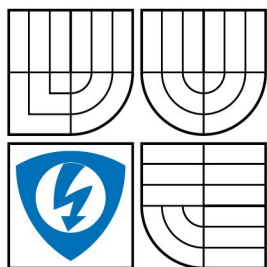


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SLEDOVÁNÍ TRAJEKTORIE

PATH MONITORING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

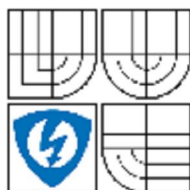
AUTOR PRÁCE
AUTHOR

JAN SEDLAŘÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MILOSLAV RICHTER, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Jan Sedlařík
Ročník: 3

ID: 72898
Akademický rok: 2008/2009

NÁZEV TÉMATU:

Sledování trajektorie

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a realizovat algoritmy, které umožňují zjistit polohu objektu a sledovat jeho pohyb. Nasnímejte databázi videí pro různé typy pohybů. Proveďte rozbor vyhodnocení těchto typů pro různé metody a zhodnoťte jejich kvalitu. Zhodnoťte možnost kombinací jednotlivých metod.

DOPORUČENÁ LITERATURA:

Kraus K.: Photogrammetrie 1 und 2, Ummier / Bonn, 1996
Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 1998, ISBN 80-251-0454-0
Hlaváč V., Sonka M.: Počítačové vidění, Grada, Praha 1992, ISBN 80-85424-67-3
Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

Termín zadání: 9.2.2009

Termín odevzdání: 1.6.2009

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být 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.

A b s t r a k t

Tato bakalářská práce se zabývá tématem sledování trajektorie. Jednou z nejzákladnějších úloh tohoto problému je detekce objektů v obraze a jejich následné sledování. Důležitou úlohu při separaci obrazu na objekty sehrává detekce hran.

V první části jsou teoreticky popsány nejdůležitější metody, které jsou v průběhu práce využívány. Jedná se o popis předzpracování videa, stručné seznámení se zobrazením a definicí pohybu, popis detekce hran a metod určení optického toku. Druhá část popisuje postupy a algoritmy použité v navrhnutém programu.

K l í ě o v á s l o v a

detekce objektů, prokládání, detekce hran, optický tok, diferenciální metody

A b s t r a c t

This bachelor's thesis deals with path monitoring theme. Detection of objects and their monitoring are one of the most elementary parts of this challenge. Important part of image separation is edge detection.

In the first part of this thesis there are described some important methods which are used in this work. It is description of video preprocessing, brief projection characterization and motion definition and description of edge detection and optical flow calculating methods. Procedures and algorithms used in designed application are depicted in the second part.

K e y w o r d s

object detection, interlacing, edge detection, optical flow, differential methods

Bibliografická citace

SEDLAŘÍK, Jan. *Sledování trajektorie* Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 56 stran, vedoucí Ing. Miloslav Richter, Ph.D.

P r o h l á š e n í

„Prohlašuji, že svou bakalářskou práci na téma Sledování trajektorie jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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:

P o d ě k o v á n í

Děkuji tímto Ing. Miloslavu Richterovi, Ph.D. za cenné připomínky a rady při vypracování bakalářské práce.

V Brně dne :

Podpis:

OBSAH

1. ÚVOD	9
2. PŘEDZPRACOVÁNÍ VIDEO.....	10
2.1 Převod reálné scény do digitální.....	10
2.2 Prokládání.....	11
3. ZOBRAZENÍ A POHYB	14
3.1 Paralelní a perspektivní projekce	14
3.2 Mechanický pohyb.....	16
4. DETEKCE HRAN.....	18
4.1 Cannyho hranový detektor	20
5. OPTICKÝ TOK	22
5.1 Obecný popis	22
5.2 Metodika určování optického toku	23
5.3 Metody určení optického toku.....	24
5.4 Diferenciální metody.....	25
5.5 Blokové metody.....	29
5.6 Metody založené na energii.....	30
6. VLASTNÍ ZPRACOVÁNÍ.....	31
6.1 Předzpracování videa	31
6.2 Vlastní zpracování	32
7. ZÁVĚR	45
8. LITERATURA	46
SEZNAM PŘÍLOH.....	48
PŘÍLOHA A: PŘÍKLADY VÝSLEDKŮ DETEKCE OBJEKTŮ	49
PŘÍLOHA B: DATABÁZE VIDEÍ	52
PŘÍLOHA C: POPIS PROGRAMU	54
PŘÍLOHA D: OBSAH PŘÍLOŽENÉHO DVD	56

SEZNAM OBRÁZKŮ

OBR 2.1 ČTVERCOVÁ VZORKOVACÍ MŘÍŽKA	10
OBR 2.2 PROKLÁDÁNÍ. VLEVO PROKLÁDANÝ SNÍMEK, VPRAVO SNÍMEK PO ÚPRAVĚ	12
OBR 3.1 PARALELNÍ A PERSPEKTIVNÍ PROMÍTÁNÍ.	14
OBR 3.2 PŘÍKLAD PERSPEKTIVY.....	15
OBR 4.1 VLEVO IDEÁLNÍ SKOKOVÁ HRANA, VPRAVO HRANA ZATÍŽENÁ ŠUMEM.	18
OBR 4.2 VLEVO PRVNÍ, VPRAVO DRUHÁ DERIVACE IDEÁLNÍ HRANY.	19
OBR 5.1 ZNÁZORNĚNÍ APERTUROVÉHO PROBLÉMU	24
OBR 6.1 SNÍMEK VIDEOSEKVENCE.	32
OBR 6.2 OBRÁZEK VE STUPNÍCH ŠEDI (GRAYSCALE).	33
OBR 6.3 VÝSLEDEK DETEKCE HRAN S VYUŽITÍM SOBELOVA DETEKTORU S VYUŽITÍM KONVOLUČNÍHO JÁDRA 5X5.	34
OBR 6.4 VÝSLEDEK DETEKCE HRAN S VYUŽITÍM SOBELOVA DETEKTORU S VYUŽITÍM KONVOLUČNÍHO JÁDRA 7X7.	35
OBR 6.5 VÝSLEDEK CANNYHO HRANOVÉHO DEKEKTORU S VHODNĚ NASTAVENÝMI PRAHY.....	36
OBR 6.6 VÝSLEDEK CANNYHO HRANOVÉHO DETEKTORU S NEVHODNĚ ZVOLENÝMI PRAHY.....	37
OBR 6.7 VÝZNAMNÉ BODY NALEZENÉ VE SNÍMKU.	39
OBR 6.8 VÝZNAMNÉ BODY NALEZENÉ VE DVOU PO SOBĚ NÁSLEDUJÍCÍCH SNÍMCÍCH.	40
OBR 6.9 CHYBA URČENÍ VEKTORU ZE DVOU BLÍZKÝCH BODŮ.....	41
OBR 6.10 HISTOGRAM POSUNUTÍ NALEZENÝCH VÝZNAMNÝCH BODŮ MEZI SOUSEDNÍMI SNÍMKY.	42
OBR 6.11 VÝSLEDNÁ DETEKCE OBJEKTŮ.....	43
OBR 6.12 PŘÍKLAD ZOBRAZENÍ TRAJEKTORIE.	44

1. ÚVOD

Převod lidem vrozené schopnosti vidění do oblasti počítačů je v dnešní době často řešeným problémem. Počítačové vidění můžeme chápat jako snahu porozumět obecné trojrozměrné scéně. Postupy a nástroje převodu reálné scény do světa počítačů jsou značně složité. Musíme si uvědomit, že lidské vnímání okolí zrakem, tedy vidění, probíhá většinou na základě předchozí zkušenosti. Jádrem pokročilejších systémů využívajících počítačové vidění tedy může být tzv. umělá inteligence.

Jedněmi z často kladených požadavků na analýzu digitálně snímané scény jsou detekce a vyhodnocení pohybu snímaných předmětů nebo automatizovaná orientace v prostoru. Nutnou součástí algoritmů plnících tyto požadavky je v první řadě snímání a digitalizace obrazu. V další řadě předzpracování pořízeného videa a konečně samotné algoritmy segmentace zaznamenaného obrazu na jednotlivé objekty a jejich popis. Jedním ze základních prvků této úlohy je vyhodnocení tzv. optického toku. Optický tok bude jedním ze stěžejních témat v této práci a bude využíván k detekci objektů v obraze a tím umožní sledování jejich trajektorie.

V této práci budou uvažovány scény s pohyblivým pozorovatelem (kamerou) a stacionárními tedy nepohybujícími se objekty. Snahou bude detekovat objekty v obraze a umožnit tak sledování jejich trajektorie. Bude nasnímána databáze videí pro různé typy pohybu kamery a rozmístění objektů.

Součástí práce bude vytvoření programu, který umožní demonstrovat funkčnost navržených algoritmů a z předložených videí bude schopen detekovat jednotlivé objekty a sledovat jejich trajektorii.

2. PŘEDZPRACOVÁNÍ VIDEOA

V této kapitole bych se rád věnoval teorii postupů a operací, které byly prováděny v rámci této práce. Jedná se o teoretický popis principů zaznamenávání videa a jeho úprav v té nejzákladnější podobě potřebné pro úspěšné splnění dalších úkolů. Nejedná se tedy o definici předzpracování obrazu, které je samo o sobě mnohem komplexnější a složitější a zahrnuje spíše než principy zaznamenávání postupy úprav již pořízené sekvence snímků – videa. Některé metody obecného předzpracování obrazu jsou použity a budou popsány dále.

2.1 PŘEVOD REÁLNÉ SCÉNY DO DIGITÁLNÍ

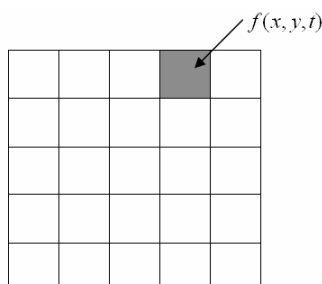
Obraz lze reprezentovat jako spojité rozložení jasu v rovině, které je definováno jako

$$f(x, y, t) \quad (2.1)$$

kde x, y ... souřadnice daného bodu v rovině

t ... čas

Pro další zpracování je nutné obraz diskretizovat, nebo-li převést do digitální podoby. To si můžeme představit jako diskrétní funkci stejného tvaru jako je rovnice (2.1). Tyto funkce obrazových elementů řadíme do vzorkovacích mřížek, čili jakýchsi matic výsledného digitalizovaného obrazu. Nejčastěji používaná je čtvercová mřížka.



Obr 2.1 Čtvercová vzorkovací mřížka

Takováto obrazová funkce je však schopná nést v jednom elementu vzorkovací mřížky (pixelu) hodnotu jasu pouze jedné barvy, vznikne nám tudíž šedo-tónový (monochromatický) obraz. Pro barevný obraz musíme každému obrazovému bodu (pixel) přiřadit funkce intenzity více barev. Pro daný obraz tedy musíme vytvořit další bitové roviny, které ponesou informaci o intenzitách různých barevných složek.

Bitová rovina je součástí framebufferu popisujícího celý obraz. Každá taková rovina, nebo jejich uskupení, nese jednotlivé informace o daném pixelu (barva, hloubka, průhlednost aj.)

Jasovou funkci tedy musíme popsat jako složení jasových funkcí jednotlivých barevných složek,

$$f[f_R(x, y, t), f_G(x, y, t), f_B(x, y, t)] \quad (2.2)$$

kde každá funkce reprezentuje jas jedné barevné složky podle zvoleného barevného modelu (RGB, CMY, YUV ...). [1] [2]

Věnujme se nyní ještě argumentu t jasové funkce, čili času. Video se skládá ze sekvence po sobě pořízených snímků zaznamenávajících danou scénu, parametr t udává čas, v jakém jsou jednotlivé snímky pořizovány. Počet snímků, jež jsou pořízeny za jednotku času, se nazývá snímková frekvence (*framerate*). V současné době jsou stále ještě celosvětově nejpoužívanější analogové normy PAL a NTSC, přičemž norma PAL má definovaných 50 fps (frames per second – snímků za sekundu) a norma NTSC 60 fps.

2.2 PROKLÁDÁNÍ

Prokládání neboli Interlace je způsob, jakým běžné videokamery zaznamenávají obraz. Sejmутý obraz se neskládá, jak bychom očekávali, z celých snímků. K pochopení dané problematiky se musíme vrátit zpět do 30. let minulého století do doby, kdy začala vznikat televizní technika a s ní spojená potřeba zaznamenávat obrazové scény.

Analogový televizní signál přenáší 50 půlsnímků za sekundu. Jeden půlsnímek je tvořen polovinou řádků celého obrazu, ve skutečnosti je tedy celých obrazů pouze 25 za sekundu. První půlsnímek obsahuje sudé řádky, druhý liché. Na analogové TV obrazovce se nejprve zobrazí první půlsnímek obsahující sudé řádky a poté druhý s lichými řádky, přičemž obrazovka již nezobrazuje snímek předchozí. Oba snímky musí být pořízeny v různém čase. Tímto postupem získáváme iluzi 50 snímků za sekundu.

Počítačové monitory a moderní LCD a plasmové televize ovšem zobrazují neprokládaně, po celých snímcích. Půlsnímký je tedy nutné poskládat do celého obrazu. Protože ale půlsnímký nejsou u prokládaného videa zachyceny ve stejných časových okamžicích, liší se v nich poloha pohybujících se objektů. Ve výsledném obraze tudíž vznikají zuby – roztřepení, rozmazání, jak je patrné na následujícím obrázku. Tímto postupem také ztrácíme iluzi 50 snímků za sekundu a vzniká nám sekvence s frekvencí poloviční, tedy 25 fps.



Obr 2.2 Prokládání. Vlevo prokládaný snímek, vpravo snímek po úpravě [4].

Protože k natočení videosekvencí zpracovávaných v této práci byla použita videokamera zaznamenávající video prokládaně, musíme se touto problematikou zabývat. Pro potřeby podrobné analýzy videa, která je součástí této práce, je nutné tento nežádoucí projev odstranit, protože v obraze nám tímto vznikají nežádoucí

hrany a rohy vzniklé řádkovým posunem struktury pohybujících se objektů. V případě pohybující se kamery, kdy se důsledkem toho pohybuje celá scéna, dochází k projevům interlacingu na celém obraze.

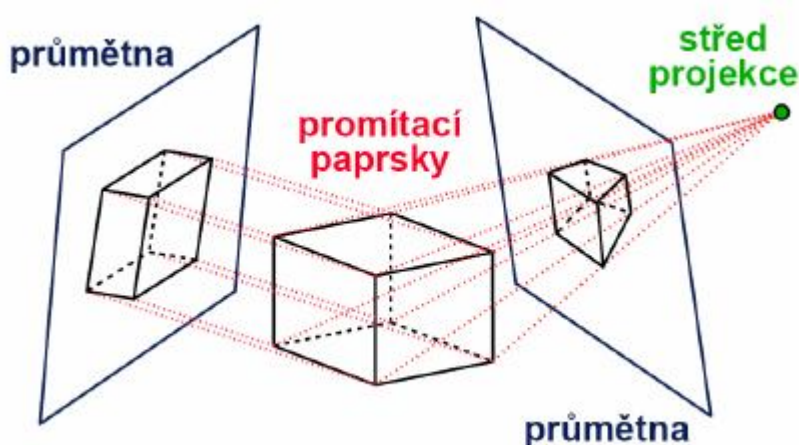
Existuje několik metod odstranění prokládání. Jedna z nejjednodušších a obecně nejrozšířenějších metod – BoB 50 – funguje tak, že v jednotlivých půlsnících duplikuje chybějící řádky. Dochází tak k výraznému snížení vertikálního rozlišení (o 50%), ovšem protože použijeme každý snímek zvlášť, zachovává se vysoký framerate výsledného videa, tedy původní hodnota, s jakou byly pořizovány jednotlivé půlsnímký. Další algoritmy a filtry, kterých je velké množství jsou popsány například v textu [4]. Tyto pokročilejší filtry jsou podstatně komplikovanější, nabízejí však kvalitnější odstranění následků prokládání. Ve filmu se snaží najít poškozená místa a ta opravit. Za všechny jmenujme například algoritmy Smart BOB, TomsMoComp nebo filtr programu VirtualDub Deinterlace MAP, který byl použit pro úpravu videí dále zpracovávaných v této práci.

3. ZOBRAZENÍ A POHYB

V této kapitole se stručně seznámíme s teorií zobrazení a pohybu. Budou popsány metody zobrazení a teorie pohybu.

3.1 PARALELNÍ A PERSPEKTIVNÍ PROJEKCE

Okolní svět je samozřejmě popsán trojrozměrně. Ovšem převod reálné scény do digitální pomocí videokamer a dalších snímacích zařízení je zároveň převodem této scény do dvourozměrného prostoru. Projekce nebo také promítání je transformace ze 3D do 2D prostoru. Během tohoto převodu dochází k nevratné ztrátě některých informací (například přesná vzdálenost objektu od pozorovatele). Rozlišujeme dva základní druhy projekcí, paralelní a perspektivní.



Obr 3.1 Paralelní a perspektivní promítání [3].

Paralelní projekce je lineární a je realizována zobrazením pomocí rovnoběžných paprsků. Jejimi největšími výhodami je, že zachovává rovnoběžnost hran a že vzdálenost zobrazovaného objektu od průmětny neovlivňuje velikost průmětu. Tento způsob promítání však při záznamu reálné scény takřka neexistuje.

Při záznamu okolního světa se však úspěšně uplatňuje tzv. perspektivní projekce. Paprsky zaznamenávající danou scénu vycházejí z jediného bodu (středu

projekce). V případě lidského vnímání je tento bod reprezentován okem. Při snímání kamerou objektivem. Tento způsob zaznamenávání způsobuje efekt perspektivy.

3.1.1 Perspektiva

Perspektiva je optický jev, který způsobuje některé optické úkazy, které budeme v naší práci využívat. Pokud budeme uvažovat konstantní stejnou rychlost dvou objektů, pak objekt, který bude blíže k pozorovateli (kameře) se bude v obraze pohybovat rychleji než ten, který je od pozorovatele dále. Další velice důležitá vlastnost tohoto optického jevu je ta, že některé ve skutečnosti rovnoběžné přímky se jeví jako přímky, které se sbíhají v jednom bodě na horizontu, jak ukazuje následující obrázek.



Obr 3.2 Příklad perspektivy [5].

Dalším typickým příkladem je pohled na rovné vlakové kolejiště, které vytváří iluzi, že koleje se k sobě postupně přibližují a někde v dálce se dokonce protínají. Tento bod konvergence, kde by došlo ke zdánlivému protnutí reálných rovnoběžek, nazýváme úběžníkem (*vanishing point*).

3.2 MECHANICKÝ POHYB

Ve fyzice se pod tímto termínem označuje každý pohyb, při kterém dochází ke změně vzdálenosti dvou hmotných bodů za určitý časový interval nebo také ke změně velikosti či tvaru tělesa. V našem případě však bude postačovat chápat pohyb tělesa jako časovou změnu jeho vzdálenosti od tělesa jiného v dané vztažné soustavě. Je důležité si uvědomit, že jako těleso můžeme chápat i bod pozorování, tedy pozorovatele, v našem případě kameru.

Vztažnou soustavou rozumíme uskupení bodů v prostoru, které jsou vzájemně v klidu, anebo ve vzájemně zadaném či známém pohybu. Vzhledem ke zvolené vztažné soustavě je pohyb fyzikálně vztahován.

3.2.1 Dělení mechanických pohybů

Mechanické pohyby můžeme dělit podle několika kritérií:

- podle tvaru trajektorie
 - přímočaré - kdy je trajektorií pohybu přímka
 - obecné – trajektorií takového pohybu je křivka a to jak otevřená (parabola, obecná křivka) tak i uzavřená (kružnice, elipsa).
- podle dimenze prostoru, ve kterém se těleso pohybuje
 - lineární (jednorozměrný)
 - rovinný (dvourozměrný)
 - prostorový (třírozměrný)
- podle změny rychlosti v čase.
 - rovnoměrný - rychlost tělesa se v čase nemění
 - nerovnoměrný
 - zrychlený – zrychlení daného tělesa je větší než 0
 - zpomalený – zrychlení daného tělesa je menší než 0

Existují i další možná dělení. Pro požadavky této práce je výčet výše zmíněných dostačující.

3.2.2 Dělení mechanických pohybů

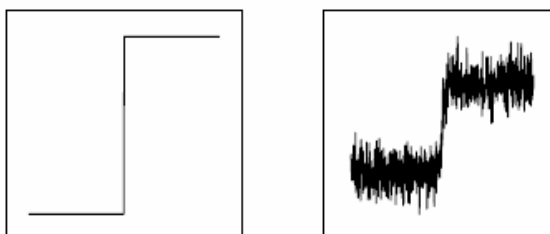
Mohou nastat čtyři následující situace vzájemného pohybu objektů snímaných ve scéně a pozorovatelem (kamerou).

- kamera v klidu, objekty v klidu
- kamera v klidu, objekty v pohybu
- kamera v pohybu, objekty v klidu
- kamera v pohybu, objekty v pohybu

O pohybu v obraze můžeme logicky mluvit pouze ve třech posledně jmenovaných případech. V naší práci se ovšem budeme zabývat pouze situacemi, kdy se bude pohybovat kamera. Pro zjednodušení nejprve v kombinaci se stacionárními objekty.

4. DETEKCE HRAN

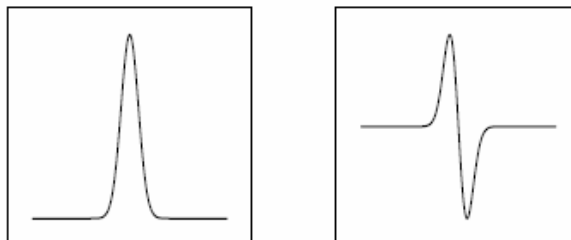
Jedna z nejzákladnějších úloh při segmentaci obrazu na jednotlivé objekty je detekce hran. V případě přítomnosti objektu v obraze se předpokládá, že je jasně odlišitelný od svého okolí a splňuje určitá kritéria, jako je například jasná ohraničenost. Hranou v obraze rozumíme skokovou změnu intenzity jasu. Tato změna je nejčastěji zastoupena jasovou nespojitostí, která definuje hranu, případně lokálním maximem definujícím linii. Linie je ovšem případ specifický a pro zjednodušení je možné ho chápat jako dvě těsně vedle sebe následující hrany. Je nutné uvažovat také uskupení, kde se tyto hrany a linie stýkají. Taková místa definují vrcholy, případně křížení. V obrazových scénách se nejčastěji vyskytují hrany. Ideální hrana, která bude zastoupena výše zmíněnou skokovou změnou intenzity, se však ve skutečných úlohách prakticky nevyskytuje. Hrany jsou vždy zatíženy určitými nepřesnostmi, jako je šum nebo rozostření. Tyto chyby způsobují, že průběh funkce jasové intenzity není ideální skok, jak ukazuje následující obrázek.



Obr 4.1 Vlevo ideální skoková hrana, vpravo hrana zatížená šumem.

K základům algoritmů detekujících hrany slouží derivace. V případě první derivace je v místech, kde je jas konstantní jasová derivace rovna nule, v místech lineárních přechodů jedné hodnoty na druhou je lineárně rostoucí či klesající. První derivace tedy pouze identifikuje přítomnost změny jasu.

Druhá derivace nám říká, zda-li došlo k přechodu z tmavší oblasti do světlejší či naopak. To o který případ se jedná určuje její znaménko. Na následující obrázku je znázorněna první a druhá derivace ideální (nezašuměné) skokové hrany.



Obr 4.2 Vlevo první, vpravo druhá derivace ideální hrany.

Maximální hodnota derivace bude vždy ve směru kolmém na hranu. Budeme-li tedy vyšetřovat obraz ve směru osy x , v posloupnosti vyšetřovaných pixelů nastane skoková změna intenzity jasu pouze v případě, že hrana bude kolmá (lépe řečeno více kolmá než rovnoběžná) na směr osy x . Hrany, které mají shodný nebo podobný směr jako tato osa, tedy nebudou zaznamenány. Z toho důvodu je hrany nutné detekovat ve více směrech. Nejčastěji se používá dvousměrný přístup, tedy detekce ve směru osy x a osy y . V praxi se někdy používají i přístupy využívající derivace z více směrů. Více o teorii detekce hran lze nalézt například v [9].

Většina metod provádí výpočty s využitím aproximací derivací pomocí konvolučních jader. Nejpoužívanější konvoluční jádra jsou 0:

Robertsův operátor:

Jeden z prvních operátorů. Jeho největší nevýhodou je velikost jádra. Kvůli ní se nehodí na zašuměné nebo jinak porušené scény. V současnosti se téměř nepoužívá.

$$G_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad (4.1)$$

Sobelův operátor

Sobelův operátor je v současnosti pravděpodobně nejpoužívanějším konvolučním jádrem ke zjištění derivací intenzity jasu v daném bodě. Díky větším maticím 3x3 je robustnější vůči šumu.

$$G_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad (4.2)$$

Sobelův operátor je často využíván i pro výpočet derivací ve více směrech, v těchto případech jsou konvoluční matice „pootáčeny“ nebo-li rotovány do požadovaných směrů. O přesné hodnotě a tedy i směru hrany v daném bodě rozhoduje maska s nejvyšší hodnotou derivace.

Další obdobné konvoluční masky jsou například Kirschova či Prewittové. Tyto masky mívají také rozměr 3x3 ovšem pracují s jinými vahami jednotlivých okolních pixelů.

4.1 CANNYHO HRANOVÝ DETEKTOR

Tento detektor je v současné době nejrozšířenější a nejpoužívanější. Je popsán v 0. Vznikl počátkem 80. let. Tento algoritmus se skládá z několika po sobě následujících kroků:

- 1) eliminace šumu
- 2) určení gradientu
- 3) nalezení lokálních maxim
- 4) eliminace nevýznamných hran

ad 1) Filtrace šumu se provádí použitím Gaussova filtru. Filtr se nejčastěji převádí na konvoluční masku, která je aplikována na vyšetřovaný obraz. Příklad takové masky o rozměrech 5x5 se standardní odchylkou rozdělení $\sigma = 1.4$ je následující:

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (4.3)$$

ad 2) Nejčastěji je používán Sobelův hranový operátor zjišťující hrany ve dvou směrech (ve směru osy x a osy y). Celková velikost derivace, tedy váha hrany ve zkoumaném místě je počítána podle vztahu:

$$G = \sqrt{G_x^2 + G_y^2} \quad (4.4)$$

a směr poté hrany určíme následujícím výpočtem. Tento směr bývá nejčastěji převáděn na směr vertikální, horizontální a dva diagonální, tedy 0° , 45° , 90° a 135° .

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (4.5)$$

ad 3) Nalezení lokálních maxim (tzv. Thinning) spočívá v nalezení bodů, které jsou lokálními maximy. To znamená, že hledáme body, které vlevo a vpravo od směru gradientu, který jsme určili v předchozím bodu mají nižší hodnotu derivace než tento bod. Tímto postupem získáme pomyslný binární obraz, ve kterém jsou pixely charakterizovány informací „je hrana“ a „není hrana“. V případě, že pixel je označen jako hrana je však stále charakterizován velikostí gradientu, tedy velikostí derivace jasové funkce v daném bodě.

ad 4) Eliminace hran se provádí prahováním s hysterezí. Tedy prahováním se dvěma prahy. Postupně procházíme všechny body označené jako „je hrana“. Pokud je gradient vyšší než větší z obou prahů T_2 , je bod automaticky označen za finální hranu. Pokud je bod menší než nižší práh T_1 , je vyřazen, tedy označen jako „není hrana“. V případě, že gradient je mezi oběma hodnotami, rozhoduje o jeho příslušnosti k finální hraně jeho okolí, tzn. pokud některý z jeho sousedních bodů je uznán jako hrana, je i tento bod jako hrana označen.

5. OPTICKÝ TOK

5.1 OBECNÝ POPIS

Jak již bylo naznačeno v úvodu, optický tok je jednou ze základních informací, které můžeme získat z digitálně zaznamenané reálné scény – videa, sekvence digitálních snímků. Pomocí optického toku je možné automaticky sledovat pohyb ve snímané scéně. V případě pohybující se kamery lze tímto postupem získat informaci o libovolném bodě v obraze, ovšem díky maximálně dynamickému obrazu s většími problémy a ne vždy jednoznačnou informační hodnotou.

Optický tok je strukturovaný vizuální pohyb okolí, který je patrný pouze pro pohybujícího se pozorovatele. Jak jsme již naznačili, zaznamenaný pohyb můžeme chápat jako časovou změnu vzdálenosti mezi dvěma objekty nebo mezi pozorovatelem a objektem. Tato změna probíhá v prostoru. Digitálním zaznamenáním však převádíme trojrozměrnou scénu do dvojrozměrné. Tímto ztrácíme některé důležité informace o stavech snímaného prostředí. S tím souvisí tzv. aperturový problém, který bude vysvětlen později.

Jedná se tedy o vektorové pole, které nese informaci o zdánlivém pohybu objektů, ploch a hran v optické scéně. Alternativně můžeme detekovat a identifikovat jednotlivé objekty a pomocí optického toku o nich získat nejružnější informace. Tedy například rychlost, zrychlení, při znalosti rozměrů vzdálenost anebo dráhu, u níž může být vypovídající nejen její délka, ale i zakřivení. Další informací, kterou lze z optického toku získat je v případě pohybu kamery směrem ke snímaným objektům místo, do kterého směřuje pohyb kamery nebo při pohybu kamery směrem od scény bod, ze kterého pohyb kamery odstartoval. V případě, že je osa kamery rovnoběžná s osou pohybu, je toto místo nebo bod ve scéně shodné s úběžníkem. Pojem úběžníku jsme definovali v předcházejících kapitolách.

5.2 METODIKA URČOVÁNÍ OPTICKÉHO TOKU

Optický tok je určován ze změny intenzity v obraze. Každý bod v obraze, který je jedním snímkem z celé videosekvence, je dán funkcí jeho intenzity v čase, která se skládá z prostorové složky (souřadnic) a časové dimenze kdy byl snímek pořízen $E(x, y, t)$. Uvažujme jeden z takových bodů. Tento bod se za čas Δt , tedy mezi dvěma snímky, posune o Δx ve směru osy x a o Δy ve směru osy y . Protože je jeho intenzita shodná s intenzitou v předešlém časovém okamžiku, můžeme psát

$$E(x, y, t) = E(x + \Delta x, y + \Delta y, t + \Delta t) \quad (5.1)$$

rozšířením pravé strany rovnice dostaneme

$$E(x, y, t) = E(x, y, t) + \Delta x \frac{\partial E}{\partial x} + \Delta y \frac{\partial E}{\partial y} + \Delta t \frac{\partial E}{\partial t} + \varepsilon \quad (5.2)$$

kde ε jsou členy druhého a vyššího řádu Taylorova rozvoje ve směrech x , y a t . Po další úpravě můžeme psát

$$\frac{\partial E}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial E}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial E}{\partial t} + \sigma(\Delta t) = 0 \quad (5.3)$$

kde $\sigma(\Delta t)$ je opět výraz zahrnující členy derivací vyšších řádů. Za předpokladu, že Δx a Δy se mění stejně jako Δt , v limitě pro $\Delta t \rightarrow 0$ je tedy $\sigma(\Delta t) \rightarrow 0$ a platí vztah

$$\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \quad (5.4)$$

Jestliže budeme předpokládat, že pro konstantní jas v obrazovém vzoru platí

$$\frac{dE}{dt} = 0 \quad (5.5)$$

a zavedeme substituce

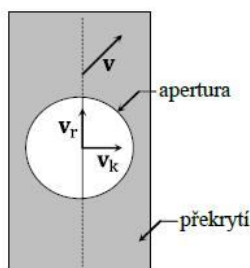
$$\frac{dx}{dt} = u \quad \text{a} \quad \frac{dy}{dt} = v$$

dostaneme výslednou diferenciální rovnici

$$E_x u + E_y v + E_t = 0 \quad (5.6)$$

kde E_x , E_y a E_t jsou parciální derivace funkce intenzity ve směrech x , y a t .

Pokud však nejsme schopni o této změně intenzity rozhodnout, nelze optický tok určit. Toto omezení je známo jako problém apertury. Znázornění je vidět na následujícím obrázku.



Obr 5.1 Znázornění aperturového problému [6]

Aperturou můžeme rozumět „výřez zobrazované scény“. Tedy pohled přes aperturu lze chápat například jako pohled skrze objektiv kamery. Scéna, která je patrná uvnitř apertury je alternativou scény snímané kamerou. Na obrázku Obr 5.1 **Znázornění aperturového problému [6]** je vidět přímka. Při pohybu této přímky ve směru vektoru $\vec{v}$ není možné přesně stanovit její pohyby v tečném $\vec{v}_r$ a normálovém $\vec{v}_k$ směru. Jsme schopni určit pouze rychlostní složku kolmou na danou přímku. Obdobný problém by nastal pokud by celá snímaná scéna byla reprezentována jednoduchou jednobarevnou plochou, například listem papíru. Pokud bychom tímto listem pohybovali před objektivem kamery, v zaznamenané scéně bychom tento pohyb nebyli schopni rozpoznat.

5.3 METODY URČENÍ OPTICKÉHO TOKU

Jak vyplývá z rovnice (5.6), cílem úkolu je spočítat složky rychlosti u a v . Řešení nalezení těchto vektorů spočívá v nutnosti určení, kam se analyzovaný bod

$E(x, y, t)$ posunul ve snímku sejmutém v následujícím časovém okamžiku. V předcházející kapitole popsaná metoda určování optického toku popisuje tzv. diferenční metody, které jsou založeny na prostorových a časových parciálních derivacích. Diferenční metody však nejsou jedinou možností, jak optický tok určovat.

Dělení metod, jak určit optický tok, není přesně definováno. V různých literaturách je dělení prováděno odlišným způsobem a jednotlivé metody jsou různě pojmenovávány. Následuje příklad dělení metod určení optického toku podle [7].

- diferenční metody – založené na parciálních derivacích obrazového signálu
 - Horn-Schunck
 - Lucas-Kanade
 - Buxton-Buxton
 - Blaf-Jepson
- blokové metody
- fázová korelace
- diskrétní optimalizace

5.4 DIFERENCIÁLNÍ METODY

Obecně nepoužívanější jsou metody diferenční. V následujících kapitolách se budeme podobněji věnovat dvěma nejrozšířenějším. Metodě, kterou popsali Horn a Schunck a metodě podle Lucase a Kanadeho.

5.4.1 Horn-Schunckova metoda

Metoda Horna-Schuncka vychází z předpokladu pomalých změn (smoothness constraint), čili z předpokladu, že rychlost pohybu sousedících pixelů je velmi

podobná. Míra, s jakou se optický tok od tohoto předpokladu liší může být pro spojitý obraz v intervalech $x \in (0, M)$, $y \in (0, N)$ vyjádřena pomocí integrálu

$$S = \int_0^N \int_0^M \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 dx dy \quad (5.7)$$

Jak vyplývá z rovnice 1, předpokládáme, že výsledek této rovnice bude nulový. Díky v praxi se vyskytujícímu šumu v obraze, však výsledek nulový nebude. Míru této odchylky můžeme vypočítat opět integrálem na stejných spojitých intervalech

$$C = \int_0^N \int_0^M \left(\frac{\partial E}{\partial x} u + \frac{\partial E}{\partial y} v + \frac{\partial E}{\partial t} \right)^2 dx dy \quad (5.8)$$

Díky těmto odchylkám je vhodné zavést konstantu α , která vyjadřuje úměrnost chybové hodnoty C k šumu v obraze. Celková míra odlišnosti od výše uvedeného předpokladu je tedy

$$f = S + \alpha C \quad (5.9)$$

Po minimalizaci této funkce obdržíme dvě rovnice o dvou neznámých u a v

$$\begin{aligned} E_x^2 u + E_x E_y v &= \alpha \nabla^2 u - E_x E_t \\ E_x E_y u + E_y^2 v &= \alpha \nabla^2 v - E_y E_t \end{aligned} \quad (5.10)$$

Toto je numerické řešení pro spojitý obraz. Protože však obraz zaznamenaný videokamerou je spojitý, musíme ho aproximovat Laplaciány

$$\nabla^2 u = (\bar{u} - u) \quad \nabla^2 v = (\bar{v} - v)$$

kde $\bar{u}$ a $\bar{v}$ jsou lokální průměry dané konvolučním jádrem:

$$\begin{pmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & -1 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{pmatrix}$$

S použitím této aproximace je možné rovnice (5.10) zapsat jako

$$\begin{aligned}(\alpha + E_x^2)u + E_x E_y v &= (\alpha \bar{u} - E_x E_t) \\ E_x E_y u + (\alpha + E_y^2)v &= (\alpha \bar{v} - E_y E_t)\end{aligned}\quad (5.11)$$

Řešením pro u a v je

$$\begin{aligned}u &= \frac{(\alpha + E_y^2)\bar{u} - E_x E_y \bar{v} - E_x E_t}{(\alpha + E_x^2 + E_y^2)} \\ v &= \frac{-E_x E_y \bar{u} + (\alpha + E_x^2)\bar{v} - E_y E_t}{(\alpha + E_x^2 + E_y^2)}\end{aligned}\quad (5.12)$$

Protože však výpočet pro každý pixel obrazu by byl hardwarově velice náročný, je v tomto případě vhodné použít některou z iteračních metod. Například Gauss-Seidelovu metodu.

$$\begin{aligned}u^{n+1} &= \bar{u}^n - E_x \frac{E_x \bar{u}^n + E_y \bar{v}^n + E_t}{(\alpha + E_x^2 + E_y^2)} \\ v^{n+1} &= \bar{v}^n - E_y \frac{E_x \bar{u}^n + E_y \bar{v}^n + E_t}{(\alpha + E_x^2 + E_y^2)}\end{aligned}\quad n \dots \text{pořadí iterace} \quad (5.13)$$

Hodnota složek vektoru rychlosti je výsledkem rozdílu průměrné hodnoty rychlosti dané složky v daném okolí určené předchozí iterací a hodnota příslušné směrové složky gradientu jasu v daném bodě, vypočtená na základě hodnot optického toku z předchozí iterace a gradientu jasu v odpovídajícím bodě.

Podrobněji je popsána metoda vysvětlena v [6].

5.4.2 Lucas-Kanadeova Metoda

Tato metoda opět vychází z aperturového problému. Předpokládá, že všechny členy parciálních derivací jasových funkcí obrazu jsou v definovaném, malém okně o velikosti $m \times m$ (okno tedy obsahuje $n = m^2$ prvků – pixelů) konstantní. Z tohoto okna, které může mít velikost 3×3 nám tedy vznikne 9 různých rovnic (pro každý pixel jedna)

$$\begin{aligned} E_{x1}u + E_{y1}v + E_{t1} &= 0 \\ E_{x2}u + E_{y2}v + E_{t2} &= 0 \\ &\vdots \\ E_{xn}u + E_{yn}v + E_{tn} &= 0 \end{aligned} \quad (5.14)$$

což lze maticově zapsat jako

$$\mathbf{A}\mathbf{v} = -\mathbf{B} \quad (5.15)$$

kde

$$\mathbf{A} = \begin{bmatrix} E_{x1} & E_{y1} \\ E_{x2} & E_{y2} \\ \vdots & \vdots \\ E_{xn} & E_{yn} \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} u \\ v \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} E_{t1} \\ E_{t2} \\ \vdots \\ E_{tn} \end{bmatrix}$$

Tuto soustavu n o dvou neznámých u a v budeme řešit pomocí minimalizace metody nejmenších čtverců

$$\sum_{i=1}^n (E_{xi}u + E_{yi}v + E_{ti})^2 \quad (5.16)$$

úpravami vycházejícími z metody nejmenších čtverců obdržíme

$$\underline{A}^T \underline{A} \mathbf{v} = \underline{A}^T (-\underline{B}) \quad (5.17)$$

tedy

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum E_{xi}^2 & \sum E_{xi}E_{yi} \\ \sum E_{xi}E_{yi} & \sum E_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum E_{xi}E_{ti} \\ \sum E_{yi}E_{ti} \end{bmatrix} \quad (5.18)$$

kde všechny sumy jsou přes pixely výše definovaného okna, tedy $i = 1, 2, \dots, n$

Do řešení optického toku pomocí metody Lucas-Kanade bývá často zahrnována tzv. váhovací funkce $W(x, y)$, která jednotlivým pixelům vyšetřovaného okna přiděluje váhu. Prostřednímu pixelu logicky největší. Výsledná soustava rovnic tedy může mít tvar

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum W^2 E_{xi}^2 & \sum W^2 E_{xi}E_{yi} \\ \sum W^2 E_{xi}E_{yi} & \sum W^2 E_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum W^2 E_{xi}E_{ti} \\ \sum W^2 E_{yi}E_{ti} \end{bmatrix} \quad (5.19)$$

kde jsou opět všechny sumy pro $i = 1, 2, \dots, n$

Podrobnější popis metody je opět možné nalézt v [6].

5.4.3 Srovnání metod Horn-Schunck a Lucas-Kanade

Výsledkem metody Horn-Schunck je tzv. husté vektorové pole. Metoda tedy vyplní vektory optického toku i vnitřní oblasti homogenních oblastí. To se děje na základě znalostí vektorů optického toku na hranicích takových homogenních oblastí. Naproti tomu metoda Lucas-Kanade produkuje výsledek ve formě vektorů optického toku, které jsou však detekovány pouze na hranicích a obrysech předmětů. Logicky vyplývá, že znalost hustého vektorového pole je výhodou. Ovšem převažující nevýhodou metody Horn-Schunck je poměrně vysoká citlivost na šum ve zpracovávaných obrazech, kdežto metoda Lucas-Kanade je robustnější a v našem případě tedy k praktickému použití vhodnější.

5.5 BLOKOVÉ METODY

Blokové metody jsou také někdy označovány jako metody založené na oblastech. Díky tomu, že diferenciální metody jsou poměrně náchylné na šum nebo aliasing vznikající při záznamu videa, nejsou vždy k řešení optického toku nejvhodnější. V těchto případech je výhodnější použít blokovou metodu. Tyto metody přistupují k problému s využitím posunutí $\mathbf{d} = (d_x, d_y)$ namísto rychlosti $\mathbf{v} = (v_x, v_y)$. V následujícím textu si krátce popíšeme Anandanovu metodu.

5.5.1 Anandanova metoda

Tato metoda je podrobněji popsána v [8]. Je založena na podobnosti čtvercových oblastí. K nalezení řešení se využívá metoda rozdílu nejmenších čtverců (sum-of-square difference SSD) a Laplaceovy pyramidy.

$$SSD_{1,2}(\mathbf{x}, \mathbf{d}) = \sum_{j=-n}^n \sum_{i=-n}^n W(i, j) \times [E_1(\mathbf{x} + (i, j)) - E_2(\mathbf{x} + \mathbf{d} + (i, j))]^2 \quad (5.20)$$

kde W symbolizuje výřez 2-D okna, (i, j) je vektor použitý k procházení tohoto okna, $\mathbf{d}$ je vektor posunutí a $\mathbf{x}$ jsou souřadnice bodu.

Existuje úzká souvislost mezi metodou rozdílů nejmenších čtverců a diferenciálními technikami. Minimalizování vzdáleností SSD přináší stejné výsledky jako maximalizace integrálu výsledného členu $E_1(\mathbf{x})E_2(\mathbf{x} + \mathbf{d})$.

5.6 METODY ZALOŽENÉ NA ENERGII

Někdy jsou také nazývány jako metody založené na frekvenci. Jsou tvořeny filtry zpracovávajícími rychlostní vektory. Z odezvy (energie) jednotlivých filtrů jsou určeny rychlosti jednotlivých pixelů. Při výpočtech je používána Fourierova transformace.

$$\hat{E}(\mathbf{k}, \omega) = \hat{E}_0(\mathbf{k})\delta(\omega + \mathbf{v}^T \mathbf{k}) \quad (5.21)$$

kde $\hat{E}_0(\mathbf{k})$ je Fourierova transformace $E_0(\mathbf{k})$,

$\delta(k)$ je Diracův impuls,

ω je časová frekvence a $\mathbf{k} = (k_x, k_y)$ je prostorová frekvence

Jeden z možných postupů založených na této metodě popsal Heeger [8].

6. VLASTNÍ ZPRACOVÁNÍ

Práce byla zpracována v programovacím jazyce C++ s využitím volně dostupného souboru knihoven OpenCV¹ pro práci s obrazem. Pro předzpracování videa byl použit freeware program VirtualDub.

6.1 PŘEDZPRACOVÁNÍ VIDEOA

Aplikace, v které budeme úkol řešit pracuje s videi v bezkompresním formátu. Byl tedy zvolen formát MPEG2 v multimediálním kontejneru AVI, který zmíněný program podporuje. Podmínka bezkompresnosti videa je požadována také díky vysokým nárokům na kvalitu zpracovávaného videa, tedy co nejmenší obsah šumu a jiných obrazových chyb. Velké množství těchto chyb vzniká právě při předzpracování videa.

Digitální kamera, která byla k dispozici k natočení vhodných videí, pořizovala videosekvence ve formátu MPEG2, která však byla uložena v kontejneru MOD. Pro převod natočeného materiálu byl použit výše zmíněný volně dostupný program VirtualDub.

Protože však byla videa pořízena prokládací metodou, bylo nutné provést jejich tzv. deinterlacing (viz. kapitola 2.2). Jak již bylo zmíněno výše, deinterlacing byl prováděn metodou Deinterlace MAP, která je součástí již použitého programu VirtualDub. Pokud by byla videa ponechána v prokládané formě, vznikaly by nám v jednotlivých snímcích nečekané rohy a hrany, které by znamenaly značné zkreslení finálních výsledků. Deinterlacing však do snímků videa vnesl mírné rozmazání. Detekce hran a rohů, jak bude popsána níže, bude proto zatížena jistými nepřesnostmi a nedokonalostmi a bude vyžadovat větší výpočetní výkon.

¹ OpenCV – Open Source Computer Vision Library

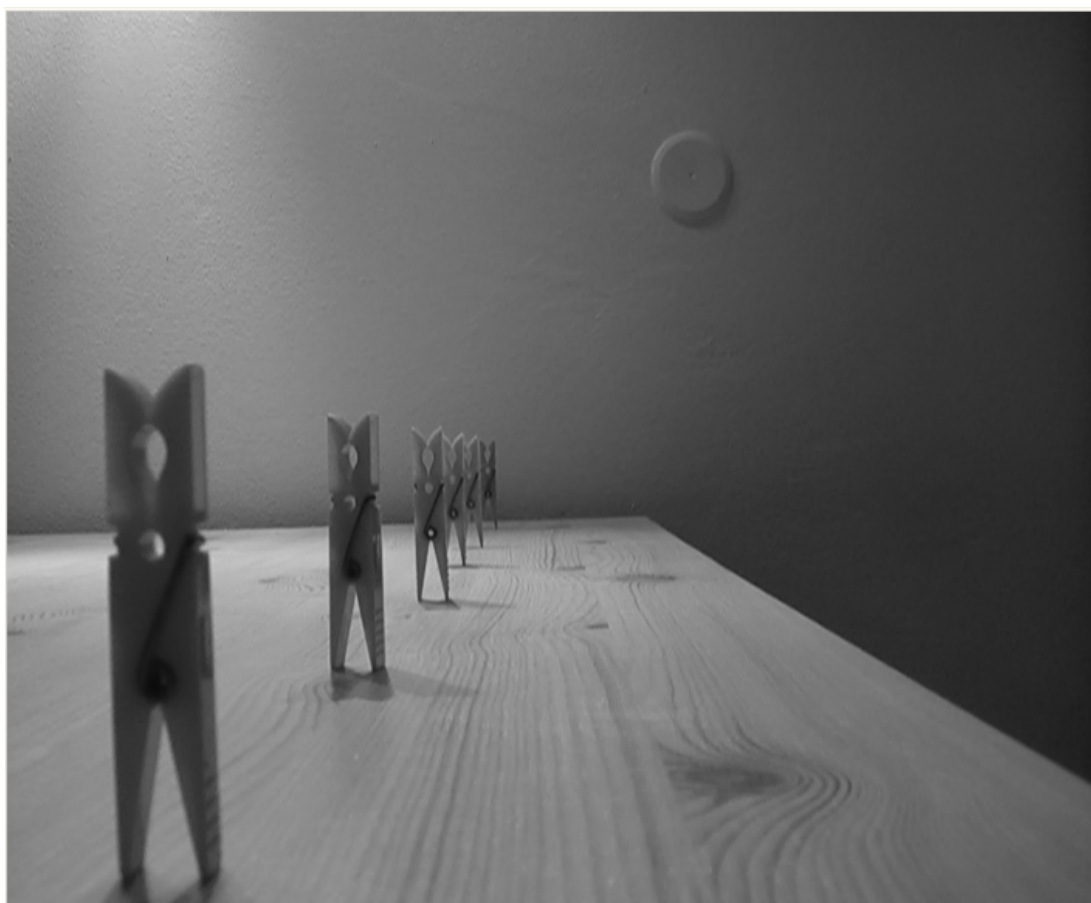
6.2 VLASTNÍ ZPRACOVÁNÍ

V následujícím textu budou popsány algoritmy samotného zpracování videa. Video je zpracováváno postupně po snímcích. V následujícím textu bude jako příklad uváděn jeden ze snímků (resp. pokud to bude zmíněno druhý snímek pořízený za krátký časový interval) zdrojového videa. Tento snímek je zobrazen na následujícím obrázku.



Obr 6.1 Snímek videosekvence.

Tento obrázek je reprezentován barevným modelem RGB. Pro jeho další zpracování bude nutné ho převést na obrázek ve stupních šedi. Tím docílíme, že jeho jasová intenzita bude vždy vyjádřena pouze jedním číslem – hodnotou intenzity jedné barevné složky.



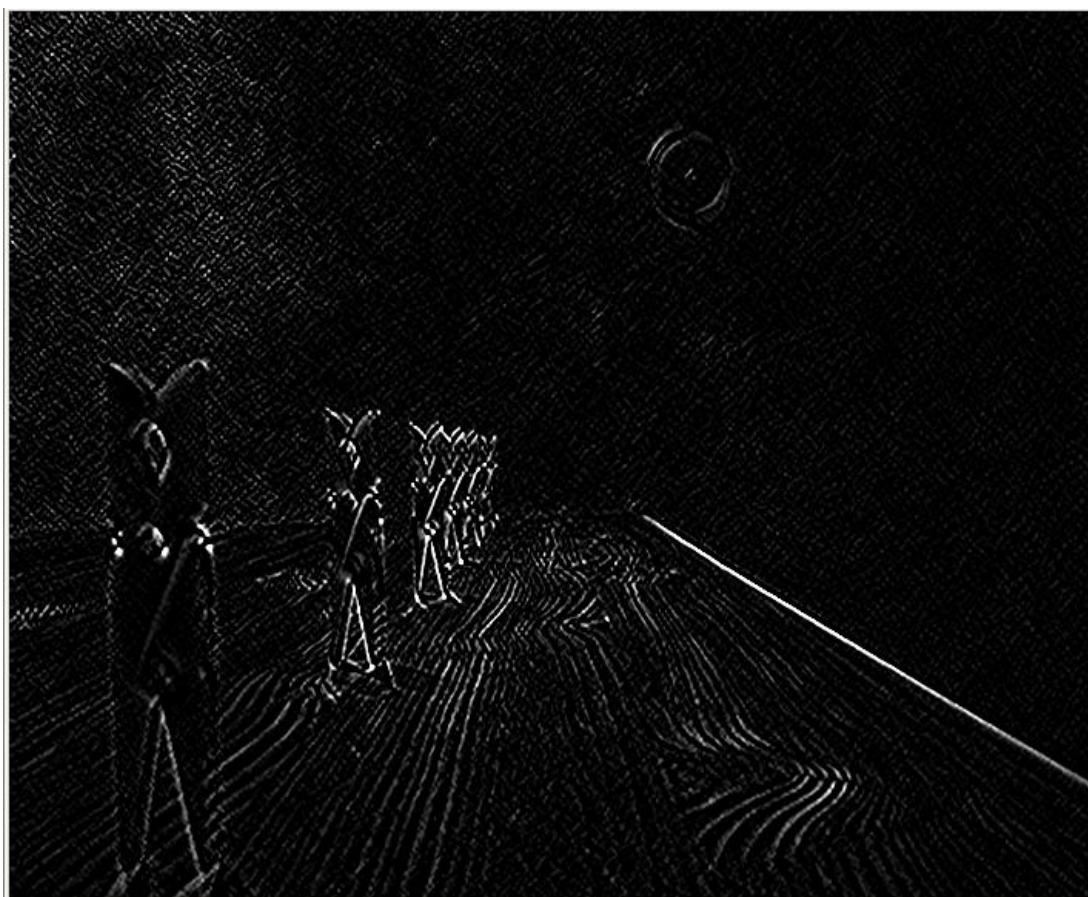
Obr 6.2 Obrázek ve stupních šedi (grayscale).

6.2.1 Detekce hran

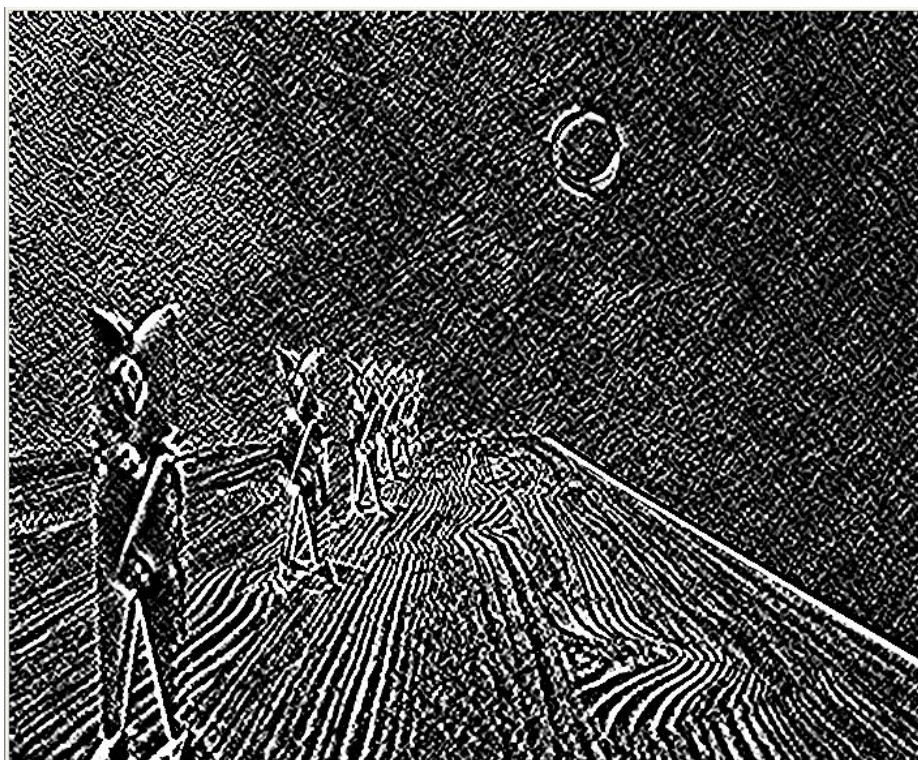
Kvůli pozdějšímu snadnějšímu zpracování se jeví jako vhodné na vstupní obraz nejprve aplikovat hranový filtr. Tento detektor má za úkol nalézt ve zkoumaném snímku místa, s vysokou hodnotou gradientu jasové intenzity. Jeho výsledkem je pole hodnot, kdy pro každý pixel známe hodnotu gradientu ve směru osy x a osy y . Z těchto hodnot lze určit i přesnější směr hrany v daném místě.

V rozšíření OpenCV je implementován Cannyho hranový algoritmus. Tento detektor hran je v současnosti nejpoužívanější a lze ho pravděpodobně považovat i za nejlepší. Jak již bylo popsáno výše, Cannyho algoritmus využívá Sobelův hranový detektor. Sobelův detektor spočívá v aproximaci derivací pomocí konvolučních jader. Aplikací masky je možné vypočíst hodnotu gradientu pro daný pixel obrazu ve

srovnání s jeho okolím. Díky tomu, že náš obraz je poměrně vysoce zašuměný, se jako nejvhodnější jeví použít konvoluční masku velkých rozměrů. Na následujících obrázcích je porovnání výsledků aplikace Sobelova hranového detektoru s konvolučními maskami různých rozměrů.



Obr 6.3 Výsledek detekce hran s využitím Sobelova detektoru s využitím konvolučního jádra 5x5.



Obr 6.4 Výsledek detekce hran s využitím Sobelova detektoru s využitím konvolučního jádra 7x7.

Je patrné, že nejlepšího výsledku je dosaženo při použití konvoluční masky o rozměrech 7x7 pixelů. Výpočet s aplikací masky o rozměrech 5x5 pixelů dosahuje také uspokojivých výsledků, ovšem stále dochází k určitým ztrátám informací o hranách, které se v pozdějším zpracování ukázaly být nevhodné. V případě použití konvolučního jádra o standardních rozměrech 3x3 pixely nedojde k detekci takřka žádné hrany. Výsledný obrázek tedy nemá smysl uvádět. Nefunkčnost nejmenší masky je dána výraznou přítomností šumu a rozmazání hran ve zdrojovém obraze.

Dalším parametrem Cannyho hranového detektoru, který můžeme ovlivnit je prahování. Jak bylo opět zmíněno v kapitole zabývající se hranováním, tento algoritmus využívá tzv. hysterezní prahování. Používají se tedy dva prahy. Hodnota větší, než je větší práh, označí dané místo jako hranu, hodnota menší než menší práh toto místo jako hranu vyřadí. Pokud je hodnota gradientu mezi oběma prahy,

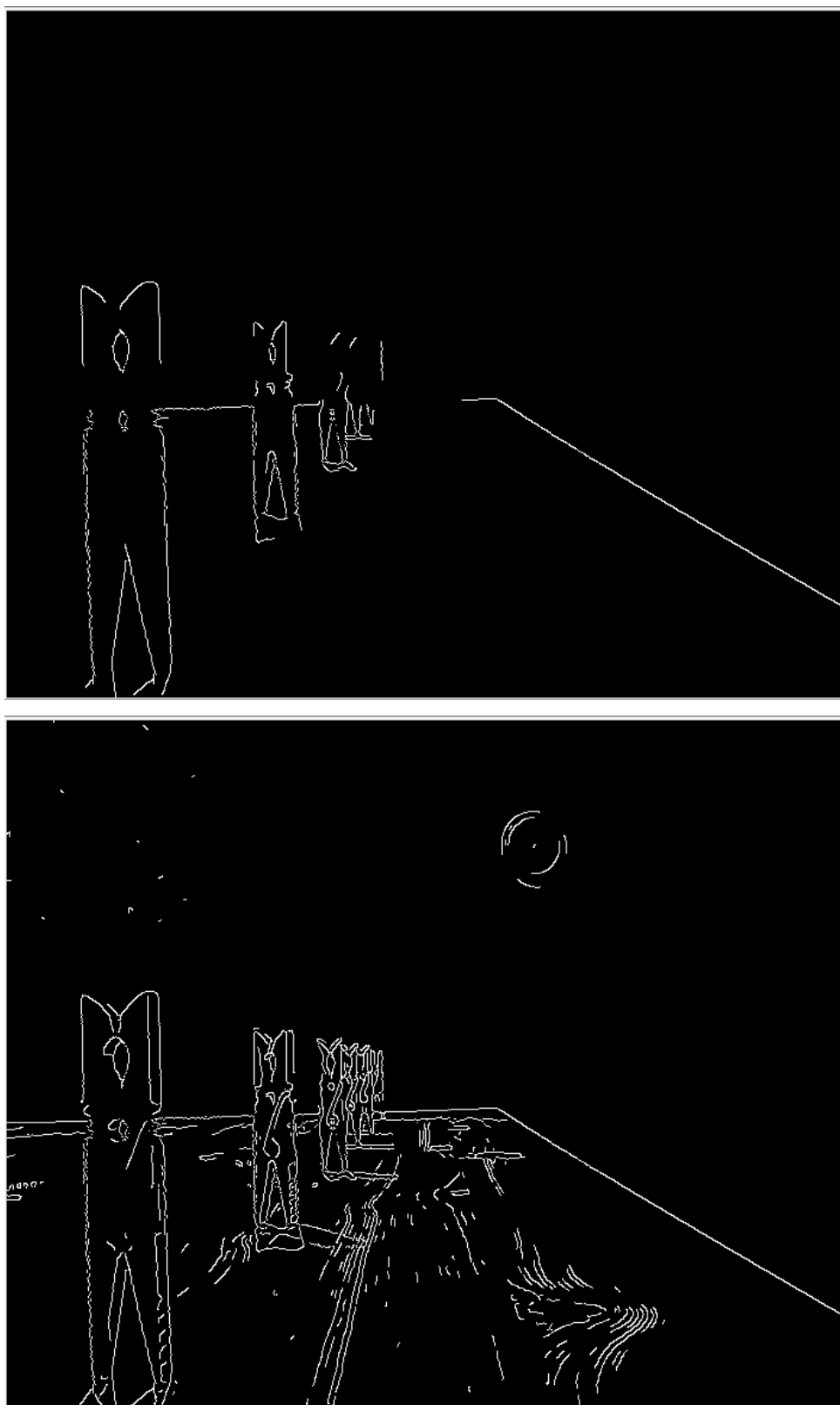
rozhoduje o příslušnosti bodu k hraně jeho okolí. Pokud ve svém sousedství má pixel již dříve označený jako hranu, je i tento bod jako hrana označen.

Vhodnou volbou prahů lze tedy docílit toho, aby ve výsledném obrazu byly označeny všechny skutečně významné hrany a naopak, aby byly odfiltrovány parazitní jasové přechody, které nejsou pro účel našeho rozboru obrazu relevantní. Bohužel se nastavení prahů liší pro každou vyšetřovanou video scénu a díky měnícím se světelným podmínkám a ostření kamery i pro jednotlivé snímky jedné sekvence. Pro zcela autonomní funkci sledování objektů v obraze by tedy bylo žádoucí automaticky měnit hodnoty prahů v závislosti na změnách podmínek. Toto se však v této práci díky své poměrně vysoké náročnosti nepodařilo implementovat. Pokud ponecháme prahy konstantní, bude i tak dosahováno uspokojivých výsledků.

Na následujících obrázcích je demonstrováno různé nastavení prahových hodnot Cannyho hranového filtru a jeho vliv na kvalitu detekce hran.



Obr 6.5 Výsledek Cannyho hranového deektoru s vhodně nastavenými prahy.



Obr 6.6 Výsledek Cannyho hranového detektoru s nevhodně zvolenými prahy.

Na Obr 6.5 jsou prahy detekce zvoleny vhodně. Ačkoli dochází k označení i některých irelevantních hran, jejich počet není nijak markantní, naopak většina významných hran byla detekována. V případě Obr 6.6 nahoře byly prahy voleny příliš vysoké, došlo tedy k vyřazení i důležitých hran a tím degradaci obrazu, zatímco na spodním obrázku byly prahy voleny příliš nízké a do výsledného obrazu bylo zahrnuto i velké množství nevýznamných hran, které ve skutečnosti nejsou hranami, nýbrž představují strukturu dřeva.

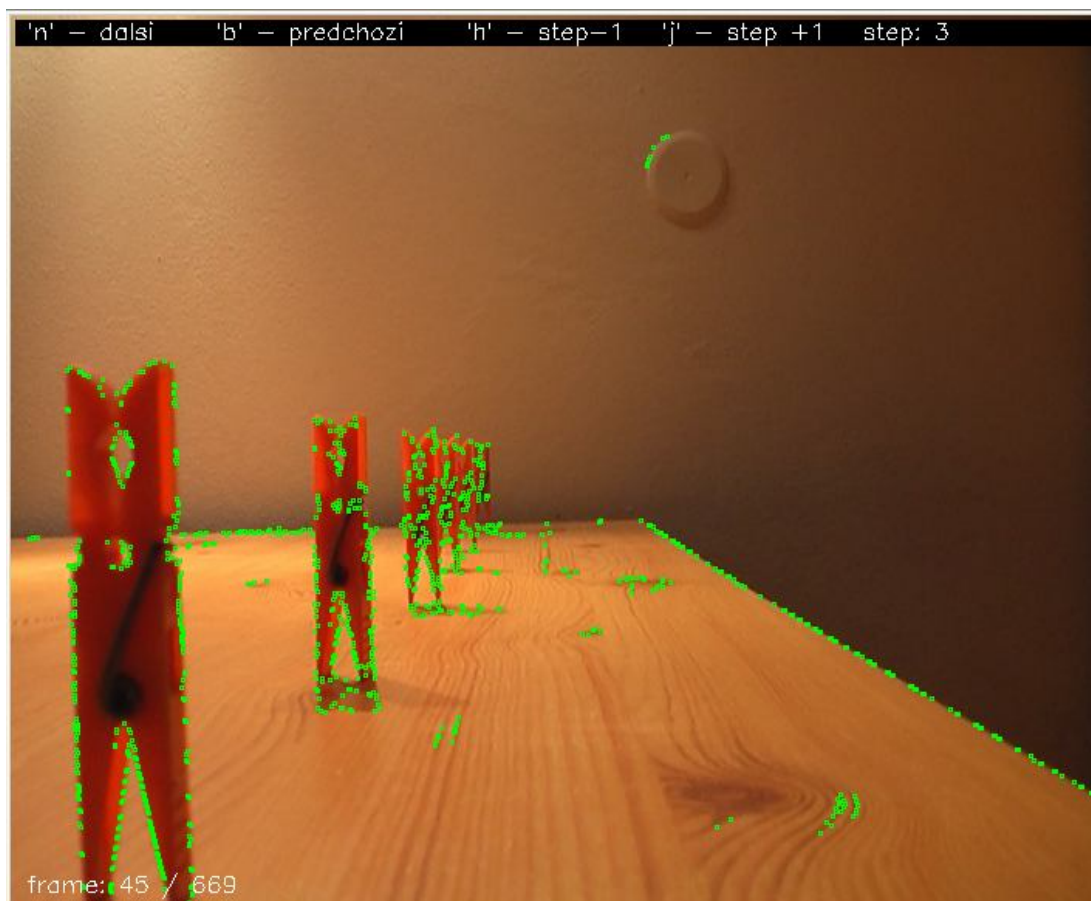
6.2.2 Výpočet optického toku

Dalším krokem zpracování obrazu je určení optického toku. Při této úloze se ukázalo být jako nejvhodnější použití kombinace metod blokového vyhledávání a diferenční metody. Připomeňme, že cílem optického toku je určit vektory pohybu v obraze. Pohyb, který metodou optického toku nalezneme, však ve skutečnosti vůbec nemusí existovat, jedná se tedy o pohyb zdánlivý. Tento pohyb je však možné využít pro detekci objektů, pokud jsou tyto od sebe dostatečně vzdáleny a jasně odlišeny.

Pro vyhledávání vektorů optického toku je nutné mít k dispozici sekvenci snímků – video. Zaznamenaný pohyb by měl mít optimální rychlost. Zvláště nevýhodný je rychlý pohyb, kdy dochází mimo jiné k problémům se snímáním obrazu. Výsledný obraz bývá rozmazaný a nejasný. Pokud máme zaznamenaný pomalý pohyb, zrychlit ho můžeme jednoduše tím, že vynecháme některé snímky z celkové sekvence. Zpomalit takto pohyb ovšem nelze, protože informaci, která nebyla zaznamenána při snímání již nemůžeme získat, nebo-li abychom mohli pohyb zpomalit, museli bychom mezi dva sousední snímky vložit snímek další, tento ovšem nemáme k dispozici.

Prvním krokem určení vektorů optického toku je nalezení významných bodů v prvním ze snímků. K nalezení těchto bodů slouží funkce implementovaná v rozšíření OpenCV. Tato funkce najde body s největší vlastní hodnotou vektorového prostoru, tedy body s velkým gradientem jasové intenzity. Tyto body jsou vyhledávány z binárního obrazu hran, který byl popsán výše. Na následujícím

obrázku jsou body, které tato funkce našla, pro názornost vykresleny do původního, neupraveného snímku.

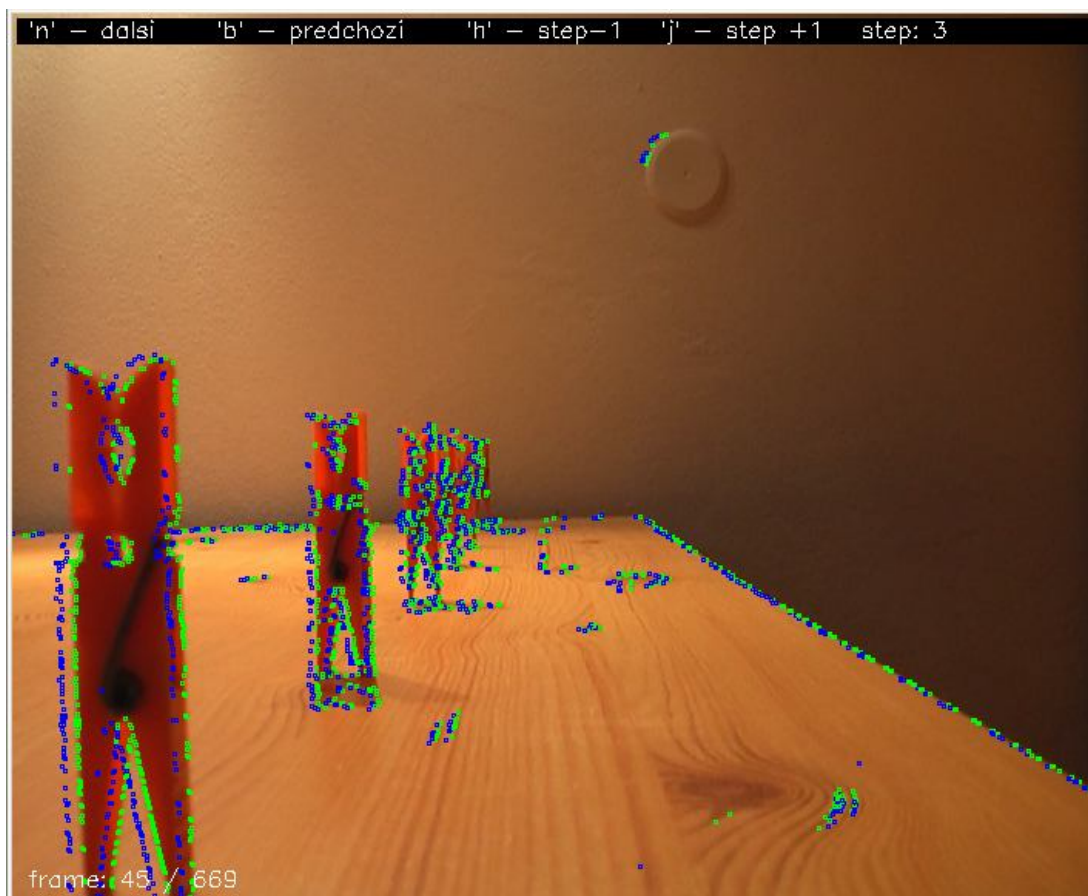


Obr 6.7 Významné body nalezené ve snímku.

Dalším krokem je nalezení bodů, které byly označeny jako významné v následujícím snímku. K tomu opět slouží funkce integrovaná v OpenCV. Tato funkce se pokouší pro každý bod nalezený na předcházejícím snímku nalézt odpovídající bod na snímku následujícím. Tyto body jsou počítány pomocí metody Lucas-Kanade využívající Gaussovské pyramidy.

Tímto postupem tedy získáme dvojice na sousedních snímcích si odpovídajících bodů. Pokud tyto body vykreslíme do jednoho obrázku, bude jasně patrné, kam se jaký bod zobrazí v následujícím snímku. Nalezené body leží na pomyslné přímce pohybového vektoru. Na následujícím obrázku odpovídají zeleně

označené body významným místům nalezeným v zobrazeném snímku (tedy stejné jako na Obr 6.7) a modré body potom představují korespondující body nalezené na snímku následujícím. Ukazují tedy, kam se jaký bod zobrazí ve snímku následujícím.

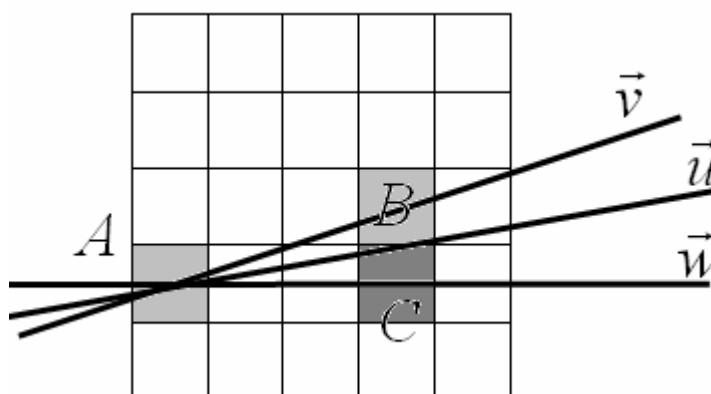


Obr 6.8 Významné body nalezené ve dvou po sobě následujících snímcích.

Po skončení jednoho cyklu běhu programu jsou vždy body, které byly nalezeny metodou optického toku (pomyslné druhé body z každé dvojice) označeny jako významné body nalezené na prvním ze snímků. Hledání významných bodů tedy probíhá pouze jednou a to na prvním snímku. V dalších krocích programu se pracuje pouze s těmito nalezenými body a hledá se řetězec odpovídajících si (posunujících se) bodů.

Tímto postupem byly získány dva body, které leží na přímce, která reprezentuje pohybový vektor v daném místě. Musíme si ovšem uvědomit, že tyto

body nejsou k získání přesného pohybového vektoru ideální. Vzhledem k tomu, že celý obraz je diskrétní, kdy jednotlivé prvky jsou poskládány do vzorkovací mřížky, máme k dispozici pouze omezené rozlišení vzniklého obrazu. Pokud tedy máme pohybový vektor charakterizovaný dvěma body, které ve zmíněné mřížce leží příliš blízko sebe, nemůžeme výsledný vektor určit s dostatečnou přesností. Tato chyba je ilustrativně naznačena na obrázku Obr 6.9. Uvažujme jako správný vektor $\vec{u}$. Vzorkováním je jasně přiřazen bod A . Ovšem pokud druhý bod leží na hranici mezi dvěma vzorky, je možné ho zobrazit buď jako bod B nebo jako bod C . Tím nám vzniknou vektory $\vec{v}$ nebo $\vec{w}$. Je na první pohled zřejmé, že oba se od správného vektoru $\vec{u}$ liší nezanedbatelným způsobem.



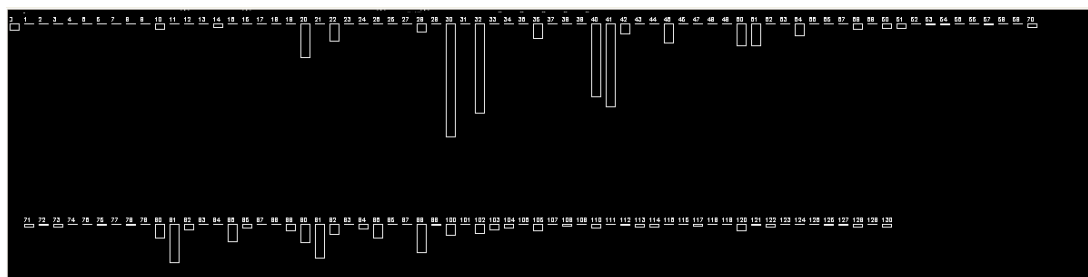
Obr 6.9 Chyba určení vektoru ze dvou blízkých bodů.

Tato skutečnost bude velmi komplikovat další výpočty a do výsledků bude vnášet velikou chybu.

6.2.3 Detekce objektů

V další fázi programu lze z v předchozí úloze určených bodů detekovat objekty. Nejprve určíme, o jak velký úsek se který bod mezi dvěma sousedními snímky posune, tedy vzdálenost mezi body jednotlivých dvojic. Tato vzdálenost je počítána prostou Pythagorovou větou a vyjádřena desetinným číslem, tedy neceločíselně. Dalším krokem potom je vytvoření histogramu těchto vzdáleností.

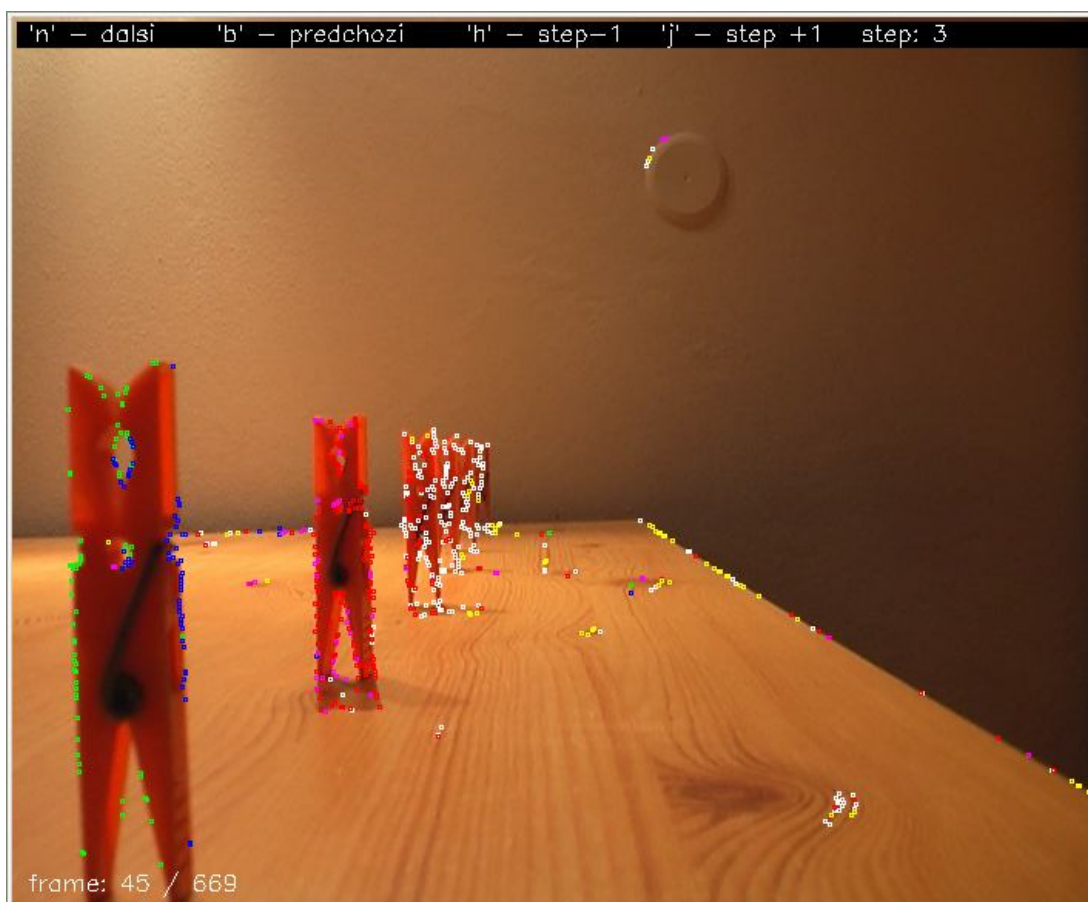
Histogram ukazuje četnost výskytu posunu bodu o danou velikost s přesností 0,1. Je znázorněn na následujícím obrázku.



Obr 6.10 Histogram posunutí nalezených významných bodů mezi sousedními snímky.

V souladu s teorií perspektivního zobrazení, kdy za předpokladu v reálné scéně stejně rychle se pohybujících objektů (v našem případě analogie se stacionárními objekty a pohybujícím se pozorovatelem – kamerou) se rychlost objektů od pozorovatele vzdálenějších jeví jako menší než objektů bližších, můžeme uvažovat, že body, které se mezi snímky posunuly o stejný úsek, náleží jednomu objektu.

S histogramem je pracováno, tak že se vyhodnotí vždy čtveřice jeho prvků. Jsou určeny součty hodnot všech takových čtveřic a z nich je vybráno šest nejvýznamnějších. Těchto šest skupin je dále testováno mezi sebou. Pokud se navzájem překrývají, je méně významná čtveřice vypuštěna a na její místo je vložena další nejvýznamnější z původního seznamu. Tímto postupem je tedy získáno šest vzájemně se nepřekrývajících čtveřic prvků histogramu. Každou takovou čtveřici budeme uvažovat jako uskupení bodů, které se posunuly o stejnou vzdálenost. Pokud body náležející jednotlivým skupinám vykreslíme, získáme následující obraz (Obr 6.11). Pro větší přehlednost jsou vykreslovány pouze odpovídající významné body nalezené na zobrazeném snímku. Jednotlivé skupiny (čtveřice prvků z histogramu) jsou vykreslovány různými barvami. Jak je z tohoto obrázku patrné, body dané skupiny poměrně kvalitně nalézají objekty.



Obr 6.11 Výsledná detekce objektů.

Některé objekty jsou určeny více skupinami bodů (na Obr 6.11 například modré a zelené body). To je zapříčiněno tím, že zobrazený objekt se nachází příliš blízko ke kameře, zobrazuje se jako větší, pohybuje se (ve smyslu optického toku) tedy rychleji a projevuje se na něm výše popsáný efekt rozdílu rychlostí při různé vzdálenosti od kamery.

V příloze A jsou uvedeny příklady snímků s detekovanými objekty z dalších videí. Video, která jsou v této příloze uvedena byla pořízena za lepších světelných podmínek a ve smyslu okolním prostředím méně rušené kvalitnější scéně.

Jak je i podle příkladů z přílohy A patrné, detekce objektů není vždy dokonalá. Největší problém nastává, pokud jsou objekty stejně nebo velice podobně vzdáleny od kamery. Vektory posunu bodů nalezených na takových objektech mají totiž stejnou velikost a použitým algoritmem jsou tedy vyhodnoceny jako vektory

náležící jednomu objektu. Tento problém by bylo možné odstranit dalším filtrováním třízením nalezených bodů.

Pomocí nalezených významných bodů a jejich přiřazení k jednotlivým objektům je tedy možné sledovat trajektorii nasnímaných objektů.

6.2.4 Sledování trajektorie

Body, které byly označeny jako významné jsou v každém kroku ukládány. Uchovávají se tím tedy souřadnice všech míst, kudy se sledované body pohybovaly. Spojíme-li tyto body úsečkami, vznikne nám křivka ukazující historii umístění jednotlivých bodů, tedy trajektorii sledovaného objektu. Příklad výsledné trajektorie je na následujícím obrázku.



Obr 6.12 Příklad zobrazení trajektorie.

7. ZÁVĚR

V průběhu vypracování této práce jsem se seznámil se základy dané tematiky digitálního zpracování videa, detekce hran, metodiky určování optického toku.

Z množství dostupných postupů detekce optického toku jsem zvolil diferenční metodu Lucas-Kanade kombinovanou s blokovou metodou. Kombinace těchto metod se jeví jako dostatečně robustní a precizní při určování optického toku a zároveň není příliš výpočetně náročná. Z informace o vektorech zdánlivého pohybu detekovaných významných bodů se podařilo separovat jednotlivé objekty. Zde ovšem nastává slabina daného řešení, která byla naznačena v textu. Navržený algoritmus není schopen odlišit objekty, které se pohybují podobnými rychlostmi (podobně vzdálené od kamery) a není proto vhodný k vyšetřování videí, v nichž tento jev nastává, jedná se především o videa, kdy je osa kamery kolmá na osu pohybu a objekty jsou od kamery konstantně vzdáleny. Naopak vhodná se tato metoda jeví pro videa, v nichž se kamera pohybuje směrem ke zkoumaným objektům a tyto objekty jsou v každém okamžiku od kamery různě daleko. Zvolená metoda má také tu výhodu, že zkoumané objekty nemusí mít jednotný povrch, jejich struktura může být poměrně složitá a mohou být zobrazeny i složitější texturou s výraznými jasovými přehody.

Výběr z nasnímaných videí vytvořil databázi, která se nachází na přiloženém DVD nosiči a je popsána v příloze B.

V práci se nabízí velké množství jejích rozšíření a doplnění. Jako velice žádoucí se jeví rozšíření o schopnost odlišení jednotlivých objektů, které jsou od kamery stejně vzdáleny. Dále poté propracovanější záznam trajektorie jednotlivých objektů, který by umožnil vyhodnocování jejich stavů jako velikost, směr pohybu nebo rychlost.

8. LITERATURA

- [1] HORÁK, K.; KALOVÁ, I.; PETYOVSKÝ, P.; RICHTER, M.
Počítačové vidění. Brno: 2007. s. 1-132.
- [2] KRŠEK P., ŠPANĚL M., Presentace k přednášce Základy počítačové grafiky – Barvy a barevné modely, Brno: FIT VUT, 2007, 19 str.,
[cit. 2009-04-20]
- [3] KRŠEK P., ŠPANĚL M., Presentace k přednášce Základy počítačové grafiky – Geometrické transformace ve 2D a 3D, Brno: FIT VUT, 2008, 57 str., [cit. 2009-04-20]
- [4] Prokládání – interlace/deinterlace
URL <<http://jech.webz.cz/deinterlace.php>>, [cit. 2009-04-22]
- [5] Vanishing point
URL <http://en.wikipedia.org/wiki/Vanishing_point>, last modified on 11 May 2009, [cit. 2009-04-25]
- [6] ŘÍHA, K. *Pokročilé techniky zpracování obrazu*. Skripta FEKT VUT, ev.č. TKO 07-083. 2007. s. 1-105.
- [7] Optical Flow
URL <http://en.wikipedia.org/wiki/Optical_flow>, last modified on 28 April 2009, [cit. 2009-05-01]
- [8] BARRON J. L. Performance of optical flow techniques. *International Journal of Computer Vision* [online]. Volume 12, Number 1 / Feb, 1994. [cit. 2009-05-19]. Vydavatel Springer Netherlands. str 43 – 77.
Dostupné z
<<http://www.springerlink.com/content/w512071331p37127/>>
- [9] ZIOU D., TABBONE S., Edge Detection Techniques - An Overview, *International Journal of Pattern Recognition and Image Analysis* [online]. Volume 8, 1998. [cit. 2009-05-20]. str. 537 – 559. Dostupné z
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.1821&rep=rep1&type=pdf>
- [10] Detektory hran

URL <http://cs.wikipedia.org/wiki/Detektory_hran>, poslední úpravy
17. 5. 2009, [cit. 2009-05-20]

[11] Cannyho hranový detektor

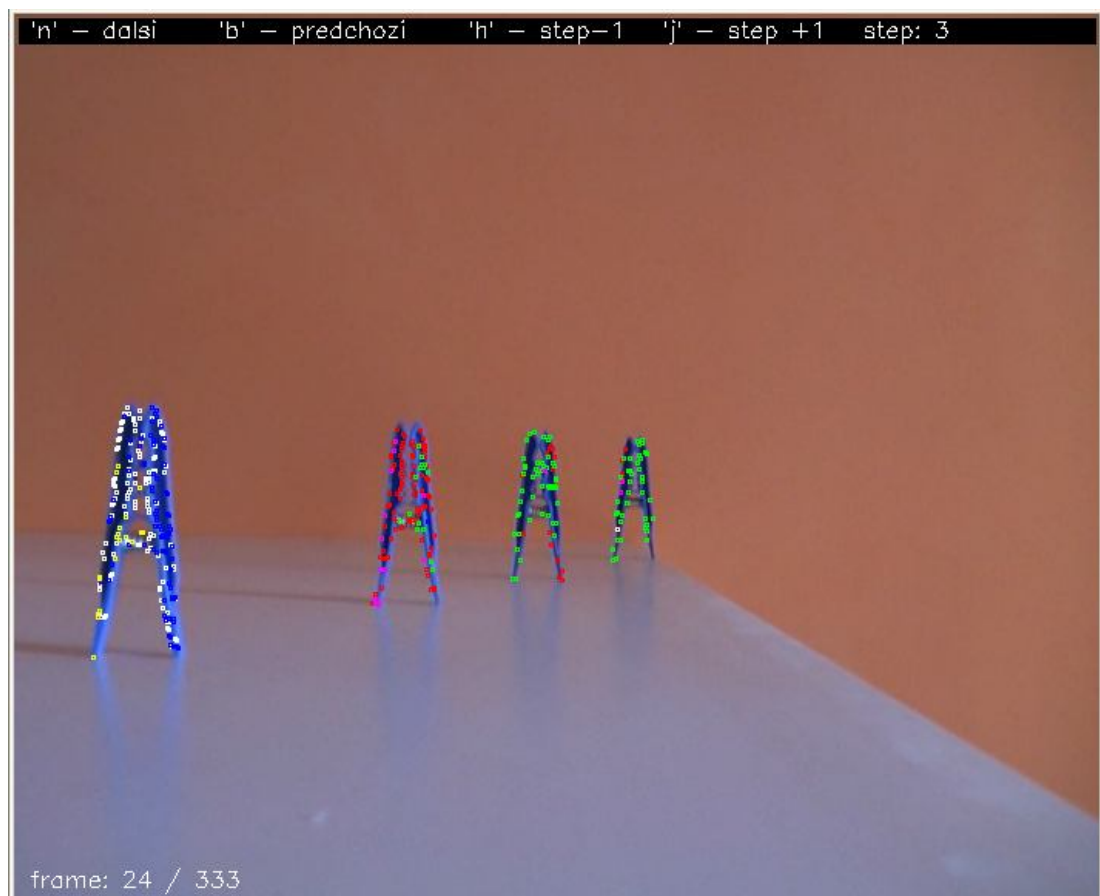
URL <http://cs.wikipedia.org/wiki/Cannyho_hranový_detektor>,
poslední úpravy 18. 1. 2008, [cit. 2009-05-20]

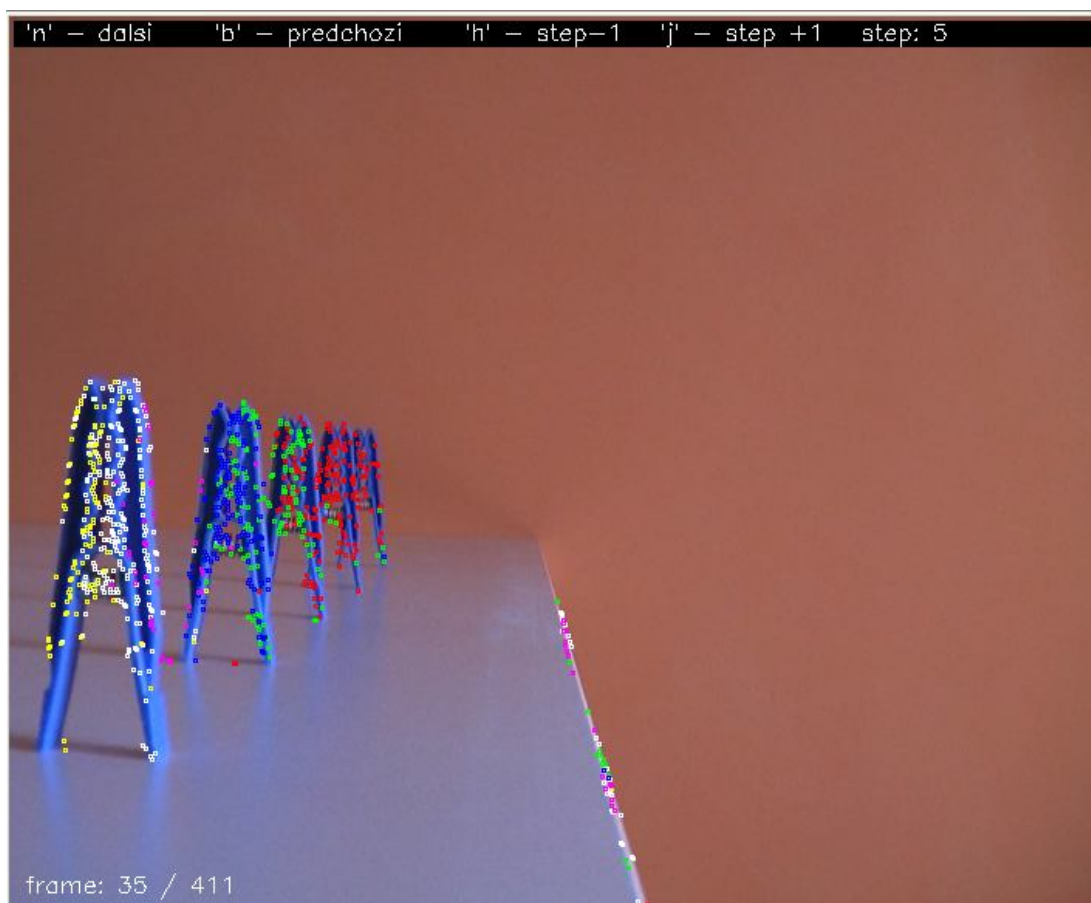
SEZNAM PŘÍLOH

Příloha A	Příklady výsledků detekce objektů
Příloha B	Databáze videí
Příloha C	Popis programu
Příloha D	Obsah přiloženého DVD

PŘÍLOHA A: PŘÍKLADY VÝSLEDKŮ DETEKCE OBJEKTŮ

Příklady dalších výsledků detekce objektů – sledování trajektorie.





'n' - dalsi 'b' - predchozi 'h' - step-1 'j' - step +1 step: 3



'n' - dalsi 'b' - predchozi 'h' - step-1 'j' - step +1 step: 2



PŘÍLOHA B: DATABÁZE VIDEÍ

Popis databáze nasnímaných videí.

Součástí práce bylo nasnímání databáze videí, která budou následně analyzována. Kvalitní nasnímání videí se ukázalo být pro celkovou zpracovatelnost a získání uspokojivých výsledků rozhodující. V první fázi návrhu jsem se pokoušel o nasnímání reálných scén „z běžného života“. Tento postup se ovšem ukázal být nevhodný, protože vzniklá videa byla velice špatně zpracovatelná. Nakonec se jako nejoptimálnější ukázal způsob natočení videí, která svým charakterem budou více odpovídat laboratorním podmínkám. Videa proto byla točena s využitím jednodušších objektů umístěných na jednobarevné ploše s co nejvíce konstantním pozadím.

Videa byla snímána digitální videokamerou Panasonic SDR-H250EP v maximální možné kvalitě, s využitím vestavěného stabilizátoru obrazu ve formátu 4:3.

Finální databáze videí je popsána v následující tabulce.

tabulka 1: Databáze videí.

název souboru	popis scény
video1.avi	osa kamery rovnoběžná se směrem pohybu objekty umístěny hustě v řadě objekty v linii rovnoběžné se směrem pohybu
video2.avi	osa kamery rovnoběžná se směrem pohybu objekty umístěny řídce v řadě objekty v linii rovnoběžné se směrem pohybu
video3.avi	osa kamery odchýlena od směru pohybu směrem k objektům objekty umístěny řídce v řadě objekty v linii rovnoběžné se směrem pohybu
video4.avi	osa kamery rovnoběžná se směrem pohybu objekty umístěny ve dvou řadách – kamery projíždí mezi nimi dvojice objektů mají stejnou vzdálenost od kamery
video5.avi	osa kamery kolmá na směr pohybu objekty umístěny řídce v řadě

	objekty v linii rovnoběžné se směrem pohybu
video6.avi	osa kamery kolmá na směr pohybu objekty umístěny v několika řadách objekty v několika liniích rovnoběžných se směrem pohybu
video7.avi	osa kamery odchýlená od kolmice ke směru pohybu objekty umístěny v jedné řadě objekty v linii rovnoběžné se směrem pohybu
video8.avi	osa kamery kolmá na směr pohybu 3 objekty umístěny v linii kolmé na směr pohybu

PŘÍLOHA C: POPIS PROGRAMU

Součástí bakalářské práce bylo vytvořit program pro demonstraci funkčnosti navržených algoritmů. Program je napsán v programovacím jazyce C s využitím některých prvků C++ a v textu zmíněné knihovny OpenCV¹. Je napsán jako konzolová aplikace využívající grafická okna. Jeho funkčnost je odzkoušena na operačním systému MS Windows XP.

Program je určený pro spouštění s následujícími pěti parametry:

```
PathMonitoring.exe video start_frame step thres1 thres2
```

kde video ... cesta k souboru videa včetně přípony, tedy například video/video1.avi

start_frame ... pořadí snímku od kterého se program spustí (není ošetřeno zadání čísla většího než je celkový počet snímků daného videa), doporučená hodnota je 0

step ... velikost kroku mezi jednotlivými vyšetřovanými snímky. Doporučená hodnota viz tabulka 2.

thres1 ... velikost menšího prahu pro prahování při algoritmu Cannyho hranového detektoru (viz. 4.1). Doporučená hodnota viz. tabulka 2.

thres2 ... velikost většího prahu pro prahování při prahování Cannyho hranového detektoru (viz. 4.1). Doporučená hodnota viz. tabulka 2.

tabulka 2: Doporučené nastavení programu pro jednotlivá videa.

název souboru	Thres1	thres2	step
video1.avi	7000	15000	5
video2.avi	7000	15000	4
video3.avi	7000	15000	3
video4.avi	7000	15000	3
video5.avi	7000	15000	2

¹ OpenCV – Open Source Computer Vision Library

video6.avi	5000	10000	2
video7.avi	5000	10000	2
video8.avi	5000	10000	1

Klávesou ‚n‘ posuneme video o počet snímků definovaný jako `step`. Dochází tak k „přehrávání“ videa.

Během běhu programu je možné měnit velikost kroku mezi dvěma vyhodnocovanými snímky. Klávesou ‚j‘ dojde k inkrementaci kroku, klávesou ‚h‘ potom k dekrementaci. Nevhodně zvolená velikost kroku se může negativně projevit na detekci jednotlivých objektů. To je zapříčiněno tím, že objekty se budou posunovat buďto velice málo, potom nastane situace, že jednotlivé body se budou posouvat o velice podobné malé úseky (1 – 2 pixely) nebo naopak bude jejich pohyb velice rychlý a algoritmus výpočtu optického toku nebude schopen správně přiřadit odpovídající dvojice bodů.

Stiskem klávesy ‚t‘ dojde k aktivaci/deaktivaci vykreslení trajektorie jednotlivých bodů. Jak bylo zmíněno v textu, trajektorie je vyhodnocována ze všech nalezených bodů, tedy nejenom z bodů vyhodnocených z histogramu a identifikujících objekty.

Stiskem jakékoli jiné klávesy se program ukončí.

Změna kroku či zobrazení/skrytí trajektorie se projeví až při dalším posunu videa klávesou ‚n‘.

PŘÍLOHA D: OBSAH PŘILOŽENÉHO DVD

Obsah přiloženého DVD

složka `video` – obsahuje databázi osmi vzorových videí

složka `program` – obsahuje zdrojové kódy a projekt programu

soubory

`PathMonitoring.exe` – samotná aplikace

`PopisProgramu.pdf` – soubor s popisem programu. Jeho obsah je shodný s Přílohou C obsažené v textu bakalářské práce.

`PathMonitoring.pdf` – text bakalářské práce

`Metadata.pdf` – soubor metadat generovaný Informačním systémem VUT