmus nepokračoval v hledání mimo vyhledávací oblast, je nežádoucím pozicím natavena maximálně nevýhodná pozice (+∞), tudíž nebudou nikdy algoritmem vybrány pro další krok.

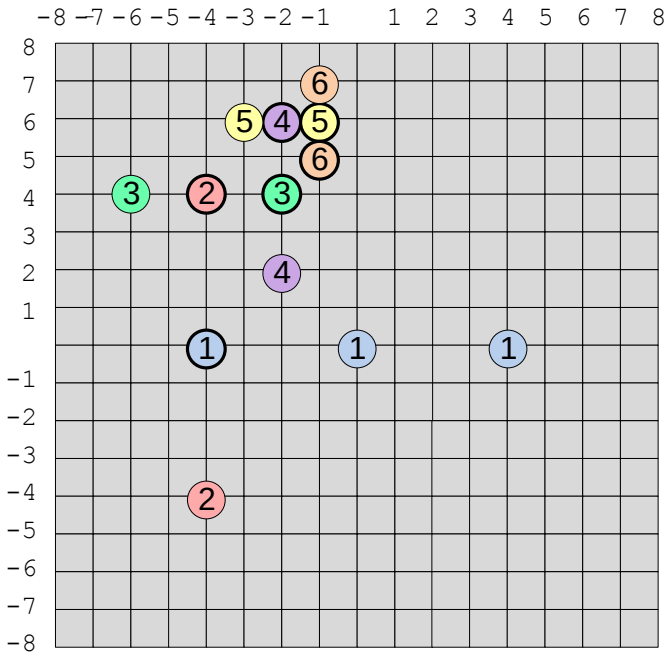
Tento algoritmus je možné používat ve více variantách lišících se zmenšováním vyhled. kroku. Původní varianta, výše v textu popisována, půlí vzdálenost pouze v případě, že je nalezena nejlepší podobnost ve středové pozici. Nadstavbou této varianty je půlení také v případě dosažení krajních hodnot vyhledávací oblasti. Existuje také možnost půlení kroku v každé fázi procesu. Obě tyto modifikace způsobují rychlejší konvergenci k výsledku. Nicméně varianta s půlením v případě krajních pozic vykazovala občasné selhání nalezení optimálního výsledku [14].



Obr. 4-3:Postup vyhledávání metodou 2-D Logaritmix Search

4.2.3 Ortogonální vyhledávání (Othogonal Search Algorithm)

Tento algoritmus lze označit za pokus o hybridní variantu předchozích dvou. Probíhá střídavě v horizontální a vertikální rovině. Velikost počátečního vyhledávacího kroku *s* je rovna $s = \frac{(d+1)}{2}$, kde *d* označuje maximální vzdálenost vyhledávací oblasti od počáteční pozice. V první fázi probíhá vyhledávání mezi počáteční pozicí a pozicemi ve vzdálenosti $\pm s$ v ose *x*. Pozice s nejlepší podobností se stává středem pro vertikální fázi vyhledávání mezi touto a dvěma pozicemi ve vzdálenosti $\pm s$ v ose *y*. Po každé dvojici fází, horizontální a vertikální, dochází ke snížení kroku *s* na polovinu. V poslední iteraci ($s = 1$) je vybrána nejvhodnější pozice ve vertikálním směru jako výsledná a algoritmus končí, [14] viz obr. 4-4.

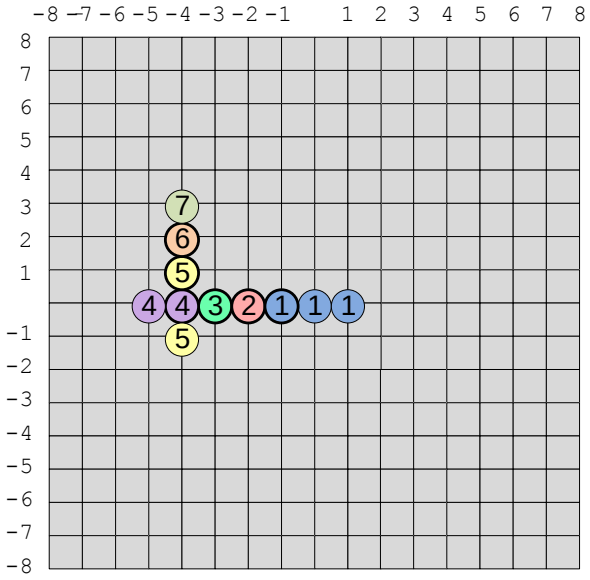


Obr. 4-4: Postup vyhledávání metodou Ortogonal Search

4.2.4 Once at a Time Search

Once at a Time Search je velmi jednoduchou, nicméně efektivní metodou. Podobně jako předchozí algoritmus vyhledává v horizontálním a vertikálním směru. Nejprve se postupuje v horizontálním směru, a to tak, že se testují body v bezprostředním okolí počátku na ose *x*. Nejvhodnější z nich je novým středem a opakuje se předchozí krok. Jakmile je středový bod nejvhodnějším, mění se směr vyhledávání na vertikální, a postupuje se obdobně jako v horizontálním směru. Nejvhodnější pozice z vertikálního směru je výsledkem algoritmu, viz obr. 4-5.

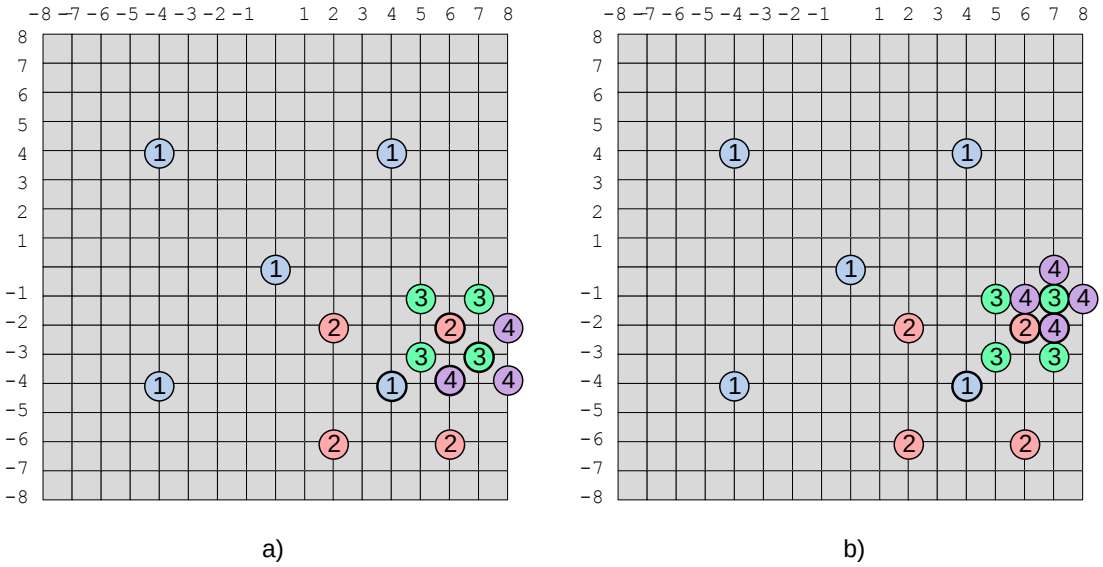
Existuje také komplexnější varianta tohoto algoritmu, která neobsahuje pouze jediný průchod ve vodorovném a svislém směru. Nýbrž pokračuje, dokud není nalezená pozice obklopena pouze horšími pozicemi [14].



Obr. 4-5: Postup vyhledávání metodou Once at a Time

4.2.5 Křížové vyhledávání (Cross Search)

Tato technika má mnoho společného s původní variantou 2-D logaritmického vyhledávání. Byla představena M. Ghanbarim v roce 1990, viz [15]. Prohledávání opět probíhá v jednotlivých stupních, přičemž v každém stupni se dělí velikost kroku *s* na polovinu. Od počáteční pozice [0,0] jsou nalezeny 4 pozice ve vzdálenosti *s*, přičemž na rozdíl od 2-D logaritmického vyhledávání tyto pozice netvoří kříž, nýbrž tvar písmena „X“. Pozice s nejvhodnější velikostí vyhledávacího kritéria se stává počáteční pozicí pro další stupeň. V posledním stupni – velikost $s = 1$ mohou nastat dvě varianty. Pokud je nejvýhodnější pozice vlevo nahoře nebo vpravo dole (oproti středu) jsou zkoumány 4 body ve vzdálenosti ±1 od této pozice ve tvaru „X“, viz Obr. 4-6. Pro druhé dva možné výsledky (vpravo nahoře, vlevo dole) jsou prohledávány 4 pozice ve vzdálenosti ±1 ve tvaru „+“, viz obr. 4-6.



Obr. 4-6:Postup vyhledávání metodou Cross Search pro a) variantu "X" b) variantu "+"

4.2.6 Greedy Block Matching Algorithms (hladový algoritmus)

Obecně Greedy algoritmy v každém svém kroku vybírají lokální minimum, přičemž je možností, že takto bude nalezeno globální minimum. Greedy algoritmus řeší např. problém obchodního cestujícího. V oblasti blokového vyhledávání ve snímcích se pokouší Greedy algoritmus redukovat počet procházených pozic tím, že se přesouvá (nastavuje novou počáteční pozici) ihned jakmile nalezne vhodnější pozici než je ta středová. Greedy algoritmus tedy nehledá v každém svém kroku nejvhodnější pozici, ale pouze vhodnější než je ta počáteční.

Základní varianta

Greedy algoritmy existují ve více variantách. V té základní začíná v nulové pozici s vyhledávacím krokem rovným polovině maximální velikosti posunutí (velikosti vyhledávacího okna) tedy $s = \frac{(d+1)}{2}$. Jako první je testována pozice ve vzdálenosti $+s$ v horizontálním směru od poč. pozice, tedy pozici [+s,0]. Pokud je vhodnější než počáteční, stává se novým počátkem. Pokud tomu tak není, je zkoumána pozice ve vzdálenosti $+s$ ve vertikálním směru od počáteční hodnoty. Algoritmus tedy prohledává pozice proti směru chodu hodinových ručiček. Pokud ani jedna z pozic není vhodnější než počáteční, je vyhledávací krok snížen na polovinu $s = \frac{(s+1)}{2}$ a prohledávání opět začíná vpravo od středu.

Další varianty

Další varianta algoritmu je obdobná, rozdíl spočívá v počáteční velikosti vyhledávacího kroku, která je čtvrtinová oproti velikosti vyhledávacího okna, tedy $s = \frac{(d+3)}{4}$. Další varianta nabízí modifikaci ve zmenšování vyhledávacího kroku. Krok není půlen, nýbrž čtvrcen, tedy velikost nového kroku v případě zjištění nejvhodnější pozice ve středu je $s = \frac{(s+3)}{4}$.

Ve všech předchozích variantách může nastat situace, kdy bude měněn směr vyhledávání v každém kroku. Jako perspektivnější se jeví vyhledávat vhodnější pozici ve stejném směru. Tuto problematiku řeší následující modifikace základního algoritmu. V první části probíhá vyhledávání stejně. Změna nastává v okamžiku nalezení vhodnější pozice a dalším postupu. Např. pokud bude nalezena vhodnější pozice (dle vyhledávacího kritéria) na pozici [0,+s], bude další vyhledávací pozice ve stejném směru, tedy [+2s,0]. Původní algoritmus by pokračoval vpravo od nové pozice tedy na pozici [+s,+s].

Zefektivnění algoritmu

Pořadí, ve kterém jsou pozice vyhledávány (v původní verzi v protisměru chodu hodinových ručiček), může výrazně ovlivnit nejen výsledek, ale i rychlost vyhledávání. Bylo zjištěno, že častěji nastává pohyb ve vertikálním směru oproti horizontálnímu, čehož využívá další z modifikací algoritmu. Nejprve jsou prohledávány pozice vpravo a vlevo od středu, poté až nahoře a dole. Robustnější modifikací s výrazným zefektivněním algoritmu je vytvoření adaptivního greedy algoritmu, který by na základě předchozích bloků předpokládal směr pohybu a tedy pozici, na které by zahájil vyhledávání [16].

5 Vyhledávací kritéria, kritéria shodnosti

Každý z uvedených vyhledávacích algoritmů v kap. 4 potřebuje pro nalezení odpovídajících bloků v obrazu původním a vyšetřovaném určité kritérium, dle kterého budou bloky porovnávány. Některá z následujících kritérií se též využívají pro objektivní posouzení kvality komprimovaného snímku. Kritéria se pak nepočítají pro jednotlivé bloky, nýbrž pro celý obraz. V definičních předpisech kritérií to pak znamená nahrazení parametrů udávajících velikost bloků velikostí (šířkou a výškou) celého snímku.

Nyní již k nejčastěji užívaným porovnávacím kritériím. Pro všechna z nich bude v jejich předpisech platit, že A, B označují dva bloky o rozměrech m a n. Označení $A[p, q]$ udává hodnotu pixelu nacházejícího se v p-té řadě a q-tém sloupci bloku A.

5.1 Střední absolutní rozdíl – Mean Absolute Difference (MAD)

MAD je nejčastěji používaným vyhledávacím kritériem. Je definováno jako rozdíl odpovídajících si pixelů v každém bloku, přičemž absolutní hodnota těchto rozdílů je sčítána. MAD je definován dle výše definovaných veličin touto rovnicí:

$$MAD = \frac{1}{m \cdot n} \sum_{p=1}^{m} \sum_{q=1}^{n} |A[p, q] - B[p, q]|. \tag{5.1}$$

Nižší hodnota MAD signalizuje lepší shodu porovnávaných bloků. Při vyhledávání bloků bude tedy preferován ten, jehož hodnota MAD bude nejnižší. Toto kritérium bývá také často označováno jako střední absolutní chyba – Mean Absolute Error (MAE) [13].

5.2 Střední kvadratická chyba – Mean Square Error (MSE)

Definiční rovnice MSE je podobná rovnici pro výpočet MAD. Rozdíl v hodnotě pixelů je však tentokrát umocněn:

$$MSE = \frac{1}{m \cdot n} \sum_{p=1}^{m} \sum_{q=1}^{n} (A[p, q] - B[p, q])^2. \tag{5.2}$$

Nižší hodnota MSE opět značí lepší shodu. MSE může být také označováno jako střední kvadratický rozdíl – Mean Square Difference (MSD) [13].

5.3 Klasifikace rozdílnosti pixelů – Pel Difference Classification (PDC)

Na rozdíl od předchozích dvou kritérií, pro hodnotu PDC již hraje roli poloha každého pixelu uvnitř bloku. Pro stejné hodnoty např. kritéria MAD je možné obdržet velmi odlišné hodnoty PDC. Výpočet hodnoty PDC spočívá v procházení bloků pixel po pixelu, rozdíl odpovídajících si pixelů (pixelů na stejných souřadnicích v obou blocích) je porovnáván s určitou hodnotou prahu. Pokud je rozdíl nižší než tento práh, je hodnota PDC inkrementována. Předpis pro výpočet PDC:

$$PDC = \sum_{p=1}^{m} \sum_{q=1}^{n} t(p, q), \tag{5.3}$$

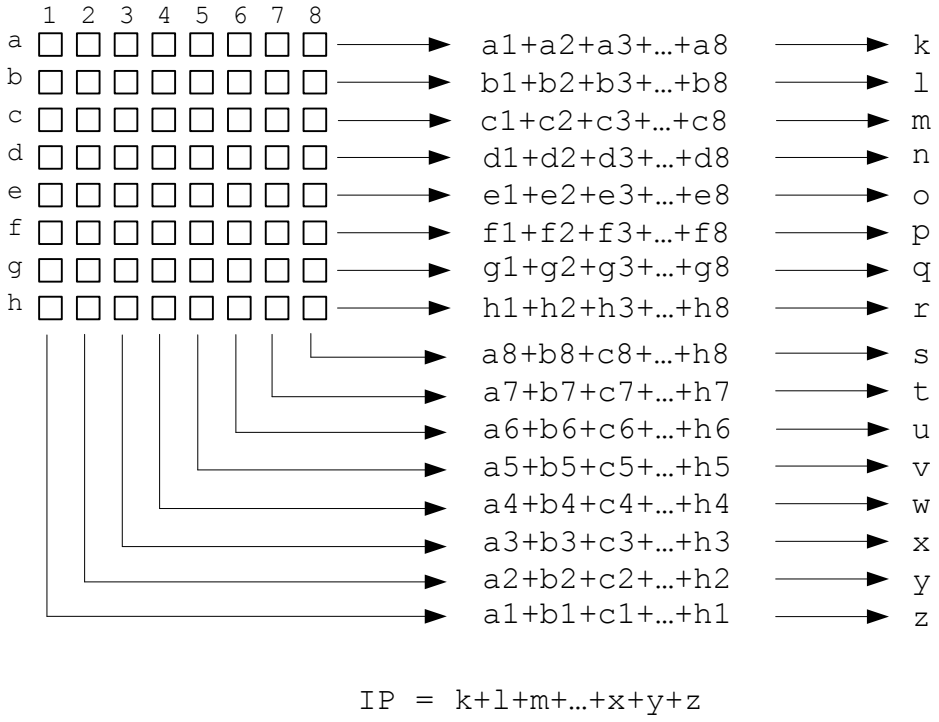
$$\text{kde } t(p, q) = \begin{cases} 1 & \text{pro } |A(p, q) - B(p, q)| \leq T \\ 0 & \text{jinak} \end{cases}, \tag{5.4}$$

kde T udává hodnotu prahu. Vyšší hodnota PDC udává vyšší počet pixelů s odpovídající si velikostí. Pro výsledek vyhledávání tedy bude preferován blok s nejvyšší hodnotou PDC [13].

5.4 Integrální projekce – Integral Projection (IP)

Hodnota integrální projekce je počítána jako suma hodnot rozdílů pixelů obou bloků v jednotlivých řádcích a sloupcích (viz obr. 5-1, kde je zobrazena ukázka výpočtu pro blok o velikosti 8x8 pixelů). Integrální projekci lze označit za filtraci dolní propustí, tudíž vykazuje menší citlivost na zašumělé snímky oproti ostatním vyhledávacím kritériím. Výhodou tohoto kritéria je možnost použití vypočtených hodnot v jednotlivých sloupcích nebo řádcích v jednom bloku pro další částečně se překrývající bloky. Tato výhoda se nejvíce projeví ve vyhledávacích algoritmech typu fullsearch, tedy technik používajících kompletní průchod vyhledávací oblastí. Předpis pro výpočet IP

$$IP = \sum_{p=1}^{m} \left| \sum_{q=1}^{n} \big(A(p,q) - B(p,q)\big) \right| + \sum_{q=1}^{n} \left| \sum_{p=1}^{m} \big(A(p,q) - B(p,q)\big) \right|. \tag{5.5}$$



Obr. 5-1: Výpočet hodnoty integrální projekce pro blok 8x8

Nižší hodnota IP značí vyšší podobnost porovnávaných bloků. Pro výsledek vyhledávání bude tedy jako nejvhodnější volen blok s nejnižší hodnotou IP [13].

6 Pohyb v obraze

Při zaznamenávání scény dochází nutně ke ztrátě jednoho rozměru reálné scény. Jestliže je pominuta dnes znovu objevená technologie 3D, jedná se vždy o záznam 3D scény na 2D plochu. Pohyb bodu ve scéně je reprezentován dvousložkovým vektorem rychlosti. Vektory rychlosti pro všechny body scény tvoří 2D obraz – rychlostní pole.

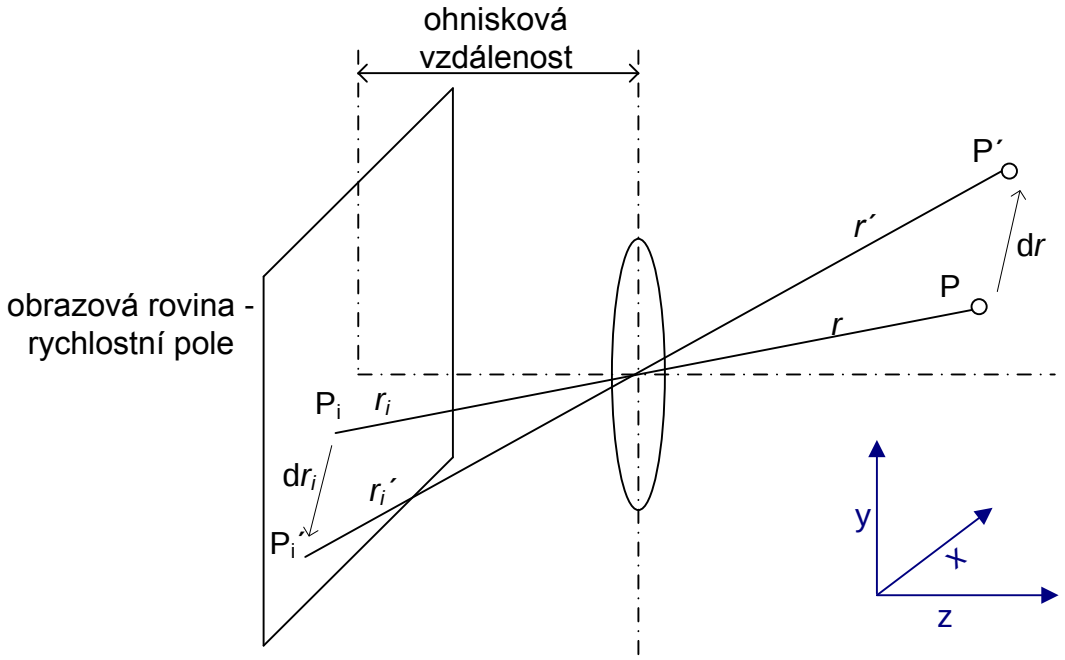
6.1 Zobrazení pohybu

Na obr. 6-1 je zobrazena situace změny polohy bodu ve scéně a její promítnutí přes zobrazovací soustavu do obrazové roviny. Bod A na pozici P, jenž je dána polohovým vektorem $\boldsymbol{r}$ $[x, y, z]$, je přemístěn do nové polohy $P´$ dané polohovým vektorem $\boldsymbol{r´}$ $[x´, y´, z´]$. V prostoru lze tuto změnu popsat vektorem pohybu $\mathrm{d}\mathbf{r} = \mathbf{r´}\text{-}\mathbf{r}$. Tento vektor lze vyjádřit jako pohyb rychlostí $|\mathbf{v}|$ ve směru vektoru $\mathbf{v}$ za čas $\mathrm{d}t$, tedy

$$\mathrm{d}r = v \cdot \mathrm{d}t. \tag{6.1}$$

Promítnutá změna polohy bodu A na obrazovou rovinu vymezí dva body P_i a $P_i´$, obdobně jako v prostoru scény, jsou definovány polohovými vektory $\boldsymbol{r_i}$ $[x_i, y_i]$a $\boldsymbol{r_i´}$ $[x_i´, y_i´]$. Tyto vektory jsou již jen dvousložkové, čímž dochází k výše diskutovanému zániku jednoho rozměru. Z podobnosti trojúhelníků lze odvodit vztah pro změnu vektoru rychlosti v rychlostním poli jako

$$\mathrm{d}r_i = r_i´ - r_i = v_i \cdot \mathrm{d}t \text{ [9]}. \tag{6.2}$$

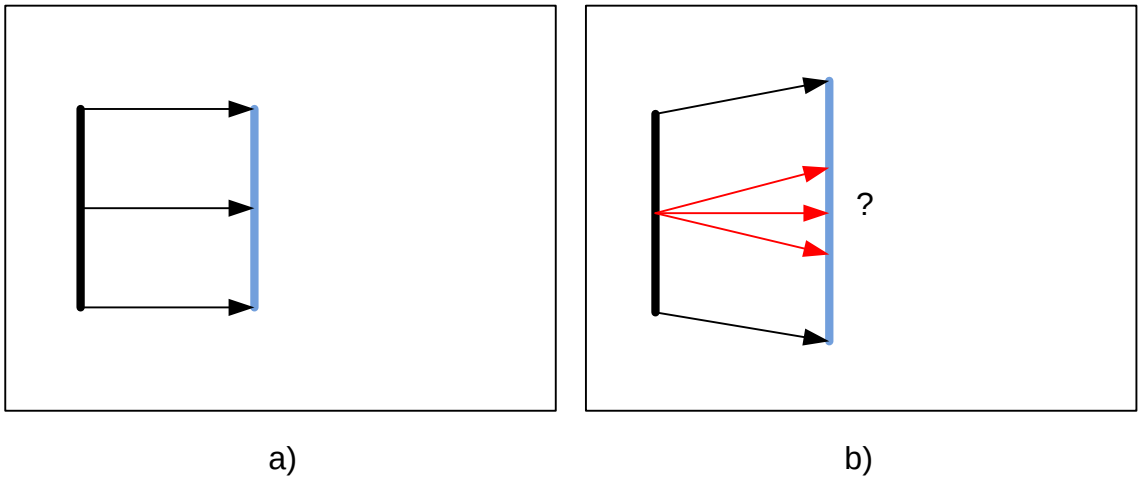


Obr. 6-1: Projekce pohybu z prostoru do obrazové roviny

6.2 Projekční problém

Tento způsob promítání pohybu a s ním spojená ztráta jednoho rozměru přináší problém nejednoznačnosti zobrazeného pohybu. Následující dva obrázky ukazují tento projekční problém. V případě obr. 6-2 a) se část pohybujícího objektu (zobrazena čárou) přemisťuje ve

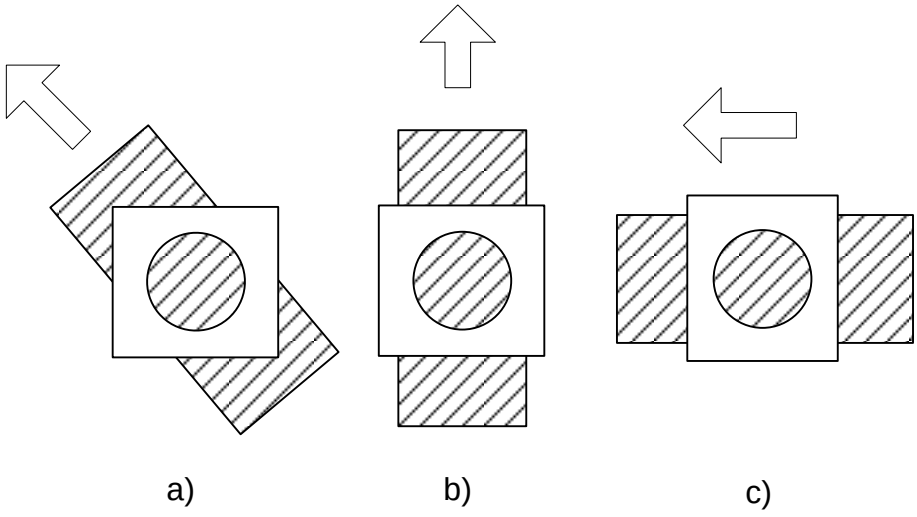
směru kolmém na normálu obrazové roviny. Segment si zachovává svoji vzdálenost od objektivu a projekční problém nenastává. Pohyb je jednoznačný. Na obr. 6-2 b) je již situace odlišná. Je patrné, že se změnila velikost segmentu – zvětšil se. Vzdálenost segmentu od objektivu se zmenšila a nyní nelze jednoznačně určit směr pohybu [10].



Obr. 6-2: a) Pohyb v rovině rovnoběžné s obrazovou rovinou - projekční problém nenastává, b) pohyb směrem k objektivu - projekční problém nastal

6.3 Aperturní problém

Další nejednoznačnost přináší aperturní problém. Ten je způsoben konečnou velikostí snímacího čipu a objektivu. Obr. 6-3 popisuje tento problém. Pokud se na texturu nahlíží průzorem menších rozměrů, jeví se pohyb vždy ve stejném směru, přestože se textura pohybuje různými směry, viz případy na obr. 6-3 [17].



Obr. 6-3: Aperturní problém: a) pohyb textury šikmo nahoru, b) svisle vzhůru, c) vodorovně vpravo

6.4 Velikost snímané scény

Rozměry scény, které je možné záznamovým zařízením zachytit určují především parametry objektivu. Objektiv je spojná optická soustava, kterou lze charakterizovat dvěma základními údaji

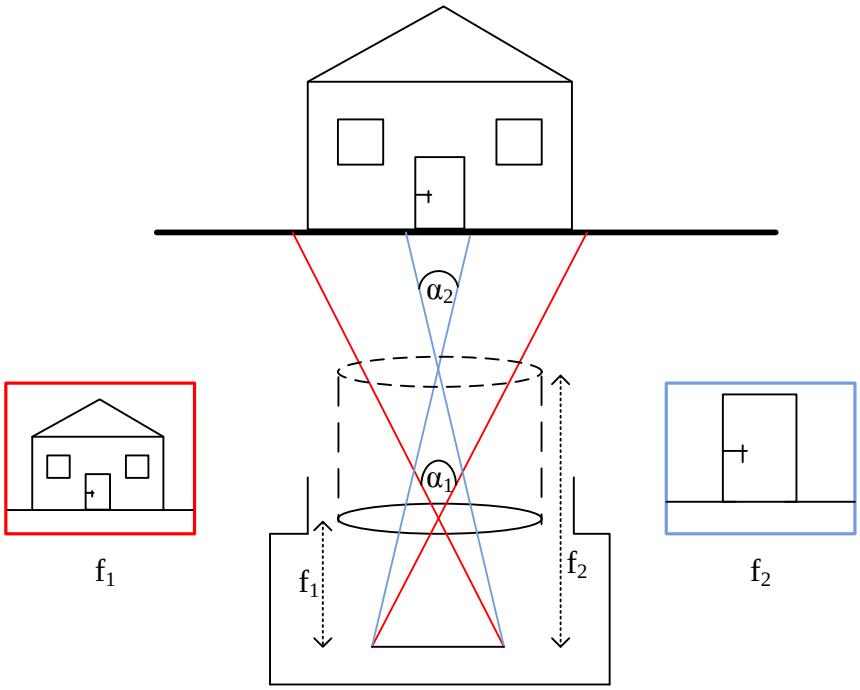
- ohniskovou vzdáleností a
- světelností [18].

6.4.1 Světelnost

Světelnost určuje množství světla, které projde objektivem a dopadne na světlocitlivý přijímač (dnes nejčastěji CCD čip). Tento parametr lze nastavit tzv. clonovým číslem.

6.4.2 Ohnisková vzdálenost

Nicméně o velikosti snímané scény rozhoduje ohnisková vzdálenost. Je to vzdálenost čočky od CCD snímače, viz kap. 6.1. Velké množství dnešních digitálních fotoaparátů a kamer umožňuje měnit ohniskovou vzdálenost objektivu (funkce zoom). Na obr. 6-4 je ukázána situace pro dvě velikosti ohniskové vzdálenosti. Protože je velikost snímače konstantní, rozhoduje o rozměrech snímané scény přímo ohnisková vzdálenost [18].



Obr. 6-4: Záznam scény pro dvě hodnoty ohniskové vzdálenosti

Klíčový je úhel, který svírají krajní paprsky vycházející ze zobrazovaného objektu, procházející středem objektivu a dopadající na okraje snímače (viz Obr. 6-4) tzv. zorný úhel. Téměř všechny dnes vyráběné snímače nemají stejnou šířku a výšku. Poměr jejich rozměrů odpovídá poměrům používaným v televizní technice, tedy 4:3 nebo novější 16:9. Zorný úhel je tedy pro vertikální a horizontální směr různý. Pro výpočet zorného úhlu α platí vztah

$$\alpha = 2 \cdot \mathrm{arctg}\left(\frac{d}{2 \cdot f}\right), \tag{6.3}$$

kde d označuje rozměr snímače, f ohniskovou vzdálenost [18].

7 Implementované operátory

Tato práce má za cíl sledovat pohyb objektu v nasnímané scéně. Pohyb je popsán dvěma parametry: směrem a rychlostí, příp. typem. Tato práce se zaměřuje na studium pohybu translačního. Dalším zjednodušením je omezení se na pohyb v rovině rovnoběžné s obrazovou rovinou. Pro vyznačení trajektorie pohybu byla zvolena metoda kumulovaného rozdílového snímku. Měření rychlosti je prováděno pomocí detekce polohy objektu ve snímcích, přičemž způsoby detekování jsou různé. Vše je implementováno v prostředí RapidMiner do projektu ImageProcessing jako jednotlivé operátory. Nyní k jednotlivým operátorům.

7.1 Kumulovaný rozdílový snímek (AccumulativeDifferenceImage)

7.1.1 Popis implementace

Teoreticky je tato metoda popsána v kapitole 2.3.1. Na obr. 7-3 je zobrazen vývojový diagram implementovaného algoritmu. Prvním krokem je načtení referenčního snímku $r(x, y)$, se kterým bude sekvence porovnávána. Pro správnou funkčnost je třeba načíst snímek scény bez pohybujícího se objektu.

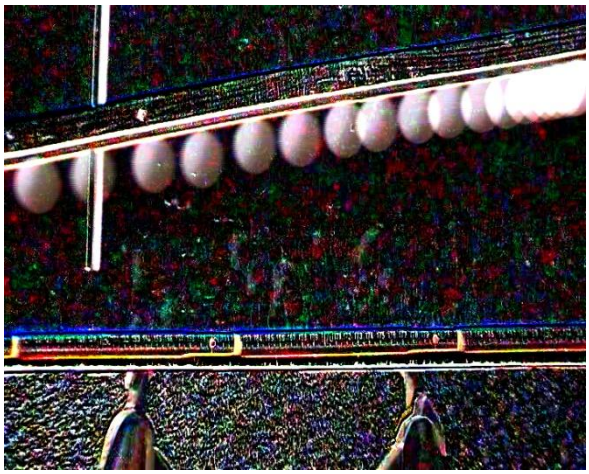
V další fázi již probíhá načítání snímků sekvence. Pro i-tý snímek sekvence $k_i(x, y)$ je vypočten rozdílový snímek $d(x, y)$ oproti referenčnímu snímku $r(x, y)$. Algoritmus zpracovává barevné snímky, tudíž rozdílový snímek je počítán ve všech 3 kanálech modelu RGB. Z důvodu možných výsledků mimo interval (v tomto případě záporných hodnot) je nutné pro tyto nežádoucí hodnoty ukládat krajní hodnotu intervalu (zde 0). Pro zjednodušení je uveden výpočet pro jeden kanál, pro ostatní bude probíhat obdobně. Matematické vyjádření výpočtu pro kanál červené bravy

$$d_R(x,y) = \begin{cases} 0 & \text{pro } k_{iR}(x,y) - r_R(x,y) < 0, \\ k_{iR}(x,y) - r_R(x,y) & \text{jinak.} \end{cases} \tag{7.1}$$

Tento rozdílový snímek je třeba přičíst v každém kroku i k výslednému snímku o(x, y). Výsledný snímek může být inicializován ve dvou variantách. Buď jako snímek nulový, tedy s hodnotami všech pixelů nulové hodnoty. Nebo může být načten referenční snímek a stopa pohybu bude zobrazena v původní scéně. Přičítání opět probíhá ve všech 3 kanálech R, G, B. Nyní může dojít k překročení maximální hodnoty intervalu pro každý kanál (0-255), proto je nutné pro hodnoty vyšší než 255 ukládat hodnotu 255. Pro červený kanál výsledného snímku bude v každém kroku i platit

$$o_R(x,y) = \begin{cases} 255 & \text{pro } o_R(x,y) + d_R(x,y) > 255, \\ o_R(x,y) + d_R(x,y) & \text{jinak.} \end{cases} \tag{7.2}$$

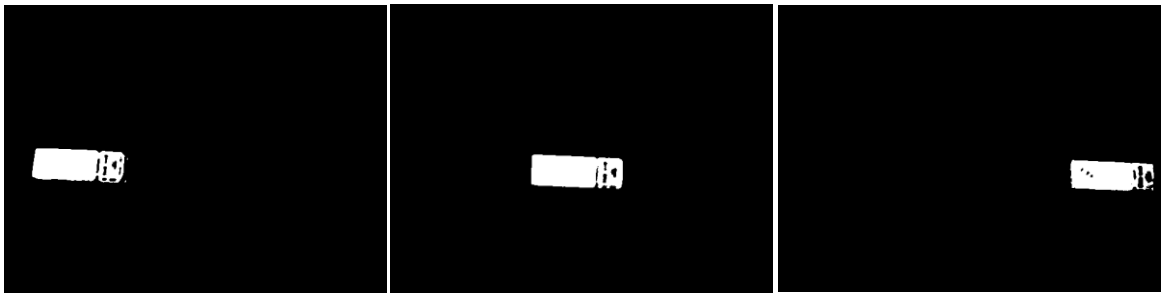
Ve výsledném snímku $o(x, y)$ je zaznamenána postupně poloha pohybujícího se objektu. Je tedy zobrazena imaginární stopa tělesa scénou. Na obr. 7-1 je zobrazen příklad výstupního kumulovaného snímku. Kromě vyznačení stopy se na něm objevují i nežádoucí oblasti nenulových hodnot pixelů. Ty jsou způsobeny na souvislých plochách šumem, na hranách nepřesnostmi snímacího zařízení. Částečně jsou tyto vlivy odstraněny částečným prahováním nízkých složek. Nicméně přičítáním rozdílových snímků dochází ke kumulaci nízkých hodnot, které se poté viditelně projeví ve výsledném snímku.



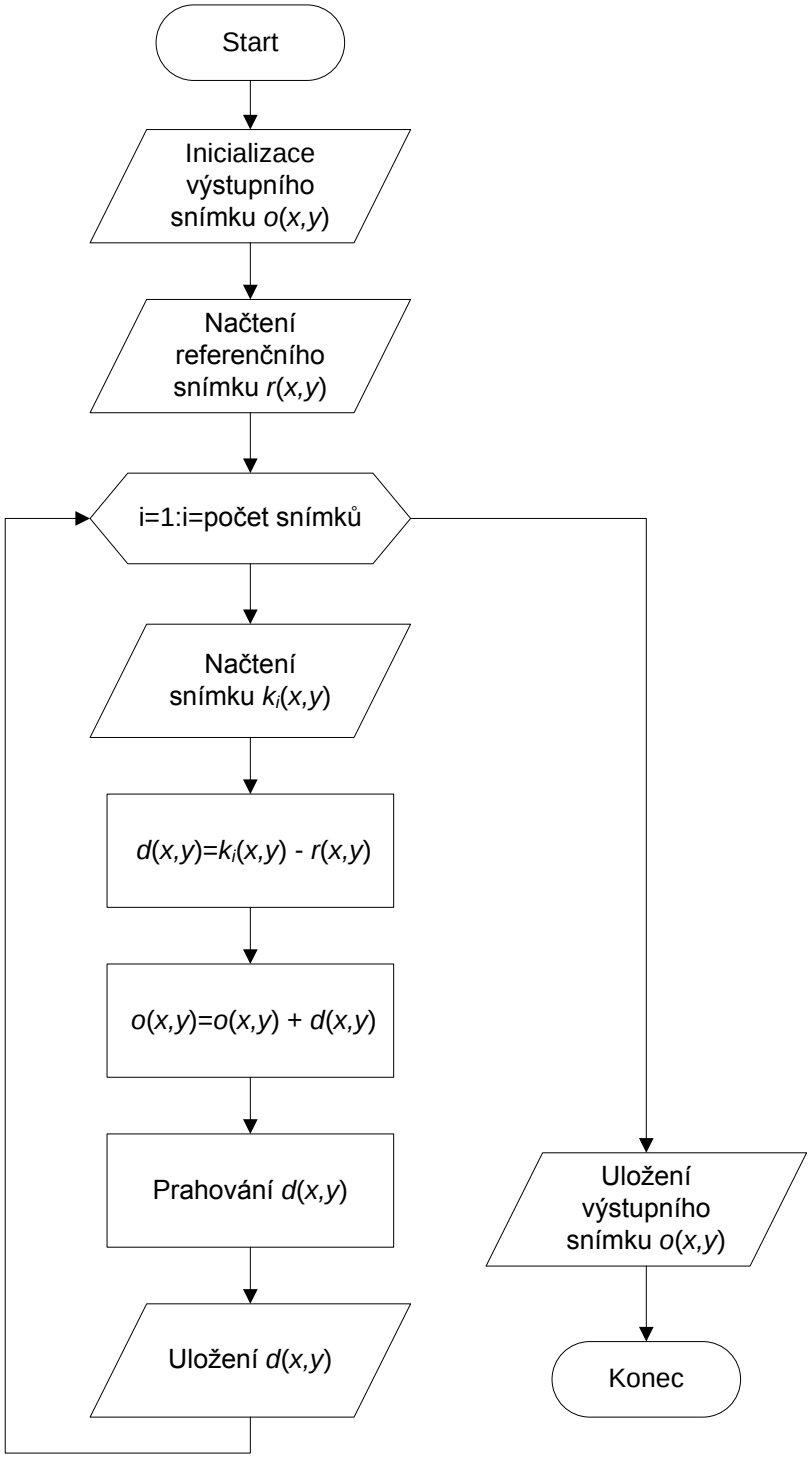
Obr. 7-1: Ukázka výstupu kumulovaného rozdílového snímku

7.1.2 Další funkce

V rámci tohoto operátoru je také využíváno částečných rozdílových snímků pro další zpracování. Jako další výstup je v každém kroku rozdílový snímek prahován. Tento prahovaný snímek je vytvářen dle klasického předpisu (2.2). Hodnota prahu ε je explicitně nastavitelná. Její velikost je třeba nastavit s uvážením míry odlišnosti pohybujícího se objektu od prostředí scény a velikosti šumu v obraze (viz statistické metody, kap. 2.2.5). Při vhodné velikosti jsou v každém kroku i vytvořeny masky korespondující s obrysy hledaného objektu. Tyto masky poté budou využity k výpočtu rychlosti pohybu. Na obr. 7-2 jsou ukázky masek pro sérii snímků video sekvence č. 1 (označení viz 8.4).



Obr. 7-2: Ukázka výstupních rozdílových snímků



Obr. 7-3: Vývojový diagram operátoru Accumulative Difference Image

7.2 Optický tok (OpticalFlow)

7.2.1 Popis implementace

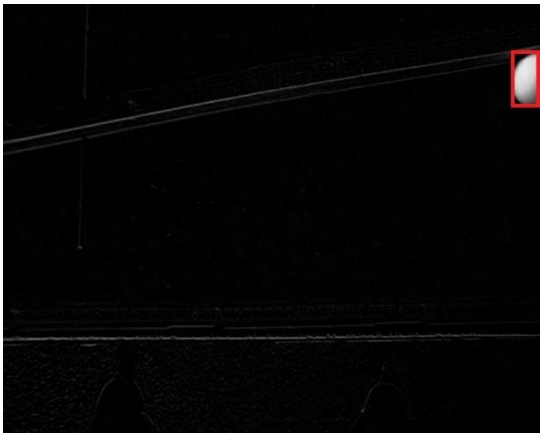
Základní myšlenkou tohoto operátoru je využít pro výpočet rychlosti pohybujícího se objektu vektorů pohybu. Bloková struktura postupu algoritmu je na obr. 7-5. Nyní k jednotlivým blokům podrobněji.

Vstup

Operátor je navržen pro zpracování černobílých snímků, proto vstupní sekvence musí být převedena do šedotónového formátu. Všechny snímky by měly mít také stejné rozměry.

Vyznačení objektu v pohybu

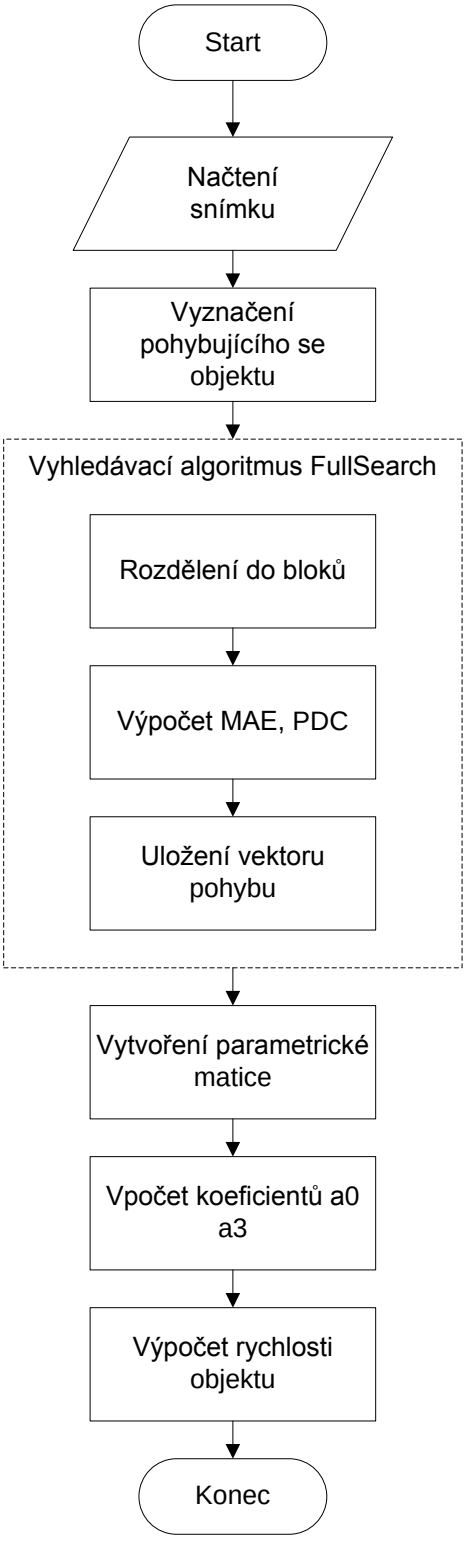
Základním problémem při každé segmentaci obrazu je nalezení hledaného objektu. V tomto případě je hledaným objekt v pohybu. Předpokladem pro jeho správné nalezení je tedy statická scéna prostředí, navíc s co nejnižší hodnotou šumu v obraze. Potom lze jako hledaný objekt označit oblast nenulových pixelů rozdílového snímku obrazu sekvence a obrazu statického prostředí. Tedy postupem obdobným jako při vytváření kumulovaného rozdílového snímku. Tento soubor pixelů je označen jako tzv. oblast zájmu (Region of Interest). Další výpočty budou probíhat pouze pro oblast zájmu, příp. pro jeho blízké okolí. Na obr. 7-4 je příklad rozdílového snímku použitého pro nalezení hledaného objektu. Pro názornost je v něm uměle vyznačena oblast zájmu, jenž objekt obklopuje.



Obr. 7-4: Vyznačení oblasti zájmu

Vyhledávací algoritmus

Nyní je tedy objekt nalezen a dalším logickým krokem je hledání tohoto objektu v následujícím snímku. To probíhá podle postupů blokových vyhledávacích algoritmů, viz kapitola 4. Nejprve je oblast zájmu rozdělena do bloků o velikosti 8x8 pixelů. Tyto rozměry jsou obvyklé a používané např. i ve standardu JPEG. Jako vyhledávací kritérium je použito MAE, viz část 5.1. Dále je implementován i výpočet PDC, popsaný v části 5.3. Použitým vyhledávacím kritériem je technika Full Search (viz 4.1). Jak je uvedeno v teorii, je tato technika výpočetně náročná, nicméně poskytuje nejpřesnější výsledky. Zde probíhá vyhledávání pouze pro oblast zájmu, tudíž je výpočetní náročnost značně snížena. Pro každý blok je tedy nalezen odpovídající blok v následujícím snímku. Změna polohy bloků je uložena do pole, které tvoří vektory pohybu pro jednotlivé bloky. Je tedy uložena informace o počtu pixelů, o který je blok posunut od původní polohy ve směru vertikálním i horizontálním.



Obr. 7-5: Posloupnost operací v operátoru Optical Flow

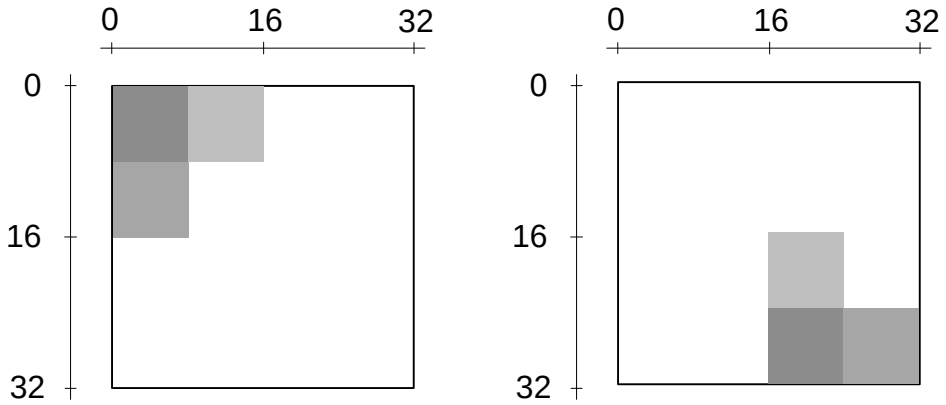
Parametrický popis optického toku

Získané vektory pohybu jsou použity pro parametrické vyjádření optického toku, viz část 1.4. Pro přehlednost je uvedena znovu dvojice rovnic pro parametrický popis:

$$u(x,y) = a_0 + a_1 x + a_2 y \tag{7.3}$$

$$v(x, y) = a_3 + a_4 x + a_5 y. \tag{7.4}$$

Z popisu významu jednotlivých koeficientů je patrné, že vektor pohybu pro jeden blok generuje vždy dvojici rovnic. Pro výpočet koeficientů a_0 – a_5, tedy postačí pouze 3 bloky. Ze složek vektorů pohybu *u*, *v* a hodnot souřadnic bloku v původním obraze *x*, *y* je vytvořena soustava rovnic. Tuto soustavu lze přepsat do maticového tvaru a z něj snadno vypočítat hledané koeficienty. Nutnou podmínkou je, aby rovnice byly lineárně nezávislé, neboli aby matice koeficientů nebyla singulární. Následuje ukázka funkce tohoto postupu pro uměle vytvořenou obrazovou sekvenci na obr. 7-6.



Obr. 7-6: Uměle vytvořená sekvence snímků

Popis funkčnosti

Obrázky jsou z důvodu přehlednosti zvětšeny. Osy podél hranic ukazují počet pixelů. Obrázky tedy mají velikost 32x32 pixelů, jednotlivé bloky o konstantní hodnotě jasu 8x8.

Nyní k parametrickému popisu popisu bloků. Souřadnice bloku je dána pozicí jeho horního levého rohu. Poloha jednotlivých bloků je tedy:
blok č. 1: x_1=0, y_1=0,
blok č.2: x_1=8, y_1=0,
blok č.3: x_1=0, y_1=8.

Po přemístění bloků do nové pozice v obr. 7-6 (vpravo) mají nové polohy souřadnice:
blok č. 1: x_2=16, y_2=24,
blok č.2: x_2=16, y_2=16,
blok č.3: x_2=24, y_2=24.

Z těchto nových poloh plynou odpovídající vektory pohybu ***r***(*u, v*), kde *u* udává změnu polohy ve směru osy *x*, *v* ve směru osy *y*, tedy
blok č.1: *u*=16, *v*=24,
blok č.2: *u*=8, *v*=16,
blok č.3: *u*=24, *v*=16.

Dosazení hodnot původních souřadnic bloků a složek vektorů pohybu do rovnic výše vytváří následující soustavu rovnic:

$$16 = a_0 + 0 \cdot a_1 + 0 \cdot a_2$$

$$24 = a_3 + 0 \cdot a_4 + 0 \cdot a_5$$

$$8 = a_0 + 8 \cdot a_1 + 0 \cdot a_2$$

$$16 = a_3 + 8 \cdot a_4 + 0 \cdot a_5$$

$$24 = a_0 + 0 \cdot a_1 + 8 \cdot a_2$$

$$16 = a_3 + 0 \cdot a_4 + 8 \cdot a_5$$

Řešením této soustavy jsou hodnoty koeficientů a_0 – a_5:

a_0 =16, a_1 = -1, a_2 =1, a_3 =24, a_4=-1, a_5=-1.

Významy koeficientů jsou uvedeny v teorii v části 1.4. Hodnota a_0 tedy znamená posun o 16 pixelů v ose *x*, a_3 o 24 pixelů po ose *y*. Dále rozdíl hodnot a_4 a a_2 je nenulový, což značí rotaci soustavy bloků

$$a_4 - a_2 = -2.$$

Záporná hodnota ukazuje na rotaci doleva, což odpovídá skutečnosti.

Pohyb je tedy složen ze dvou základních složek, translačního a posuvného. Obecně záleží na pořádí, ve kterém jsou tyto dva pohyby prováděny. Popsaný příklad ukazuje nejprve na pohyb posuvný a poté rotační, což je patrné z hodnot koeficientů a_0 a a_3. Bylo toho dosaženo zvolením hodnot souřadnic bloků v novém obraze. Souřadnice byly voleny opět jako levý horní roh bloku. Pokud by jako souřadnice byly voleny body, do nichž se dostal původní levý horní roh každého bloku, bylo by pořadí pohybů opačné, nejprve tedy rotační a poté translační. Dokazují to také hodnoty koeficientů a_0 a a_3, jejichž velikosti jsou a_0 =16, a_3 =32. Odpovídá to tedy představě rotace souboru bloků doleva kolem bodu [0,0] (bloky se dostávají do záporných hodnot v ose y) a následném posunu o příslušné hodnoty do pozic korespondujících s polohou na obr. 7-6 (vpravo).

Koeficienty a_0 – a_5 popisují základní parametry pohybu. Hodnoty a_0 a a_3 lze výhodně použít pro výpočet změny polohy, příp. rychlosti změny, ostatní koeficienty pro popis typu a povahy pohybu.

Hodnoty koeficientů a_0 a a_3 jsou tedy použity pro výpočet rychlosti objektu. Tato funkce je řešena samostatně v operátoru SpeedMeasuring, a proto bude popsána dále v kapitole 7.3 věnující se tomuto operátoru.

Zhodnocení funkčnosti operátoru

Přes počáteční úspěšné testování operátoru na jednoduchých uměle vytvořených obrázcích, se ukázal tento postup měření rychlosti jako neefektivní. Hlavní problém nastává v nalezení skutečně si odpovídajících bloků. Citlivost výpočtu soustavy rovnic na špatně nalezená data se ukázala jako významná. Automatizované odstranění nežadoucích dat bylo možné provést jen

částečně a to tak, že pro výpočet byly brány bloky jen s nejvyšším odpovídajícím charakterem. Nicméně ani tohle nemohlo obtíže s nalézáním neodpovídajících si bloků vyřešit. Problém totiž tkví již v pořízených videosekvencích. V reálné scéně lze jen velice těžce vytvořit stejnoměrné osvětlení scény. Různá intenzita světla má pak za následek, že blok snímku popisující část pohybujícího se objektu má v následujícím snímku jinou jasovou hodnotu pro stejnou část objektu. Operátor tedy není možné v této základní podobě pro měření rychlosti nebo určení typu pohybu použít. Pro zlepšení funkčnosti by ho bylo třeba rozšířit o předzpracování odstraňující nehomogenitu osvětlení. Tento proces není nikterak triviální a nesouvisí přímo s problematikou této diplomové práce, proto nebyl vypracován.

7.3 Výpočet rychlosti (SpeedMeasuring)

Tento operátor slouží k výpočtu rychlosti objektu pohybujícího se po scéně. Vstupem jsou binární masky vyznačující polohu objektu. Ty jsou vytvořeny předchozím zpracováním některou segmentační variantou. Přesnost výpočtu tedy závisí především na kvalitě segmentace.

7.3.1 Popis implementace

V první fázi jsou načteny vždy 2 snímky a v každém z nich je hledána poloha objektu. Protože se jedná již o masky, degraduje tento problém pouze na vyhledání hranic světlé oblasti pixelů. Prostým odečtením krajních souřadnic oblasti v jedné a druhé masce je vypočten rozdíl poloh objektu v obraze ve směru osy *x* i osy *y*. Tyto hodnoty se stávají vstupními veličinami pro určení rychlosti.

Hlavní část operátoru tvoří funkce, která na základě výše popsaných změn souřadnic určí reálnou rychlost objektu ve scéně. K tomu jsou však třeba další parametry, které je třeba zadat explicitně dle podmínek, za jakých byla videosekvence vytvářena.

7.3.2 Vstupní parametry

Důležitou součást tvoří vlastnosti snímacího zařízení. Především je třeba znát zorný úhel objektivu v obou směrech. Ten se běžně u kamer neuvádí, proto je počítán dle vztahu (6.3). V něm jsou jako proměnné ohnisková vzdálenost a rozměr čipu.

Většina dnešních kamer má funkci zoom, která umožňuje měnit ohniskovou vzdálenost, proto je u kamer zapsán rozsah hodnot ohniskových vzdáleností. Správné určení je možné zajistit užíváním těchto krajních hodnot nebo, pokud to přístroj umožňuje, z údaje uváděném při konkrétní hodnotě zoomu.

Rozměr čipu není u kamer uváděn přímo ve velikosti šířky a výšky CCD snímače, ale je uváděna v poměru čísel např. 1/2, 2/3 atd. Jednotkou je palec. Toto číslo udává přibližně 3/2 skutečné úhlopříčky CCD snímače. Z její velikosti je již snadnou trigonometrickou úlohou zjištění výšky a šířky snímače. Je totiž znám poměr stran snímače. Ten bývá v klasické podobě standardního televizního formátu 4:3. V dnešní době se však již mohou vyskytovat i širokoúhlé snímače s poměrem stran 16:9. Pro korektní výpočet je tedy třeba znát i to, zda se jedná o klasický nebo širokoúhlý snímač.

Dalším potřebným parametrem je rozlišení vytvořeného záznamu. Tedy velikost šířky a výšky snímků video sekvence udávané v pixelech. Důvodem je výše zmíněný postup výpočtu rozdílu polohy masek, který je udáván také v pixelech.

Mezi vstupní parametry zadávané explicitně pro konkrétní obrazovou sekvenci tedy patří

- Parametry snímacího zařízení:
  - velikost CCD snímače [in],
  - ohnisková vzdálenost [mm],
  - poměr stran (4:3 nebo 16:9),
- rozlišení snímků:
  - výška snímku [px],
  - šířka snímku [px],
- časová informace:
  - počet snímků za 1 sekundu [s^{-1}],
- vlastnost scény:
  - vzdálenost objektivu od scény [m].

7.3.3 Odvození vztahu pro výpočet rychlosti

Nyní bude popsáno odvození vztahu pro výpočet reálné rychlosti ze vstupních obrazových dat a zadaných parametrů.

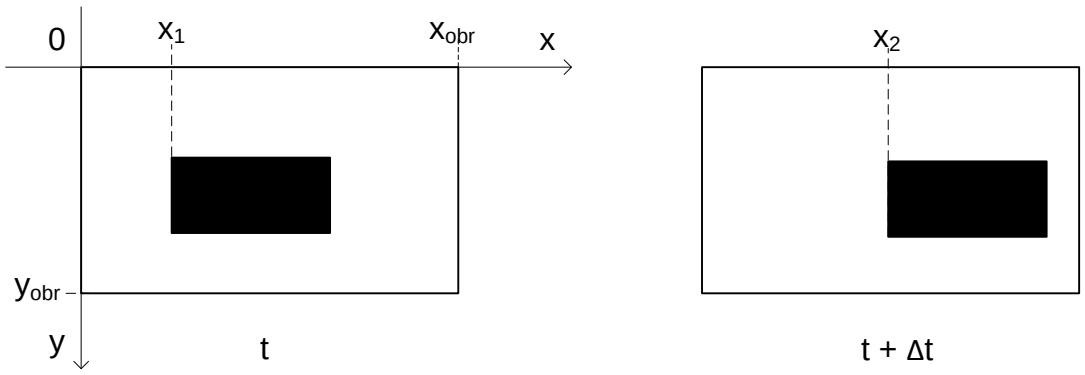
Pro velikost rychlosti *v* objektu platí známý vztah

$$v = \sqrt{{v_x}^2 + {v_y}^2}, \tag{7.5}$$

kde v_x udává rychlost ve směru osy *x*, v_y ve směru osy *y*. Výsledná rychlost je tedy složena ze dvou dílčích složek rychlostí, z nichž jedna udává horizontální a druhá vertikální změnu polohy za jednotku času. Další odvození výpočtu se zaměří na horizontální složku v_x. Postup pro výpočet rychlosti v_y je obdobný.

Nejprve je třeba se zaměřit na vstupní obrazová data. Na obr. 7-7 jsou zobrazeny dva binární snímky, tedy masky, ve kterých je tmavou barvou vyznačen objekt v pohybu. Pro přehlednost jsou zde barvy snímků invertované – operátor chápe jako objekt bílou oblast pixelů. Je známo, že rychlost je definována jako změna polohy za jednotku času

$$v = \frac{\mathrm{d}x}{\mathrm{d}t}. \tag{7.6}$$



Obr. 7-7: Popis změny polohy objektu v souřadném systému

Změna polohy je v tomto případě rozdíl v souřadnicích osy *x* v obou snímcích. Časovou informaci poskytuje údaj o době mezi pořízením obou snímků. Z anglické literatury se vžilo označení převrácené hodnoty této veličiny fps (frames per second), tedy počet snímků za 1 sekundu, v dalším postupu bude tato veličina označena *n*. Pro tuto „obrazovou" rychlost v_{ox} lze odvodit z předchozí úvahy a vztahu (7.6) vztah

$$v_{ox} = \frac{x_2 - x_1}{\frac{1}{n}}. \tag{7.7}$$

Tato rychlost popisuje časovou změnu pixelů, takže jednotkou je px/s. Sama veličina nemá velký praktický význam, je jen mezikrokem pro další výpočet.

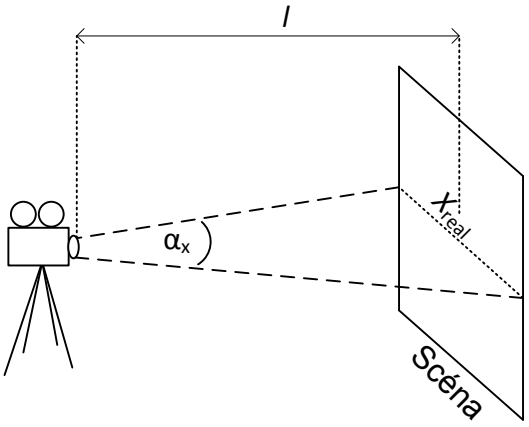
Nyní je důležité zjistit, jaké jsou rozměry reálné scény zabírané každým snímkem. Úloha lze také položit tak, že je třeba zjistit, jak velká oblast je reprezentována jedním pixelem. Je tedy hledán poměr mezi šířkou snímku x_{obr}(px) a reálnou šířkou zobrazované scény X_{real} (m), symbolicky

$$\frac{X_{\text{real}}}{x_{\text{obr}}}. \tag{7.8}$$

Pro určení výsledné reálné rychlosti už jen postačí vynásobit „obrazovou" rychlost v_{ox} tímto poměrem

$$v_x = \frac{X_{\text{real}}}{x_{\text{obr}}} \cdot v_{0x}. \tag{7.9}$$

Pro výpočet hodnoty X_{real} je třeba si uvědomit zavedené zjednodušení pohybu objektu po rovině rovnoběžné s obrazovou rovinou. V obecném případu by určení šířky nebylo možné. Zde je šířkou scény vzdálenost, kterou vytínají paprsky zorného úhlu na myšlené rovině, po které se objekt pohybuje, viz obr. 7-8.



Obr. 7-8: Popis veličin použitých pro další výpočty

Jak je z obrázku i popisu patrné, změření této hodnoty by bylo velice obtížné a nepřesné. Efektivnější se jeví využití znalosti zorného úhlu a pohodlněji měřitelné vzdálenosti kamery od roviny. Význam veličin je zřejmý z obr. 7-8. Z nákresu lze vypozorovat, že velikost X_{real} bude rovna pro popsané veličiny pomocí goniometrických funkcí

$$X_{\text{real}} = 2 \cdot l \cdot \tan \alpha_x. \tag{7.10}$$

Zorný úhel je počítán dle vztahu (6.3).

Nyní nezbývá než dosadit do vztahu (7.9) za X_{real} (7.10) a za v_{ox} (7.7) a výsledkem je vztah pro výpočet reálné rychlosti v_x ze zjištěných polohových změn v obraze a zadaných parametrů

$$v_x = \frac{2 \cdot l \cdot \tan \alpha}{x_{\text{obr}}} \cdot \frac{x_2 - x_1}{\frac{1}{n}}. \tag{7.11}$$

8 Testování operátorů

8.1 Použitá záznamová zařízení

8.1.1 Digitální kamera

Pro účely testování byla použita domácí kamera značky Samsung, model VP-MX20. Tento typ vytváří videa ve formátu H.264 BP s rozlišením záznamu 720x576. Kamera obsahuje funkci zoom, ohnisková vzdálenost je tedy nastavitelná, konkrétně v rozmezí 2,3 – 78,2 mm. Pro záznam videí byla použita nejnižší hodnota ohniskové vzdálenosti. CCD čip je výrobcem udáván jako širokoúhlý o rozměru 1/6“. Kamera poskytuje možnosti automatického ostření a nastavování clonového čísla, což je však pro záznam testovacích sekvencí nežádoucí. Doostřování vede k mírné změně ohniskové vzdálenosti, ta se projeví v záznamu jako skokové přiblížení nebo oddálení scény. Automatická korekce clonového čísla způsobuje změnu hodnoty vstupujícího světla přes čočku, což má za následek změnu velikosti průměrného jasu v jednotlivých snímcích sekvence. Obě tyto funkce byly tedy vypnuty a příslušné parametry nastaveny pro optimální podobu videa manuálně. Parametry kamery explicitně zadávané pro výpočet rychlosti objektu jsou pro obě pořízená videa tyto

- ohnisková vzdálenost f=2,3 mm,
- velikost čipu 1/6“,
- rozlišení záznamu:
    - šířka snímků 720 px,
    - výška snímků 576 px.

8.1.2 Digitální fotoaparát

Pro ověření funkčnosti byl využit ještě další přístroj, a to fotoaparát značky Samsung, model Cyber630. Tento typ umožňuje i záznam videí o nízkém rozlišení. Oproti výše popisované kameře neobsahuje vysokou podporu videa, proto nastavení neumožňuje mnoho možností. Zde jsou uvedeny parametry, za kterých byla video sekvence pořizována:

- ohnisková vzdálenost f=5,8 mm,
- velikost čipu 1/2,4 “,
- rozlišení záznamu:
    - šířka snímků 640 px,
    - výška snímků 480 px.

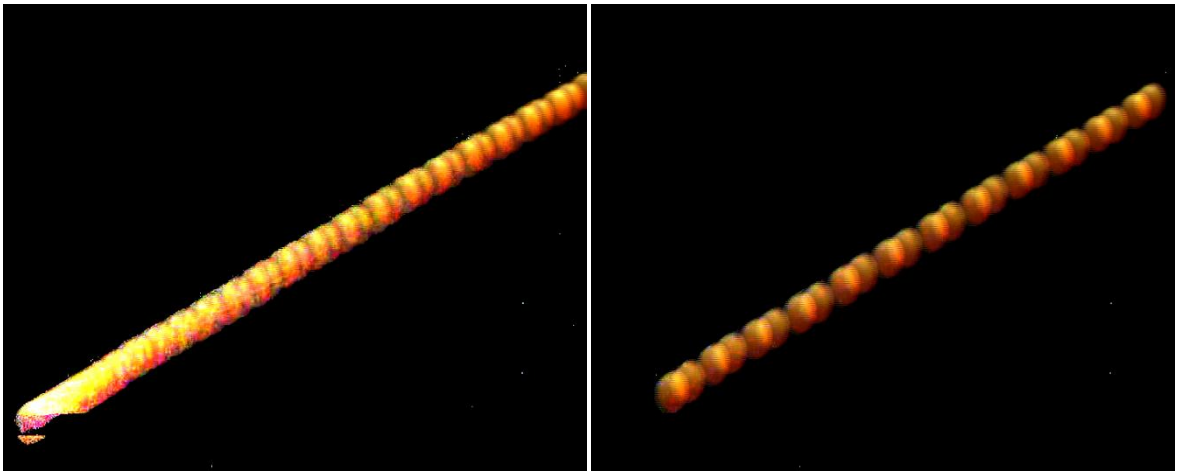
8.2 Způsob zpracování

Video sekvence byly zkoumány jako sekvence obrázků. Pro zpracovávání bylo tedy nejprve nutné z videí vytvořit sekvence snímků. Pro účely testování byl pro tuto funkci využit freewarový program Free Video to JPG Converter 1.8.7. Ten umožňuje uložit z video sekvence různý počet snímků za různý časový údaj. Informace o počtu snímků vyčtených z video sekvence poslouží v dalším zpracování pro určení snímkové frekvence.

8.3 Výsledky pro operátor AccumulativeDifferenceImage

8.3.1 Různé velikosti rychlostí objektu

Kumulovaný rozdílový snímek vytváří záznam o poloze objektu v jednotlivých snímcích video sekvence. Obsahuje tedy záznam trajektorie pohybu tělesa scénou. Z podoby kumulovaného snímku lze vypozorovat i relativní velikost rychlosti objektu. Na obr. 8-1 jsou zobrazeny 2 kumulované snímky též scény pro dvě různé rychlosti objektu.

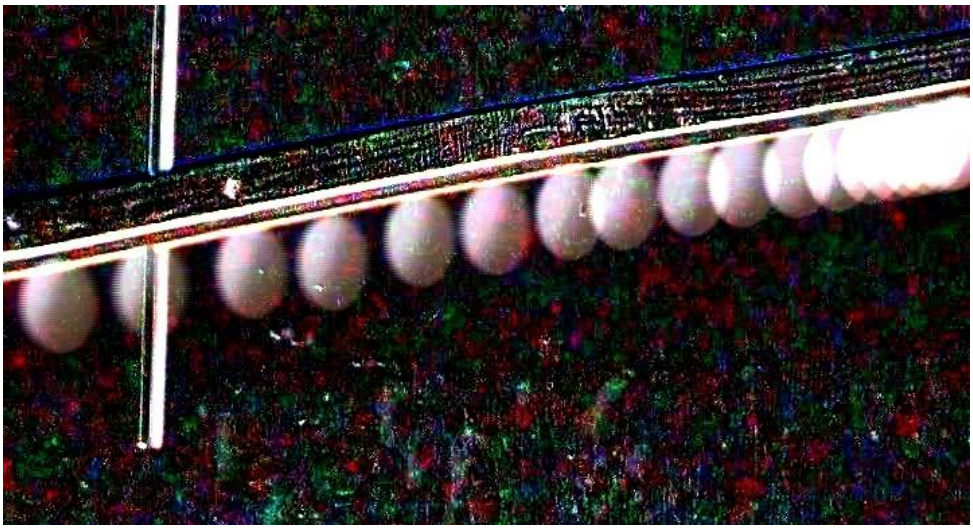


Obr. 8-1: Kumulovaný rozdílový snímek pro 2 různé rychlosti objektu

Z optického porovnání snímků je patrné, že objekt se v záznamu videa, z něhož byl vytvářen obrázek vlevo, pohyboval nižší rychlostí, než je tomu v případě pravého snímku. Na snímku vlevo je zřetelná vyšší hustota stop objektu.

8.3.2 Nerovnoměrný pohyb objektu

Podobně je možné z kumulovaných snímků vyčíst nekonstantnost rychlosti objektu. V takových případech dochází ke zřeďování nebo zhušťování stop v kumulovaném snímku. Tuto situaci zachycuje kumulovaný snímek na obr. 8-2. Z něj je patrná vysoká hustota stop objektu napravo, tedy nižší rychlost objektu v pravé části scény, a směrem k levé části snímku postupné vymizení překryvu stop a zvětšování jejich vzájemné vzdálenosti charakterizující zvyšování rychlosti objektu.



Obr. 8-2: Kumulovaný rozdílový snímek pro nerovnoměrný pohyb

8.4 Podoba testovaných video sekvencí

Pro testování funkčnosti operátoru SpeedMeasuring byly natočeny tři video sekvence, dvě domácí kamerou popsanou v části 8.1.1, třetí fotoaparátem s funkcí záznamu videa, viz 8.1.2.

Scéna všech videí byla vytvořena tak, aby se na ní pohyboval jen jediný, a to zkoumaný objekt. Byla snaha o vytvoření konstantního osvětlení během trvání záznamu, nicméně některé vlivy okolí, jakými jsou odrazy lesknoucích ploch nebo stín pohybujícího se objektu, nebylo možné odstranit.

První sekvencí (v dalším textu označována jako č. 1) je záznam pohybu modelu nákladního auta po hraně stolu. Jedná se pouze o pohyb ve vodorovném směru, jak ukazuje několik snímků video sekvence na obr. 8-3. Vzdálenost objektu od kamery byla 91 cm.

...

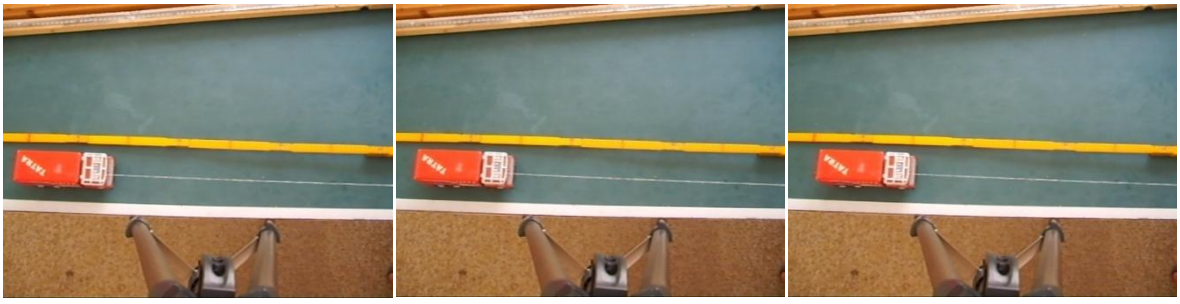
Obr. 8-3: Ukázka prvních a posledních tří snímků video sekvence č. 1

Druhým testovaným videem je pohyb míčku na stolní tenis po připravené drážce. Tentokrát se objekt pohybuje z oblasti levého dolního rohu do pravého horního rohu, viz ukázka několika snímku sekvence na obr. 8-4. Vzdálenost míčku od objektivu kamery byla 78 cm.

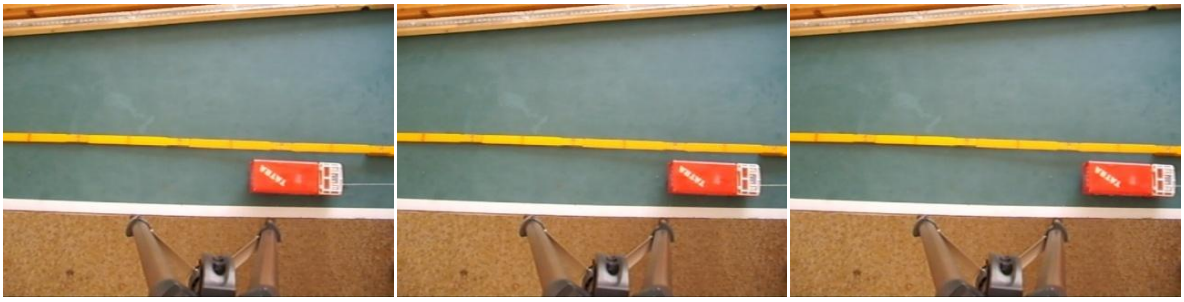
...

Obr. 8-4: Ukázka prvních a posledních tří snímků video sekvence č. 2

Třetí testované video, vytvořené fotoaparátem, zachycuje záznam pohybu modelu automobilu přemisťovaného z levé části scény do pravé, viz obr. 8-5. Model byl vzdálen od objektivu 75 cm.



...



Obr. 8-5: Ukázka prvních a posledních tří snímků video sekvence č. 3

8.5 Výsledky testů operátoru SpeedMeasuring

8.5.1 Srovnání skutečné a vypočtené rychlosti

Z popisu fungování operátoru SpeedMeasuring plyne, že rychlost je počítána vždy mezi dvěma po sobě jdoucími snímky. Z těchto hodnot lze vypozorovat snadno trend vývoje velikosti rychlosti v průběhu sekvence. Hodnotu průměrné rychlosti je však třeba dopočítat ručně.

Základním parametrem, dle kterého je možné posoudit správnou funkčnost operátoru, je porovnání výsledků vypočtených rychlostí se skutečnou rychlostí objektu. Skutečná rychlost byla počítána dle vztahu (7.6). Vzdálenost, kterou objekt urazil na záznamu videa, byla ručně naměřena. Doba, za kterou tak bylo učiněno, byla zjištěna z časových údajů záznamu.

Video sekvence č. 1

V případě video sekvence č. 1, pohybujícího se modelu z bočního pohledu, se zjevně jedná o pohyb téměř výhradně ve vodorovném směru. Ze dvou složek vektoru rychlosti v, viz vztah (7.5), bude tedy zaměřena pozornost na rychlost v_x. V grafu (viz obr. 8-6) jsou zobrazeny hodnoty rychlostí v_x a v_y v jednotlivých okamžicích záznamu. Z každé sekundy záznamu videa byly zkoumány 2 snímky, čemuž odpovídá počet bodů v grafu pro dobu trvání videa 17 s. V grafu je také vyznačena lineární aproximace průběhu rychlosti v_x, z níž je patrný vzestupný vývoj. Pro srovnání se skutečnou rychlostí je třeba vypočítat z těchto jednotlivých rychlostí rychlost průměrnou jako prostý aritmetický průměr. Pro průměrnou rychlost v_{px} tedy platí

$$v_{px} = \frac{1}{n} . \sum_{i=1}^{i=n} v_{xi}, \tag{8.1}$$

kde v_{xi} značí jednu „mezisnímkovou“ rychlost v_x a n udává počet vypočtených rychlostí.

Číselně je tato hodnota rovna

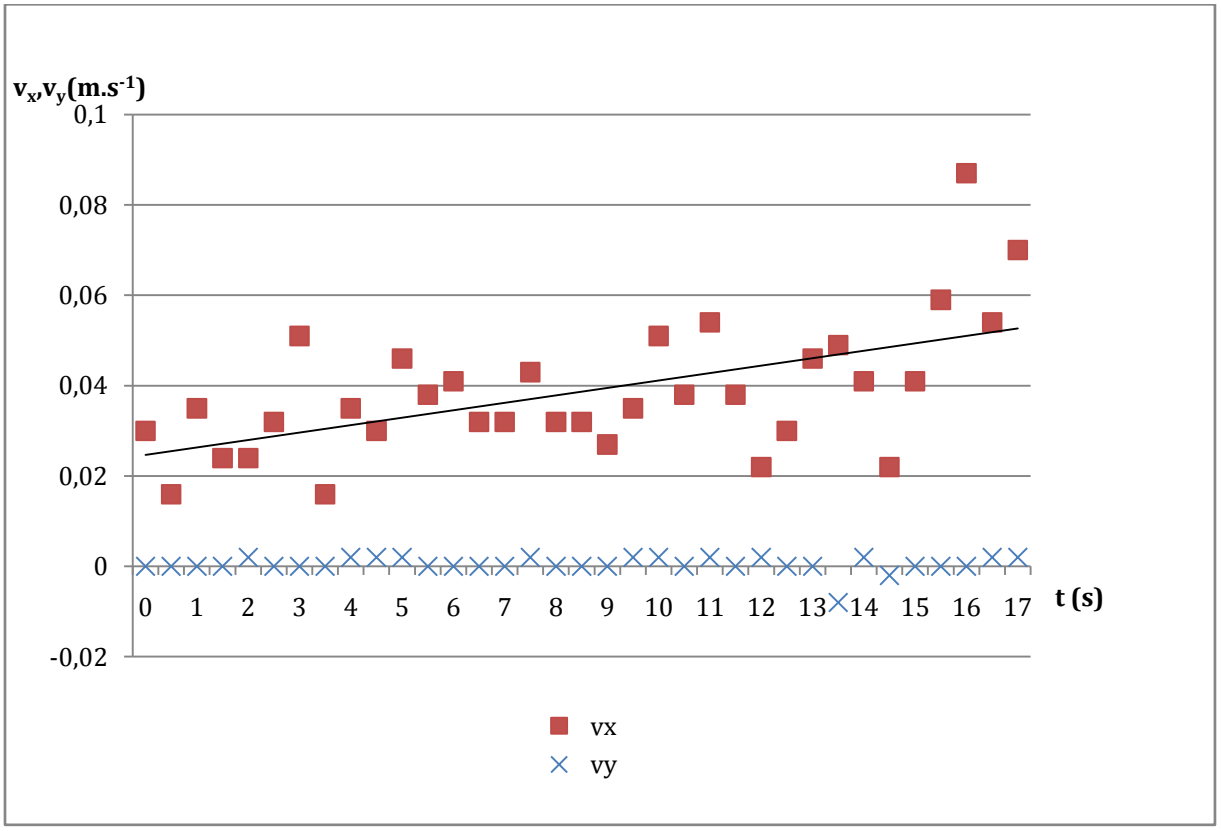
$$v_{px} = 0{,}038 \text{ m·s}^{-1}.$$

Skutečná rychlost je pro změřenou dráhu tělesa 0,6 m a dobu pohybu 17 s rovna

$$v_{px}^{'} = \frac{0{,}6}{17} = 0{,}035 \text{ m·s}^{-1}.$$

Rozdíl tedy činí

$$r = v_{px} - v_{px}^{'} = 0{,}003 \text{ m·s}^{-1}.$$



Obr. 8-6: Hodnoty rychlostí v_x a v_y pro video sekvenci č. 1

Video sekvence č. 3

Pro video sekvenci č. 3, pohyb modelu z horního pohledu jiným snímacím zařízením, je situace obdobná jako v předchozím případu. Opět je hlavní směr pohybu vodorovný, v kladném směru osy *x*. Naměřené výsledky jsou zobrazeny v grafu na obr. 8-7. Pro porovnání budou použity stejné vztahy jako pro video sekvenci č. 1. Průměrná rychlost v_{px} je rovna

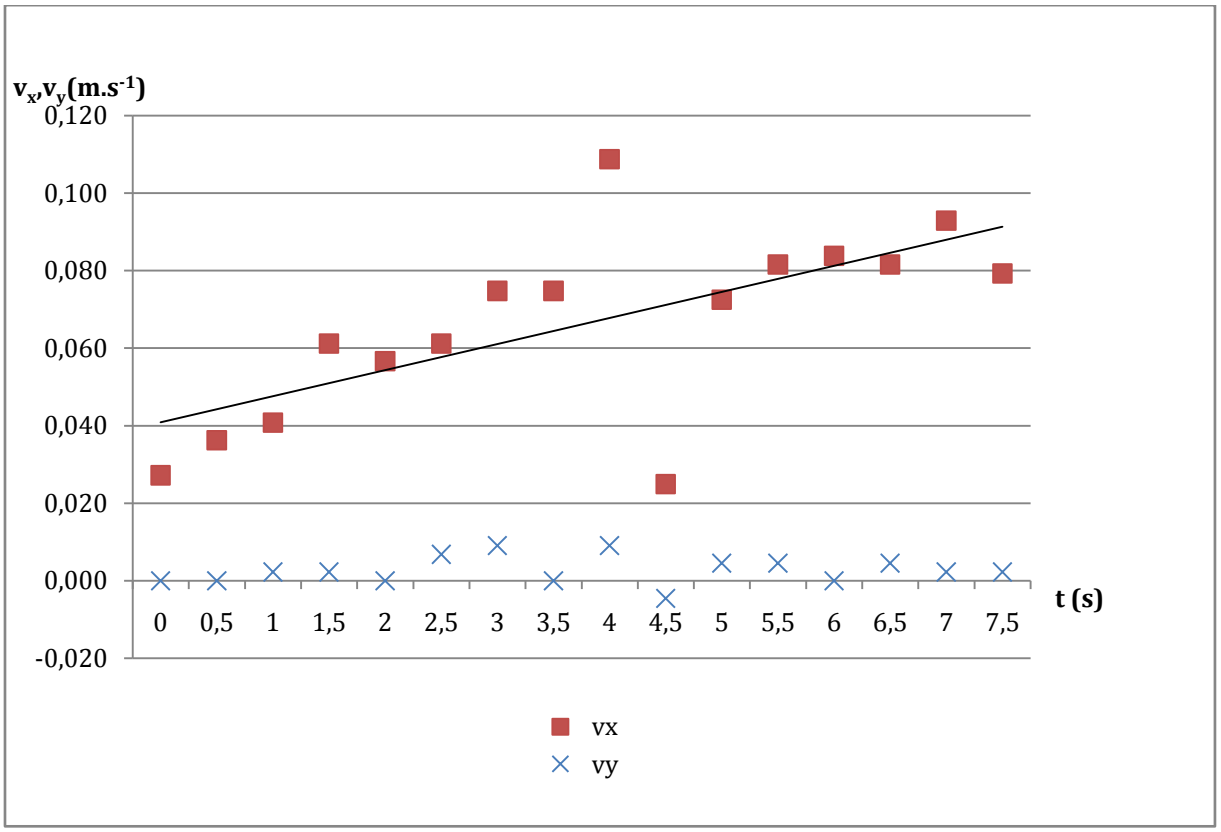
$$v_{px} = 0{,}066 \text{ m·s}^{-1}.$$

Vypočtená rychlost je pro hodnotu urážené vzdálenosti 0,54 m za dobu 8 s rovna

$$v_{px}^{'} = \frac{0{,}54}{8} = 0{,}064 \text{ m·s}^{-1}.$$

Rozdíl tedy činí

$$r = v_{px} - v_{px}^{'} = 0{,}002 \text{ m·s}^{-1}.$$



Obr. 8-7: Hodnoty rychlostí v_x a v_y pro video sekvenci č. 3

Video sekvence č. 2

Ve video sekvenci č. 2 mají již na pohyb objektu vliv obě složky rychlosti v – vodorovná i svislá. Pro popis pohybu tak jsou důležité hodnoty obou složek celkové rychlosti v. V duchu obvyklého popisu obrazových dat je položen počátek os do levého horního rohu. Pohyb směřuje z levého dolního rohu do pravého horního. Hodnoty vertikální rychlosti jsou tedy záporné. Ze zobrazených hodnot na obr. 8-8 je patrné mírné klesání velikostí v obou směrech. Pro porovnání se skutečnou rychlostí bude tentokrát použita průměrná celková rychlost vypočtená dle vztahu pro celkovou rychlost v (7.5) a vztahu pro průměrnou rychlost (8.1). Velikost průměrné celkové rychlosti pro vypočtené hodnoty je

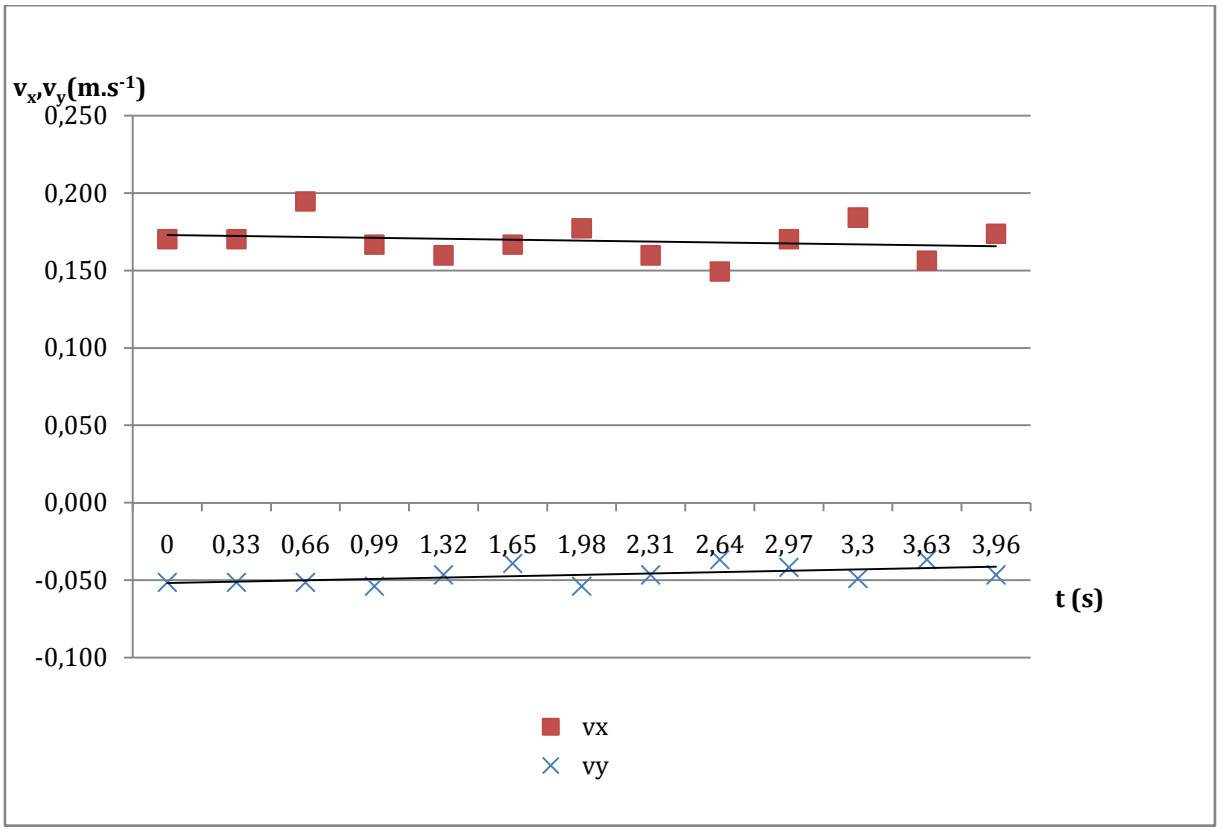
$$v_p = 0{,}176 \text{ m·s}^{-1}.$$

Pro skutečnou velikost celkové průměrné rychlosti při uražené vzdálenosti 0,8 m za dobu 4,6 s platí

$$v_p' = \frac{0{,}8}{4{,}5} = 0{,}178 \text{ m·s}^{-1}.$$

Rozdíl tedy činí

$$r = v_p - v_p' = -0{,}002 \text{ m·s}^{-1}.$$

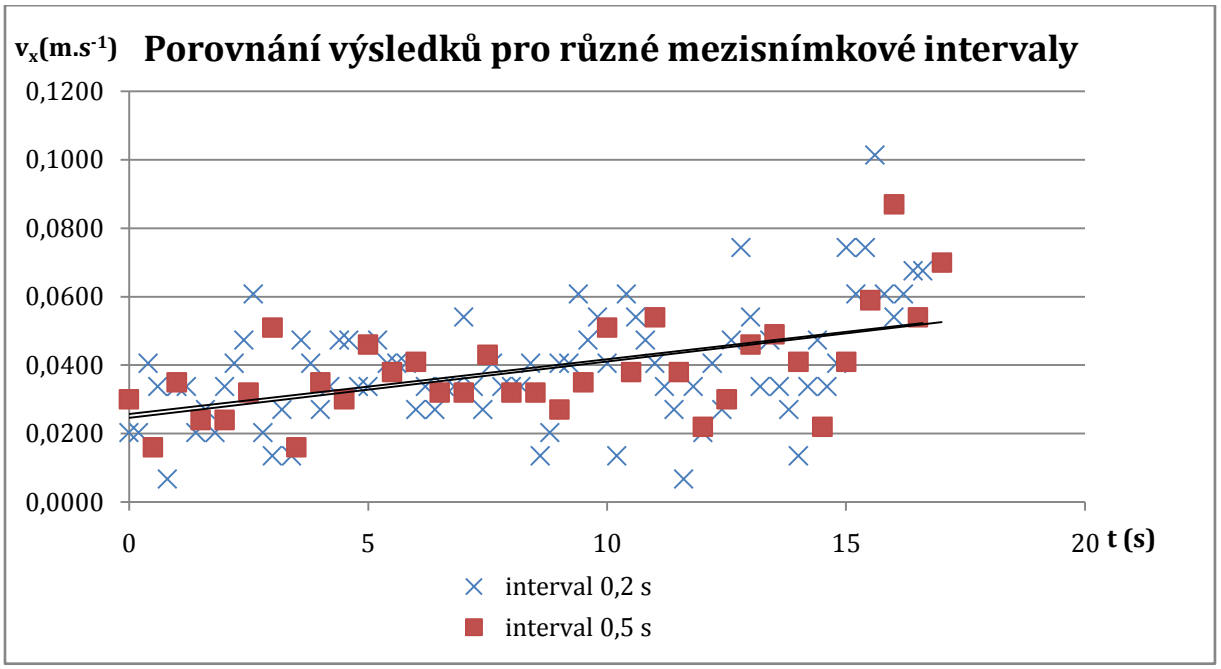


Obr. 8-8: Hodnoty rychlostí v_x a v_y pro video sekvenci č. 2

8.5.2 Porovnání výsledků pro různé mezi snímkové intervaly

V části 7.3 bylo řečeno, že výpočty rychlosti probíhají mezi 2 po sobě jdoucími snímky. Interval, s jakým jsou snímky z videa vyčítány, je volitelný. Vyčítání probíhá externě, zde pomocí freewarového programu. Volba hodnoty času, o který budou snímky od sebe vzdáleny, ovlivňuje především dobu výpočtu. Se snižováním intervalu dochází ke zvyšování počtu zpracovávaných snímků. Maximální hodnota počtu snímků je dána typem záznamu. Jedná se o počet snímků nasnímaných kamerou za jednu sekundu. Tento počet bývá běžně 25 snímků za 1 s, viz část 1.1. Tak je tomu i pro zde použitou kameru. Použití všech snímků pro výpočet je však neefektivní. Změna polohy objektu mezi dvěma snímky je jen velice malá. Počet snímků pro zpracování je třeba volit s přihlédnutím k přibližné rychlosti objektu. Pro rychle se pohybující objekt zvolit nižší interval, tedy větší počet snímků a pro pomalý pohyb naopak.

Graf na obr. 8-9 ukazuje porovnání výsledků video sekvence č. 1 pro hodnoty mezisnímkového intervalu 0,2 s a 0,5 s s lineární aproximací obou průběhů, které se téměř překrývají. Z čehož je patrné, že výsledky se pro oba intervaly v průměru liší jen málo. Dále je možné vidět nárůst počtu zpracovávaných snímků. Pro interval 0,5 s to je 35 snímků, pro 0,2 s již 84 snímků. Při zpracování všech snímků sekvence by byl počet snímků přes 400.



Obr. 8-9: Hodnoty rychlosti v_x pro různé mezisnímkové intervaly

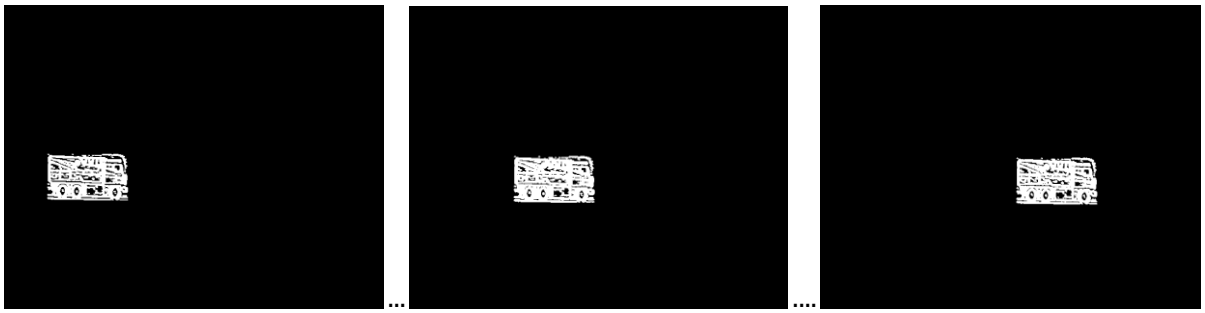
8.5.3 Porovnání hodnot pro různé metody vytvoření binárních masek

Ve všech předchozích výpočtech rychlostí bylo pro vytvoření binárních masek vstupujících do operátoru SpeedMeasuring využito prahovaných rozdílových snímků tvořených v operátoru AccumulativeDifferenceImage. Existuje mnoho dalších postupů, pomocí nichž je možné ve snímku nalézt objekt a vyznačit binární maskou jeho polohu. Jedním z nich je detektor hran vytvořený v rámci projektu ImageProccesingExtensions. Má za úkol vyhledat objekt dle předlohy v sérii obrázku. Jedním z jeho výstupů jsou masky s vyznačením polohy objektu v každém snímku. V rámci této práce se podoba objektu za dobu záznamu nemění, proto lze tento detektor hran s výhodou použít.

Pro testování fungování byla vybrána video sekvence č. 1. Vyhledávaná předloha byla vytvořena výřezem hledaného objektu (modelu automobilu) z jednoho ze série snímků. Její podoba je na obr. 8-10.

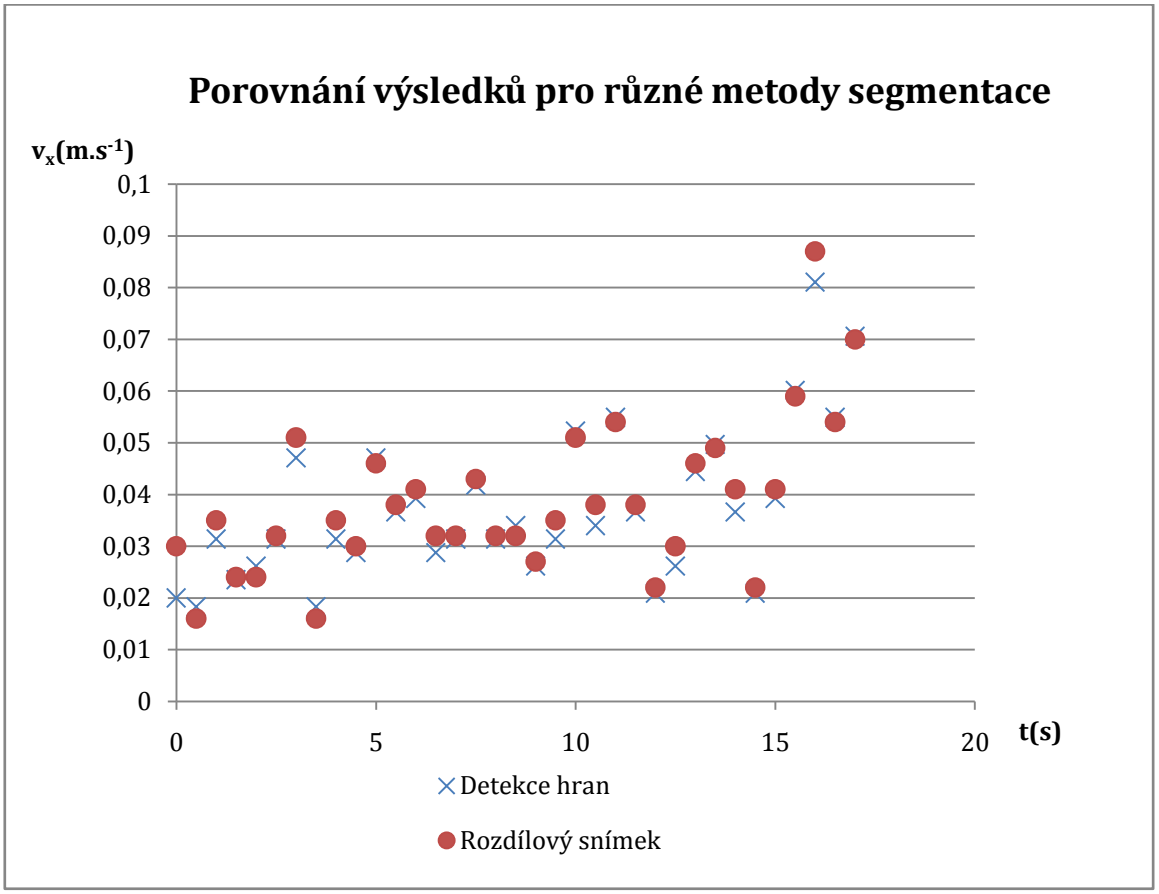
Obr. 8-10: Vzor pro algoritmus vyhledávání známého objektu

Po nastavení vhodného prahu detektoru hran vrátí tento algoritmus na výstup binární masky s vyznačením polohy, ve které našel nejlepší shodu s předlohou. Na obr. 8-11 jsou ukázky masek pro 3 snímky sekvence.



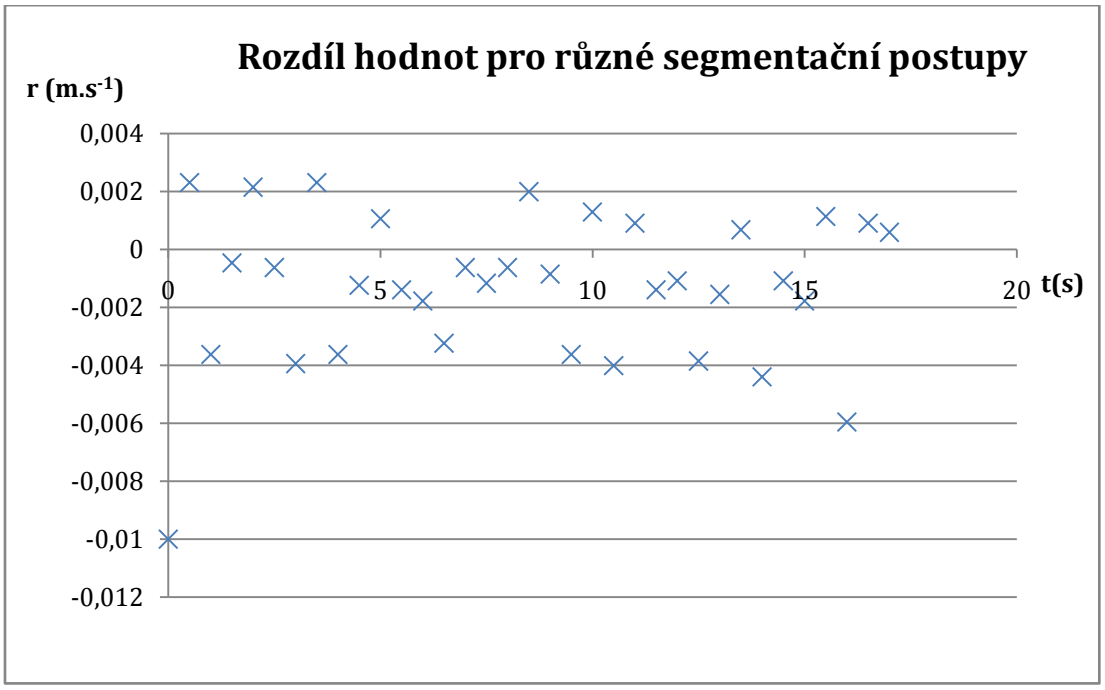
Obr. 8-11: Výstupní binární masky z algoritmu vyhledávání známého objektu

Tyto masky tedy mohou být vstupem pro operátor počítající velikost rychlosti objektu. Graf na obr. 8-12 ukazuje výsledky vypočtené pro masky vytvořené rozdílovými snímky jako v předchozích případech a metodou detekce hran popsanou v této kapitole.



Obr. 8-12: Výsledky pro rychlost v_x při použití různých segmentačních metod

Z výsledků je patrná malá odchylka hodnot pro obě metody, což dokládá i graf zobrazující rozdíly velikostí rychlostí v každém měřeném snímku na obr. 8-13.



Obr. 8-13: Rozdíl hodnot v_x pro různé segmentační postupy

9 Závěr

Tato diplomová práce se zabývá studiem záznamu pohybu objektu uvnitř statické scény. V rámci práce byly vytvořeny 3 operátory v prostředí RapidMiner popisující pohyb objektu ve video sekvenci. Studium pohybu se omezilo na případ jednoho pohybujícího se objektu v jinak statické scéně.

Pro záznam trajektorie pohybu byla využita technika kumulovaného rozdílového snímku. Testování operátoru prokázalo, že přehledných výsledků je dosaženo vhodnou volbou počtu analyzovaných snímků s co nejmenším obsahem obrazového šumu. Pro zjištění typu pohybu a určení velikosti změny polohy objektu byl implementován operátor využívající technik optického toku a vektorů pohybu. Zde testování na reálných video sekvencích nepřineslo uspokojivé výsledky především z důvodu nekonstantního osvětlení scény. Techniky kompenzace tohoto nežádoucího jevu nejsou triviální a přesáhly by obsahově tuto práci.

Pro výpočet velikosti rychlosti ve vertikálním a horizontálním směru byl vytvořen operátor detekující hranice objektu ve vstupních binárních maskách. Operátor na základě vypočtené změny polohy objektu v po sobě následujících snímcích a explicitně zadaných parametrech určujících podmínky vzniku záznamu určí rychlost v m/s v jednotlivých směrech. Z výsledků testování operátoru na různých video sekvencích vytvořených různými záznamovými zařízeními je patrná jen malá odchylka od skutečné velikosti rychlosti.

Použitá segmentační technika rozdílových snímků neposkytuje robustní nástroj pro zpracování širokého rozsahu reálných video sekvencí. V případě náhlých změn pozadí scény, např. z důvodu změn osvětlení, tato technika selhává. Pro složitější scény s případným pohybem jiného objektu než je měřený, je třeba použít jiný segmentační postup. V práci byl otestován způsob nalezení objektu pomocí detekce hran dle vzoru hledaného objektu. Výsledky se jen málo lišily od metody rozdílových snímků. Bylo tak potvrzeno, že způsob vytvoření binárních masek je pro operátor měřící rychlost transparentní, tudíž může být využit po jakékoliv segmentační technice a lze ho aplikovat i na video sekvence zachycující složitější scény než testované v práci.

Literatura

[1] ROZSÍVAL, P**.** *Oční lékařství.* Praha : Galén, 2006. ISBN: 80-7262-404-0.

[2] ŘÍČNÝ, V**.** *Videotechnika.* [skriptum] Brno : Vysoké učení technické v Brně, FEKT, UREL, 2006. ISBN 80-214-3225-X.

[3] ČÍKA, P**.** *Multimediální služby.* [skriptum] Brno : Vysoké učení technické v Brně, FEKT, UTKO, 2007.

[4] ŠPANĚL, M**.** *Rozpoznávání gest ve video sekvencích*. Brno : Vysoké učení technické v Brně, FIT, 2003. Vedoucí diplomové práce Doc. Dr. Ing. Pavel Zemčík

[5] LINKA, A.; VOLF, P.; KOŠEK, M. *Zpracování obrazu a jeho statistická analýza.* [skriptum] Liberec : Technická univerzita v Liberci, 2004.

[6] ŠEBESTA, V.; SMÉKAL, Z. *Signály a soustavy.* [skriptum] Brno : Vysoké učení technické v Brně, FET, UTKO, 2006.

[7] ŠPANĚL, M.; BERAN, V. *Obrazové segmentační techniky.* [skriptum] Brno : Vysoké učení technické v Brně, FIT, UPGM, 2005.

[8] ŠONKA, M.; HLAVÁČ, V.; BOYLE, R. *Image Processing, Analysis and Machine Vision.* Boston : PWS, 1999.

[9] JIŘINA, M.; ŠNOREK, M. *Neuronové sítě a neuropočítače.* [skriptum] Praha : České vysoké učení technické, 1996.

[10] HORÁK, K.; KALOVÁ, I.; PETYOVSKÝ, P.; RICHTER, M. *Počítačové vidění.* [skriptum] Brno : Vysoké učení technické, FEKT, 2008.

[11] LENG, B.; DAI, Q. *Video Object Segmentation based on accumulative frame difference.* Peking : Tsinghua University, 2007.

[12] RAPID-I**.** Approaching Vega: The final descent, How to extend RapidMiner 5.0. *rapid-i.com.* [Online] 12. únor 2010. [Citace: 15. listopad 2010.] http://rapid-i.com/content/view/26/84/.

[13] SYMES, P. *Digital Video Compression.* New York : McGraw-Hill, 2003. 0-07-142487-3.

[14] SURHONE, M., T.; TENNOE, T., M.; HENSSONOW, F.,S. *Block-Matching Algorithm.* místo neznámé : VDM Verlag Dr. Mueller AG Co. Kg, 2010. 6-13-460727-4.

[15] GHANBARI, M. *The cross-search algorithm for motion estimation.* Pisa : Universita Pisa s Anna, 1990. 0090-6778.

[16] MANNING, C. Video Compresion Explained. *Colin Manning.* [Online] 1996. [Citace: 10. 4 2011.] http://newmediarepublic.com/dvideo/compression/adv18.html.

[17] ŘÍHA, K. *Pokročilé zpracování obrazu.* [skriptum] Brno : Vysoké učení technické v Brně, FEKT, UTKO, 2007.

[18] SOUKUP, R. *Škola digitální fotografie.* Praha : Grada, 2006. 80-247-1077-3.

Seznam použitých zkratek

fps	frames per second
IP	Integral Projection
JPEG	Joint Photographic Experts Group
MAD	Mean Absolute Difference
MAE	Mean Absolute Error
MPEG	Motion Pictures Experts Group
MSD	Mean Square Difference
MSE	Mean Square Error
PDC	Pel Difference Classification

Seznam použitých veličin

a	velikost vyhledávacího okna
$a_0 - a_5$	parametry modelu optického toku
A, B	blok pixelů
d	vyhledávací vzdálenost
d	rozměr snímače
$d\,(x, y)$	obrazová funkce rozdílového snímku
$d_{\mathrm{kum}}\,(x, y)$	obrazová funkce kumulovaného rozdílového snímku
$d_{\mathrm{R}}\,(x, y)$	obrazová funkce rozdílového snímku pro červený kanál
$\mathrm{d}\boldsymbol{r}$	prostorový vektor pohybu
$\mathrm{d}\boldsymbol{r_i}$	obrazový vektor pohybu
$\mathrm{d}t$	časový úsek
$\mathrm{d}x$	změna polohy na ose x
f	ohnisková vzdálenost
$f\,(x, y)$	obrazová funkce
$f_1\,(x, y)$	obrazová funkce
$f_2\,(x, y)$	obrazová funkce
$g\,(x, y)$	obrazová funkce
$k_i(x, y)$	obrazová funkce i - tého snímku
$k_{i\mathrm{R}}(x, y)$	obrazová funkce i - tého snímku pro červený kanál
l	vzdálenost scény od objektivu
m	výška bloku
n	šířka bloku
n	počet snímků
n	počet snímků za 1 sekundu

N	počet kroků
$o(x, y)$	obrazová funkce výsledného snímku
$o_R(x, y)$	obrazová funkce výsledného snímku pro červený kanál
p	hodnota aktuálního řádku
$P, P´$	body v prostoru
$P_i, P_i´$	body v obrazu
q	hodnota aktuálního sloupce
r	rozdíl skutečné a naměřené průměrné rychlosti
$\boldsymbol{r}, \boldsymbol{r}´$	prostorový polohový vektor
$\boldsymbol{r_i}, \boldsymbol{r_i}´$	obrazový polohový vektor
$r\,(x, y)$	obrazová funkce referenčního snímku
$r_R\,(x, y)$	obrazová funkce referenčního snímku pro červený kanál
s	velikost vyhledávacího kroku
$t\,(p, q)$	shodnost pixelů na pozici (p, q)
T	hodnota prahu
$u\,(x, y)$	horizontální složka optického toku
v	celková rychlost
v_i	celková rychlost v obraze
v_{px}	naměřená průměrná rychlost ve směru osy x
$v´_{px}$	skutečná průměrná rychlost ve směru osy x
v_x	rychlost ve směru osy x
v_{xi}	i-tá naměřená rychlost
v_y	rychlost ve směru osy y
v_{0x}	rychlost v obraze ve směru osy x
$v\,(x, y)$	vertikální složka optického toku
w_i	váha
$x, x_1, x_2, x´$	hodnota souřadnice osy x
$x_i, x_i´$	hodnota souřadnice osy x v obraze
x_{obr}	šířka snímku
X_{real}	šířka scény
$y, y´$	hodnota souřadnice osy y
$y_i, y_i´$	hodnota souřadnice osy y v obraze
$z, z´$	hodnota souřadnice osy z
α	zorný úhel
α_x	zorný úhel v horizontální rovině
ε	hodnota prahu

Seznam příloh

A. Popis obsahu přiloženého CD

Na přiloženém CD se nachází celý projekt ImageProccesingExstension, na kterém pracoval větší počet lidí. Třídy zpracované v rámci této diplomové práce mají označení AccumulativeDifferenceImage, OpticalFlow a SpeedMeasuring. Ve složce „Operatos“ se nachází dvě podsložky obsahující operátorová zapojení určená k použití navrhnutých tříd. V podsložce „Accum+Speed“ se nachází zapojení operátorů AccumulativeDifferenceImage a SpeedMeasuring. Podsložka OpticalFlow obsahuje zapojení operátoru OpticalFlow. CD také obsahuje ve složce „Videosekvence“ video sekvence vytvořené pro testování funkčnosti operátoru SpeedMeasuring. Označení video sekvencí odpovídá popisu v textu.

Dále je na CD projekt RapidMiner_Vega, u kterého je přiložen návod na instalaci, příslušný dokument nese označení „jaknarm.doc“. Další přiloženou součástí je instalační aplikace programu Free Video to JPEG Converter, který byl použit k ukládání snímků z video sekvencí a elektronická verze textu práce.