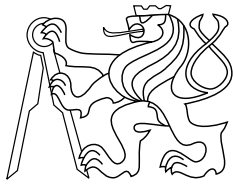




CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY

DIPLOMOV PRCE

Rozvržení stříhu ve vícekamerové nahrávce

Tomáš Losert

losert.tomas@gmail.com, doub@cmp.felk.cvut.cz

CTU-CMP-2008-04

7. února 2008

Lze získat na
<ftp://cmp.felk.cvut.cz/pub/cmp/articles/doubek/Losert-TR-2008-04.pdf>

Školitel: Petr Doubek

Práce byla podpořena grantem České akademie věd č.
1ET101210407, projekt MultiCam.

Research Reports of CMP, Czech Technical University in Prague, No. 4, 2008

Published by

Centrum strojového vnímání, Katedra kybernetiky
Fakulta elektrotechnická ČVUT
Technická 2, 166 27 Praha 6

fax: (02) 2435 7385, tel: (02) 2435 7637, www: <http://cmp.felk.cvut.cz>

České vysoké učení technické v Praze
Fakulta elektrotechnická



Diplomová práce

Rozvržení stříhu ve vícekamerové nahrávce

Tomáš Losert

Vedoucí práce: Mgr. Petr Doubek, Dr.sc.ETH

Studijní program: Elektrotechnika a informatika dobíhající magisterský

Obor: Výpočetní technika

leden 2008

Poděkování

Velmi rád bych chtěl vyslovit poděkování všem, kteří mi byli nápomocni při psaní této práce. Především Petru Doubkovi, vedoucímu mé diplomové práce, za trpělivé vedení a množství praktických rad.

Práce byla podpořena grantem České akademie věd č. 1ET101210407, projekt MultiCam.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 16.1. 2008

.....

Abstract

Desirable cut-composition has a significant impact on viewers perceiving of movie production. We have designed a heuristics which solves optimal cut-collocation of multi-camera recording in offline video-processing context. Based on the information about events in multi-camera system, acquired with the help of computer vision methods, the cut-collocation problem has been transformed into the shortest path in an oriented graph problem. Final optimal solution can be simply found using well-known algorithms. The main focus of the design was on respecting at least the fundamental techniques used in cinematography. The resulting method was tested on recordings from two different setups.

Abstrakt

Vhodná stříhová skladba má zásadní dopad na divákovo pochopení filmového díla. Navrhli jsme heuristiku, řešící optimální rozložení stříhu vícekamerové nahrávky v kontextu offline zpracování videa. Na základě informací o dění v multikamerovém systému, získaných pomocí metod strojového vnímání, byl problém stříhové skladby převeden na problém hledání nejkratší cesty v grafu. Výsledné optimální řešení je pak již možné lehce nalézt pomocí známých algoritmů. Během návrhu byl kladen důraz na respektování alespoň základních technik používaných v kinematografii. Správná funkce metody byla testována na nahrávkách z dvou různých multikamerových systémů.

Obsah

Seznam obrázků	xi
Seznam tabulek	xiii
1 Úvod	1
1.1 Motivace	1
1.2 Specifikace cíle	1
1.3 Předchozí práce	1
1.4 Struktura diplomové práce	3
2 Teorie a návrh řešení	5
2.1 Základy stříhové skladby a její vztah k rozmístění stříhu	5
2.1.1 Základní pravidla stříhové skladby	6
2.2 Záběr	6
2.2.1 Filmový záběr	6
2.2.2 Záběr v kontextu této práce	6
2.3 Výsledná sekvence	8
2.3.1 Požadavky na výslednou sekvenci	8
2.3.2 Kvalita výsledné sekvence (ohodnocení)	9
2.4 Převedení na již známé problémy a možné metody řešení	10
2.4.1 Hrubá síla	10
2.4.2 Genetický algoritmus	11
2.4.3 Hledání nejkratší cesty v grafu	11
3 Formální analýza výsledného řešení	13
3.1 Uvedení do kontextu	13
3.1.1 Počítání času	13
3.1.2 Vstup	13
3.1.3 Výstup	14
3.2 Předzpracování	15
3.2.1 Vytěžení informací ze vstupních dat	16
3.2.2 Korekce atributů	19
3.3 Vytvoření záběru	20
3.3.1 Hledání míst stříhu	20
3.3.2 Rozdělení jednotlivých sekvencí a vytvoření záběrů	21
3.3.3 Kombinace záběru a fragmentace nahrávky na pásy	22
3.4 Transformace na problém hledání cesty v grafu	22
3.4.1 Význam záběru a stříhu v grafu	22
3.4.2 Ohodnocení uzlů	24
3.4.3 Ohodnocení hran	24

3.5	Algoritmus hledání cesty	26
3.5.1	Vytvoření výsledné sekvence	28
4	Složitosti, ověření funkčnosti a experimenty	29
4.1	Paměťová a časová složitost	29
4.1.1	Fáze předzpracování	29
4.1.2	Konstrukce grafu a hledání nejkratší cesty	29
4.2	Výsledky dosažené na testovacích sekvencích	30
4.2.1	Posouzení závislosti výstupu na nastavovaných parametrech heuristiky .	30
4.2.2	Ukázka výsledků a shrnutí funkčnosti	33
5	Návrh Metody statistického učení	37
6	Závěr	39
6.1	Budoucí práce	40
A	Uživatelská / instalační příručka	43
A.1	Spuštění a popis práce s experimentální aplikací	43
B	Obsah přiloženého CD	45

Seznam obrázků

1.1	Ukázka z nahrávek jednotlivých kamer v MS	2
2.1	Snímky $A \div E$ - kamera č. 1, snímky $A' \div E'$ - kamera č. 2	6
2.2	Skupina snímků odpovídající záběru	7
2.3	Skupina snímků neodpovídající záběru	7
2.4	Kombinace záběrů rozdělením sekvence na pásy	10
2.5	Obecná struktura řešení - mřížka	10
2.6	Obecná struktura grafu reprezentující problém stříhové skladby	12
3.1	Posloupnost jednotlivých kroků heuristiky.	14
3.2	Atribut <i>angle</i> – směr pohybu hlavního objektu.	17
3.3	Atribut <i>border</i> – dotek hlavního objektu tří stran snímku.	17
3.4	Identifikace lidské tváře (fialová barva) ze seznamu objektů s barvou kůže (červená barva).	18
3.5	Atribut <i>movement</i> před a po provedení normalizace.	19
3.6	Korekce Atributu <i>angle</i> - Snímky zachycují část sekvence, kde dochází ke zmíněnému jevu. Červeně podtržená část znázorňuje střídání úhlu.	20
3.7	Příklad scény (o délce $m = 4$) ve výsledném řešení	23
3.8	Demonstrace struktury grafu.	23
3.9	Závislost <i>price_e</i> na délce (počtu snímků) scény.	25
3.10	Struktura grafu (pro přehlednost jsou zobrazeny pouze hrany z uzlů $v_{c,t}$ kde $c = 1$).	27
4.1	Závislost časové složitosti řešení na počtu pásů P pro metodu hrubé síly (čárkovaná čára) a pro metodu hledáním nejkratší cesty v grafu (plná čára).	31
4.2	Rozdělení nahrávek na záběry v závislosti na hodnotě minimálního pohybu.	32
4.3	Rozdělení části první testovací sekvence do záběrů.	34
4.4	Rozdělení části druhé testovací sekvence do záběrů.	36
5.1	Srovnání výsledku dosaženého použitím klasifikátoru SVM a námi navržené heuristiky (ad hoc).	38

Seznam tabulek

3.1	Hodnota atributu <i>border</i>	17
3.2	Hodnota atributu <i>head</i>	18
3.3	Vybrané atributy pro ohodnocení uzlů	24

Kapitola 1

Úvod

1.1 Motivace

Multikamerové systémy (dále jen MS) se stávají poslední dobou s vzrůstajícími nároky na bezpečnost stále více a více potřebnějšími. Příchodem levných digitálních kamer se staly MS dostupnými i pro oblasti, kde by se dříve pořízení složitějšího systému nevyplatilo.

Díky digitálnímu záznamu také vzniká velké množství dat, které je nutné sofistikovaněji zpracovávat a poskytnout tak divákovi přirozenější náhled na dění sledované pomocí MS. Sekvence složená ze záznamů několika kamer poskytuje, oproti sledování všech sekvencí zároveň, pozorovateli komplexnější pohled na zaznamenané dění a upoutá více pozornost na důležité momenty.

1.2 Specifikace cíle

Uvažujme sledovaný prostor, ve kterém se ve stejném čase může vyskytovat jen jeden objekt – osoba. Dále jsou v prostoru umístěny kamery, pokrývající místa u kterých očekáváme možnost nějaké akce objektu. Cílem této práce je navrhnout odpovídající heuristiku (algoritmus) pro offline zpracování nahrávek z těchto kamer. Heuristika, na základě údajů získaných z nahrávek viz obr. 1.1, nalezne posloupnost přepnutí kamer (stříhů) v čase, z které lze vygenerovat výslednou sekvenci tak, aby obsahovala všechny důležité momenty akce bez zmatení a ztráty pozornosti diváka. Jinými slovy je naším cílem vytvoření sekvence s maximální snahou dodržet alespoň základní kinematografické postupy[13][1]. Jak již bylo řečeno jedná se o offline zpracování, což je podstatný rozdíl od prací, které řeší problém stříhové skladby v reálném čase. Použitý termín stříhová skladba u nich může být zavádějící, protože se rozhodují pouze na základě informací z předchozích snímků, lze tedy pouze předpokládat dění a délku na dalším záběru. Výhoda offline zpracování spočívá ve využití informací z celého průběhu nahrávky a tím i možnost dosáhnout globální optimalizace řešení.

1.3 Předchozí práce

Problém rozvržení stříhu můžeme neformálně definovat jako “výběr správné kamery ve správném čase”. Důležitost stříhové skladby od počátků prvního filmu, díky větším a větším nárokům diváka, rychle vzrůstala. Nyní je stejně významnou částí filmové tvorby jako jiné obory kinematografie.

Význam stříhové skladby vzhledem k vnímání diváka do jisté míry zkoumal William W. Gaver [6]. Provedl experiment, kdy v jedné místnosti bylo umístěno několik kamer a v druhé místnosti měl účastník experimentu za úkol, na základě pohledů z kamer, nakreslit rozložení



Obrázek 1.1: Ukázka z nahrávek jednotlivých kamer v MS

nábytku v první místnosti. V situaci, kdy účastník volil pohledy manuálně dle svého uvážení bylo pro subjekt, díky menší koncentraci, obtížnější úkol splnit. Oproti tomu, při použití automatického přepínání pohledů respektující jejich vzájemné souvislosti, docházelo u subjektů k lepšímu prostorovému pochopení a vyšší koncentraci na plnění úkolu.

Otázkou zpracování a následné sumarizace velkého množství dat získaných z běžných web kamer se zabývá článek [10]. Řeší problematiku prezentace pouze zajímavých momentů celé nahrávky. Z velmi dlouhých záznamů (délky řádově mnoha hodin) vybírá na základě detekovaného pohybu souvislé skupiny snímků zachycujících zajímavé akce. V této práci bude navržená heuristika také vyhledávat části nahrávek odpovídající pohybu objektu. Nám však tyto informace poslouží jako součást metodiky ohodnocení použité při výběru zobrazované kamery.

Automatickému střihu videosekvencí se podrobně věnuje [12]. Popisuje základní terminologii střihové skladby, kterou se následně snaží aplikovat při vytváření modelu dané situace. Zaměřuje se speciálně na nahrávky pořízené z jednání či schůzí. Řeší problém, kdy a jakého účastníka schůze zabrat konkrétní kamerou. Oproti tomu, při řešení našeho problému, víme koho zabrat, ale rozhodujeme, kterou kamerou. V této práci se také dále snažíme při vytváření střihové skladby, dosáhnout její globální optimalizace pro celou nahrávku.

Naše práce navazuje na [2], kde je navržená metodika výběru kamery pro online zpracování videa. Při výběru kamery je kladen důraz na dodržení některých technik kinematografie. Výběr pohledu je zde rozdělen do dvou částí. První část, na základě detekce pohybu a výskytu objektů s barvou kůže, volí hodnotící kritérium aplikované druhou částí při výběru konkrétního pohledu. Z důvodu zpracování v reálném čase je volba kamery posuzována pro každý snímek. To je podstatná odlišnost od naší práce, která posuzuje celé skupiny snímků, tj. záběry.

1.4 Struktura diplomové práce

Pro lepší pochopení navržené heuristiky uvedeme čtenáře do základní teoretické myšlenky řešení. Popíšeme podrobněji problém stříhové skladby a blíže si ujasníme zastoupení filmařských postupů v celé heuristice. Na základě posouzení několik odlišných metod řešení vybereme jednu, které se budeme detailněji věnovat.

V další části (kapitola 3) formálně definujeme metodu vytvoření výsledné video sekvence. Dopodrobna popíšeme jednotlivé kroky řešení.

V kapitole 4 zhodnotíme poznatky získané experimentální kontrolou funkčnosti a odhadneme časovou a paměťovou složitost.

V poslední kapitole navrhne a z části prozkoumáme možnosti vylepšení heuristiky pomocí statistického učení.

Kapitola 2

Teorie a návrh řešení

V této kapitole budou probrány jednotlivé části navrhovaného algoritmu, které mají hlavní roli v řešení problematiky střihu.

2.1 Základy střihové skladby a její vztah k rozmístění střihu

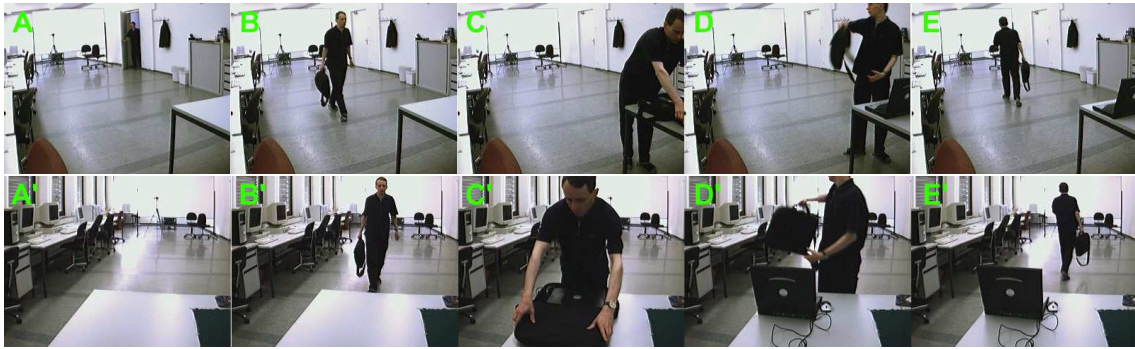
Střihová skladba, nebo-li montáž, je velmi důležitý nástroj autora k vyjádření jeho původní myšlenky a zajištění plynulého výkladu děje. Autor může správnou volbou střihové skladby ovládat, podle své libosti, čas a rytmus filmového díla, může vytvářet divákovi pocity a vjemy, zkrátka realizovat své umělecké představy. My se však musíme vzdát jak uměleckého cítění, rozumového uvažování střihače, tak i části filmového materiálu. Naše situace by se dala přirovnat k natáčení filmové scény více kamerami najednou, přičemž musíme respektovat statické nastavení kamer v MS. Proto se budeme soustředit na zachycení akce a dodržení kontinuity děje. Z tohoto hlediska lze na střihovou skladbu pohlížet jako na dva po sobě jdoucí kroky.

1. Nalezení nejvhodnějších míst střihu, rozdělení sekvencí na záběry.
2. Spojování a kombinace záběrů, potažmo celých skupin záběrů.

Podobně nahlížela na střihovou skladbu i polská teoretička W. Wertensteinová, která kdysi pronesla tvrzení, že “vlastní skladebná činnost spočívá ve vyhledání nejvhodnějšího místa střihu a že střihová skladba je dále slučování scén a sekvencí v ucelené filmové dílo” [13]. Pro nás je toto velmi zjednodušené pojetí střihu ideální, avšak pro popis práce střihače ve filmařské praxi je nedostačující.

Podívejme se na problém z opačné strany. Pomocí neformální definice míst, která nejsou vhodná ke střihu, popíšeme vlastnosti míst, která vhodná jsou. Myšlenku vysvětlíme v následujícím příkladu.

Příklad 2.1.1. *Představme si záznam dvou kamer 2.1 zachycující následující děj. Objekt vstupuje do zorného pole kamery – snímky $A \div C$ a $A' \div C'$, zastaví se a začne konat nějakou akci – snímky C a C' . Po dokončení této akce opět opouští zorné pole kamery – snímky $D \div E$ a $D' \div E'$. Pokud bychom jako místo střihu označili nějaký snímek sekevence v průběhu vstupu objektu (například snímek B) a nebo v průběhu konání akce, vedlo by to pravděpodobně k zhoršení čitelnosti záběru (záběr skončí dříve, než divák pochopí jeho smysl) a zhoršila by se možnost návaznosti na další záběry. Oproti tomu, představme si situaci, kdy místo střihu položíme mezi konec vstup objektu a začátek konání akce (snímek C a C'). Získáme záběr zachycující vstup objektu (snímky $A \div C$), na který můžeme vhodně navázat záběrem jiné kamery (pokud takový záběr máme) zachycující detail prováděné akce (snímek C').*



Obrázek 2.1: Snímky A÷E - kamera č. 1, snímky A'÷E' - kamera č. 2

Poznámka 1. Na obrázku 2.1 jsou zobrazené vždy jen první snímky jednotlivých částí sekvence.

Pravidlo 1: Filmové dílo, resp. stříhová skladba, je divákem lépe hodnoceno, pokud je sestaveno ze záběrů odpovídajících dílčím ucelným akcím.

Na základě tohoto pravidla budeme vyžadovat od algoritmu, aby jako záběry vyhledával konkrétní ucelené akce objektu.

2.1.1 Základní pravidla stříhové skladby

2.2 Záběr

Pro další postup řešení problému stříhové skladby je třeba přesněji definovat termín **záběr** a princip popisu akce, kterou záběr zachycuje.

2.2.1 Filmový záběr

Záběr z hlediska filmové praxe je vnímán jako plynulá část filmu mezi dvěma stříhy. Surový záběr pak od jedné klapky ke druhé. Obecně je záběr natáčen jednou kamerou a jeho délka je omezena pouze množstvím použitého materiálu pro jeho záznam. Hlavní hledisko hodnocení záběrů v hraném filmu je z pravidla ve vztahu k postavě a funkci prostředí. Kinematografie rozděluje záběry do mnoha kategorií, my si z nich uvedeme jen tři základní[1][13].

- VC – velký celek neboli dlouhý záběr (long shot). Obvykle zachycující celkový náhled na situaci (orientace v prostředí, umístění hlavní postavy, masové scény).
- PC – polo celek (medium shot). Hranice mezi PC a VC není zcela přesně definována. Obecně PC ukazuje hlavní děj, zachycuje celou postavu, prostředí hraje vedlejší roli.
- D – detail (close-up). Zachycuje konkrétní část akce, například detail tváře hlavní postavy nebo detail prováděné akce.

2.2.2 Záběr v kontextu této práce

Z pohledu navrhovaného řešení bude záběr vnímán jako stavební prvek výsledné sekvence. V tom je podstatná odlišnost od dřívějších přístupů používajících jako stavební prvek jeden snímek[2]. Jednalo se především o metody pracující v reálném čase, které rozhodují jen na základě informací získaných z předchozích snímků, a tedy je možné pouze odhadovat, jaká akce bude následovat v příštím záběru. Ve filmovém průmyslu se můžeme setkat s termíny jako je délka, čitelnost nebo tempo záběru. Určení těchto vlastností lze přibližně formulovat i pomocí

informací z MS. Na základě požadavků na výslednou sekvenci (části 2.1 a 2.3) lze konstatovat, že sekvence by měla být rozdělena tak, aby záběry co nejlépe odpovídaly jednotlivým akcím objektu. Uveďme si některé aspekty, které uvažujeme jako stěžejní pro nalezení ucelených akcí, respektive záběrů.

- výskyt objektu
- rozdíl polohy objektu (pohyb)
- směr pohybu objektu

Poznámka 2. *V této práci se omezujeme pouze na výše uvedené aspekty, do budoucna by se daly pro vyhledávání akce uvažovat i faktory jako velikost objektu, detekce výskytu hlavy a objektů s barvou kůže.*

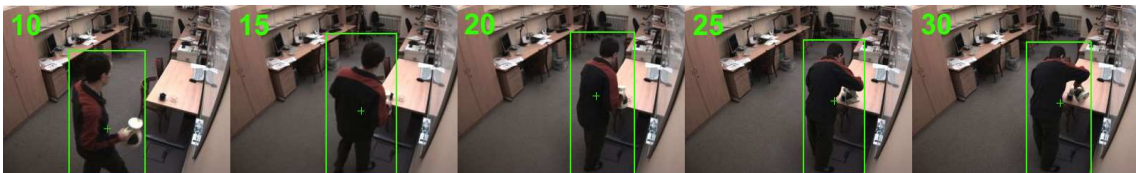
Pokusme se nyní definovat “správný” záběr v kontextu této práce.

Definice 2.2.1. *Záběr definujeme jako spojitou skupinu snímků sekvence jedné kamery, u kterých nedochází k prudké změně těchto vlastností objektu: výskyt objektu, rozdíl polohy objektu (pohyb), směr pohybu. Vyjádříme-li tyto vlastnosti číselně, musí být s tolerancí ϵ konstantní. Označme libovolný záběr $z_{c,i}$ kde c identifikátor kamery a i je pořadí záběru v sekvenci. Každý záběr má svůj počáteční snímek `start_frame`, koncový snímek `end_frame`*

Příklad 2.2.1. *Za záběr považujeme skupinu dvacetipěti snímků (obr. 2.2) jedné kamery, zachycující nepřetržitý pohyb objektu směrem 25° . Naopak skupina snímků zobrazena na 2.3 za záběr považovat nelze, protože na snímku č. 25 se již objekt přestal pohybovat. Tudiž v tomto místě začíná nový záběr.*



Obrázek 2.2: Skupina snímků odpovídající záběru



Obrázek 2.3: Skupina snímků neodpovídající záběru

Podle definice 2.2.1 jsou záběry tvořeny snímky s podobnými vlastnostmi, avšak pro další postup potřebujeme brát záběr jako homogenní prvek. Proto v jednom z dalších kroků sjednotíme vlastnosti snímků daného záběru spočtením aritmetického průměru té konkrétní vlastnosti přes všechny snímky záběru. Tím vznikne základ pro možnost ohodnocení záběrů, které je potřebné pro použití pokročilejších metod řešení.

2.3 Výsledná sekvence

Jak jsme se již zmínili v sekci 2.1, výsledná sestříhaná sekvence se skládá z po sobě jdoucích záběrů. Záběr podle definice v 2.2.1 je natočen určitou kamerou $c \in \{1..C\}$, má svůj počáteční a koncový snímek. Můžeme pak napsat následující definici.

Definice 2.3.1. *Sestříhaná video sekvence je posloupnost n záběrů $z_{c,i}$ kde c je identifikátor kamery a i je pořadí záběru. Pro každý záběr $z_{c,i}$ kde $i \in \{1..n-1\}$ a $c_i \in \{1..C\}$ musí platit*

$$z_{c_x,i}.end_frame + 1 = z_{c_x,i+1}.start_frame .$$

Výslednou video sekvenci lze také popsat posloupností dvojic (c, t) , které říkají, že na snímku t se má přepnout na kameru c .

2.3.1 Požadavky na výslednou sekvenci

Shrňme několik poznatků a předpokladů vybraných na základě pravidel kinematografie a na základě popsaného omezení pro MS. Tyto poznatky, popisující “správnost” výsledné sekvence, nám poslouží jako základ pro vytvoření metodiky ohodnocení výsledného řešení.

Zachycení akce

Následující aspekty je nutno brát v úvahu při zachycení akce. Jsou seřazeny podle své důležitosti, například, pokud nebude první aspekt splněn, nemůže pak ani jeden z následujících být splněn.

- Výskyt objektu – pokud se v záběru kamery nevyskytuje nebo se nepodaří rozpoznat hlavní objekt, tak nemáme žádnou informaci o dění před kamerou a tudíž “zajímavost” takového záběru můžeme považovat za minimální.
- Velikost objektu – hraje také poměrně důležitou úlohu. Přirozeně předpokládáme, že čím je objekt na záběru větší, tím detailnější pohled nám kamera poskytuje. Není však vhodné, když se objekt dotýká okrajů obrazu. Jestliže se dotýká jednoho či dvou souvisejících okrajů, je pravděpodobné, že objekt právě vstupuje na scénu a není zachycen celý. Dotýká-li se tří nebo všech čtyř okrajů, je nejspíše zabrán moc velký detail a obraz kamery nepokrývá celou akci.
- Rozdíl polohy objektu (pohyb) – předpokládáme, že objekt se při konání akce pohybuje. Pak tedy souřadnice těžiště objektu pro dva odlišné snímky budou rozdílné.
- Výskyt objektů s barvou kůže – vycházejme z faktu, že zachycujeme akce prováděné lidmi, pokud v záběru sledujeme pohyb objektů s barvou kůže (lidské paže, tvář), pravděpodobně sledujeme i prováděnou akci.
- Výskyt hlavy člověka – sledovaný objekt na záběrech kamer je člověk. Je tedy vhodné, aby na záběru byla vidět jeho hlava. Takový záběr bude zřejmě ve většině případů zajímavější než záběr, kde hlava vidět není.

Omezení rušivých vlivů na diváka

V předchozí části jsme se zabývali ukazateli popisujícími akce na jednotlivých záběrech. Zde se budeme věnovat především přechodům a návaznostem mezi záběry, tedy montáži neboli střihu. Popsaná pravidla neslouží k pevné definici, co může a nemůže nastat, ale spíše se snaží vyjádřit jaký střih je vhodný a jaký ne.

- Pohyb a jeho směr – v navazování záběrů zachycujících pohyb hraje velkou roli “pravidlo směru”. Pokud se v prvním záběru pohybuje objekt zleva doprava, působilo by na diváka velmi rušivě, kdyby v dalším záběru pohyb pokračoval zprava doleva. Představte si filmový záběr automobilové honičky, kdy ujíždějící vůz v jednom záběru jede zprava doleva, a pak v následujícím záběru, aniž by se stal nějaký zvrát (otočka, smyk atd.), jede vůz zleva doprava. Střih bude působit rušivě a dojde ke zmatení diváka.
- Změna polohy – je rušivé, pokud na sebe navazujeme dva záběry, kde na prvním se vyskytuje objekt na opačné straně obrazu než na druhém. Toto pravidlo v kombinaci s předchozím částečně popisuje tolik známé filmařské pravidlo osy.

Pravidlo osy: vytvoříme na scéně osu, například směr (prostorově) pohybu objektu. Aby nedošlo k prostorovému zmatení diváka, je vhodné volit záběry jen z kamer na jedné straně osy.

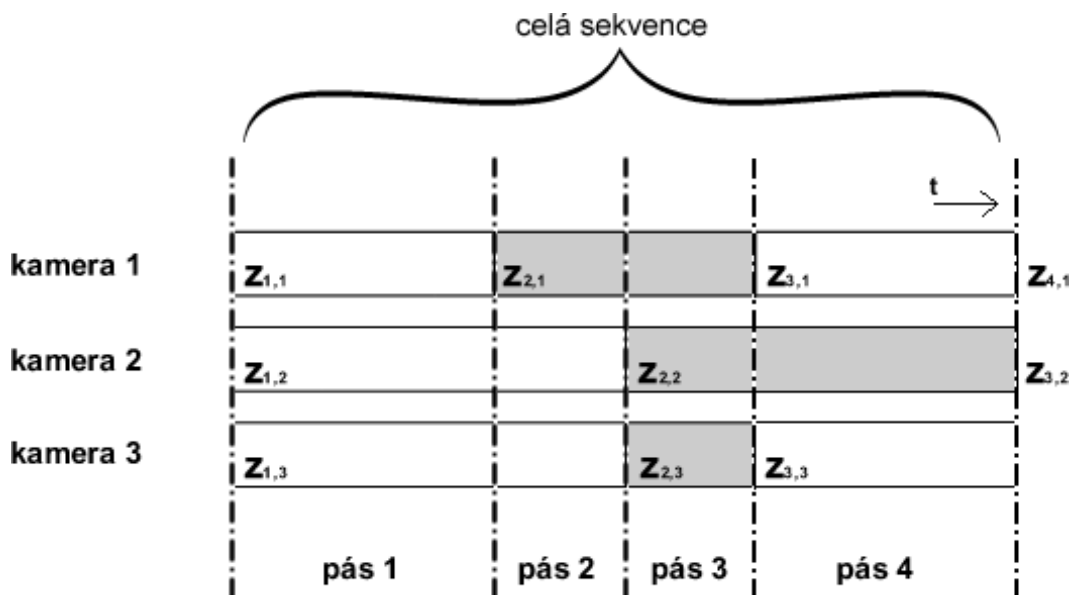
- Změna záběru – pro diváka je nepříjemné, pokud se při přechodu mezi záběry změní obsah záběru jen nepatrně. Například, pokud po záběru detailu následuje jen lehce odlišný detailní záběr.
- Délka záběru, ucelenost akce – záběr by měl být tak dlouhý, aby ho divák stihl “přečíst”, tj. pochopil co se na záběru děje [13]. V našem případě je toto velmi obtížné zajistit. Můžeme však vysledovat ucelené akce jako například, pohyb objektu jistým směrem, nebo naopak téměř nulový pohyb objektu. Důležité je také omezit ve výsledné sestřihané sekvenci výskyt příliš krátkých (jen několika snímkových) záběrů, způsobujících zmatení diváka. Stejně tomu je i naopak, pokud by byl záběr příliš dlouhý, v extrémním případě stejně dlouhý jako celá sekvence, mohlo by snáze dojít ke ztrátě pozornosti diváka.

Nalezené záběry v nahrávkách jednotlivých kamer nám určují místa v sekvencích, kde končí nebo začíná nějaká akce objektu. V těchto místech budeme rozhodovat, zda přepnout na jinou kameru, nebo pokračovat se stejnou kamerou. Komplikace nastane v situaci, kdy v jedné sekvenci začíná (resp. končí) záběr a v sekvenci pro jinou kameru je ve stejném čase záběr v průběhu. Jinými slovy, v ideálním případě by u sekvencí ze všech kamer začínaly a končily záběry ve stejném čase. Nebyl by pak problém v libovolném místě střihu přejít na libovolnou kameru. V reálné situaci se toto však stane jen velice zřídka. Tuto komplikaci odstraníme kombinací nalezených záběrů všech kamer a dalším rozdělením záběrů v místech, kde nastane takováto situace. Postup demonstruje obr. 2.4.

Časovou osu jsme řezy (znázorněny čerchovaně) rozdělili na P různě širokých pásů. Jeden konkrétní záběr musí být součástí alespoň jednoho pásu. Počet pásů a jejich délky budou pro další pokračování práce velmi důležitými údaji.

2.3.2 Kvalita výsledné sekvence (ohodnocení)

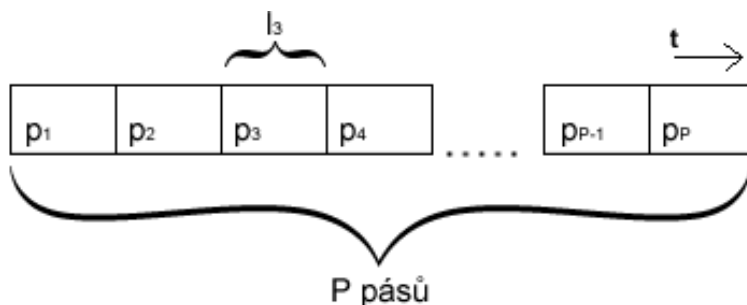
Z úvodní kapitoly 1.2 Specifikace cíle je patrné, že problém střihové skladby patří do kategorie optimalizačních problémů. Naším optimalizačním kritériem bude cena řešení vypočtená na základě požadavků výsledné sekvence (viz 2.3.1.). Výpočet ceny řešení bude podrobněji popsán v příští kapitole.



Obrázek 2.4: Kombinace záběrů rozdělením sekvence na pásy

2.4 Převedení na již známé problémy a možné metody řešení

Využití známých a ověřených postupů umožňuje usnadnění celého procesu řešení. Abychom mohli tyto metody použít, je nutné předem náš specifický problém transformovat na instanci problému, pro který je metoda řešení již známá.



Obrázek 2.5: Obecná struktura řešení - mřížka

Z definice 2.3.1 můžeme obecně popsat výslednou formu řešení. Máme-li v sekvenzi nalezené záběry, známe tedy počet pásů P a jejich délky l . Z těchto informací můžeme vytvořit mřížku 2.5 o rozměrech $1 \times P$, kde každé i -té políčko reprezentuje právě jeden pás p_i a může nabývat hodnot z množiny $\{1..C\}$, tj. hodnota identifikátoru vybrané kamery. Pak vyplněná mřížka reprezentuje právě jedno řešení.

2.4.1 Hrubá síla

Metoda hrubé síly spočívá v systematickém vygenerování všech řešení a výběru toho nejlepšího. Náhodné řešení vygenerujeme jako variaci s opakováním z C prvků třídy P , kde C je počet kamer a P počet pásů. Tedy počet všech možných řešení je C^P .

Budeme-li brát jako jednu iteraci kontrolu kvality řešení, je asymptotická časová složitost tohoto řešení $O(C^P)$. Jedná se o úplnou metodu s velkou časovou složitostí, která vždy nalezne

nejsprávnější řešení. Časová složitost by se dala vylepšit různými zlepšujícími praktikami jako například metodou Branch & Bound, přesto se však stává pro větší multikamerové systémy neúnosnou, díky její exponenciální závislosti na počtu pásů.

2.4.2 Genetický algoritmus

Velká část úspěšnosti genetického algoritmu[9] je zastoupena vhodnou volbou zakódování zkonstruovaného řešení do chromozomu. Od formy reprezentace řešení se odvíjí účinnost a možnost použití konkrétních operátorů mutace a křížení. Z úvodu této části jasně plyne základní možná reprezentace. Chromozóm jedince zastupuje výše popsaná mřížka (viz obr. 2.5), pak jedno políčko mřížky odpovídá jednomu genu v chromozómu. Gen reprezentuje zvolenou kameru pro daný pás (konkrétní políčko mřížky) a může nabývat $\{1..C\}$. Na takto definovaného jedince lze snadno použít základní jednobodový nebo dovoubodový operátor křížení.

Poznámka 3. *při bitové reprezentaci, kdy jedno políčko mřížky obsahuje bitově zapsaný identifikátor kamery, musíme respektovat místa, v kterých je možné křížení*

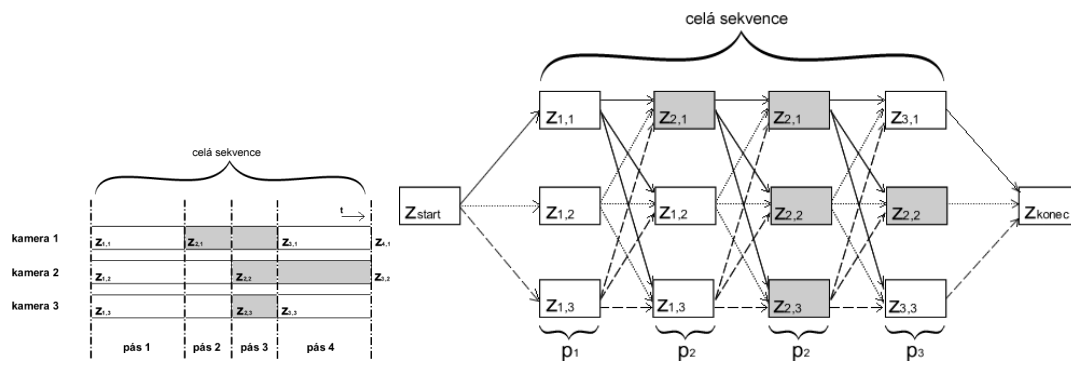
Metoda GA není úplná a není exaktní, tedy ne vždy nalezne nejlepší řešení. Počet vygenerovaných jedinců (řešení) potřebný k dosažení optima závisí na použití konkrétních strategií GA. I přesto by časová náročnost řešení mohla být díky většímu počtu iterací GA příliš velká. Na základě těchto důvodů jsme se rozhodli nepoužít GA pro řešení problému rozvržení stříhu.

2.4.3 Hledání nejkratší cesty v grafu

Tato práce se zabývá především touto metodou řešení, budeme se jí proto věnovat poněkud podrobněji. Popsali jsme si princip nalezení záběrů a postup jejich vzájemných kombinací, kterými získáme takzvané “pásy”. Pásy a záběry dávají základ obecné struktury grafové reprezentace našeho problému. Vytvoříme graf, kde jednotlivé uzly budou odpovídat záběrům rozdělených do částí s délkami odpovídajícími délkám konkrétních pásů. Výjimkou budou dva speciální uzly sloužící pro usnadnění použití metod hledání cesty, nezastupující žádný záběr, máme na mysli koncový a počáteční uzel. Hrany v takto postaveném grafu budou reprezentovat přechody mezi záběry, respektivě jednotlivá přepnutí kamery.

Na základě poznatků uvedených na začátku sekce 2.4 nám k reprezentaci postačí ohodnocený orientovaný graf obsahující jen dopředné hrany, tedy graf, kde uzly odpovídají topologickému uspořádání. Ohodnocení uzlů vyjadřuje popis akce na jemu odpovídající části záběru. Ohodnocení hran odpovídá ohodnocení vhodnosti přechodu mezi dvěma záběry (uzly) podle poznatků z části 2.3.1. Obecnou strukturu grafu demonstruje obr. 2.6. Je znázorněna jen zjednodušená obecná struktura grafu, skutečná struktura použitého grafu bude popsána v příští kapitole. Pro lepší představu zopakujeme na obr. 2.6 princip rozdělení sekvencí na pásy.

Náhodné řešení získáme jednoduše, lze si ho představit jako libovolnou cestu mezi počátečním a koncovým uzlem. Kvalitu řešení vyjádříme jako součet ohodnocení hran zvolené cesty. Pro nalezení optimální cesty v grafu použijeme již ověřené algoritmy. Jelikož uzly odpovídají topologickému uspořádání, musí se jednat o acyklický graf, a proto nám postačí Dijkstrův algoritmus nalezení nejkratší cesty [8].



Obrázek 2.6: Obecná struktura grafu reprezentující problém stříhové skladby

Kapitola 3

Formální analýza výsledného řešení

Uveďme si stručný přehled jednotlivých kroků navržené heuristiky v odpovídajícím pořadí viz obr. 3.1. Po načtení informací ze vstupních souborů, je pro další použití nutné vypočítat parametry lépe popisující dění v MS. Pomocí těchto údajů algoritmus vyhledá místa střihu a rozdělí sekvenci na jednotlivé záběry, z nichž pak vytvoří konkrétní uzly grafu. Vzájemným propojením uzlů hranami vznikne orientovaný graf. Před aplikací algoritmu hledání nejkratší cesty, je nutné uzly a hrany v grafu vhodně ohodnotit, tak, aby graf respektoval zachycené akce v MS. V posledním kroku vygenerujeme, na základě získané nejkratší cesty v grafu, výslednou sestříhanou videosekvenci.

3.1 Uvedení do kontextu

3.1.1 Počítání času

Obrazové informace z kamer jsou ve formě jednotlivých snímků. Označme libovolný snímek jako $f_{c,t} \in F$ kde F , je množina všech snímků sekvence, c identifikátor kamery $c \in \{1..C\}$ a t vyjadřuje čas, resp. pořadí snímku od začátku sekvence pro nahrávku konkrétní kamery. Snímková frekvence zpracovávaných nahrávek je $\approx 7.5Hz$, jeden snímek pak odpovídá $\approx \frac{1}{7.5} = 0,13s$.

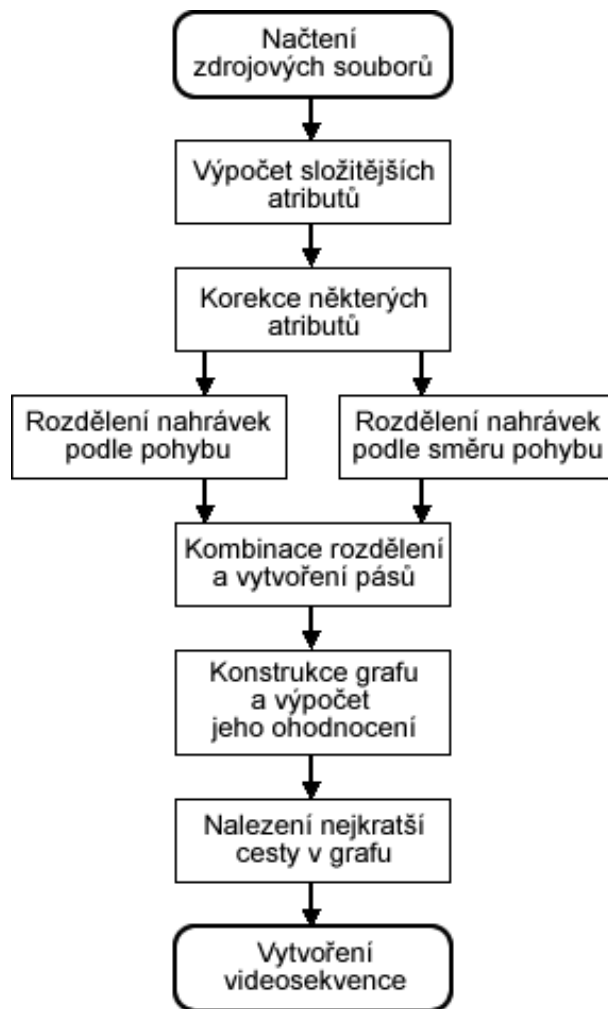
Jelikož máme C nahrávek pro stejný časový úsek (t_1 až t_2), je možné ve výsledné sekvenci zopakovat akci zachycenou například kamerou c_1 z pohledu kamery c_2 . Tento přístup skýtá možnost divákovi poskytnout více informací o dění v danou dobu. Oproti tomu ale porušení časové kontinuity působí na diváka velmi zmatečně, a proto se tomuto postupu vyhneme. Během celého postupu řešení budeme dodržovat následující pravidlo, popisující zmíněnou kontinuitu času v kontextu této práce.

Pravidlo 2: dosáhne-li hodnota “času” t konce právě zvoleného záběru $z_{c,i}$, pak je t místem, kde budeme volit přechod na jiný záběr. Označíme-li $z_{c,i+1}$ jako záběr navazující na $z_{c,i}$. Pak pro každý takovýto přechod musí platit

$$z_{c,i}.end_frame = t \wedge t + 1 = z_{c,i+1}.start_frame .$$

3.1.2 Vstup

Celá heuristika je postavena na informacích o jednotlivých snímcích uložených v souborech formátu xml. Soubory obsahují údaje o pozici a velikosti objektu získané metodou modelování pozadí obrazu a detekce pohybujiících se objektů [11, 7]. Xml dokumenty obsahují tyto pro nás důležité elementy.



Obrázek 3.1: Posloupnost jednotlivých kroků heuristiky.

- hlavní objekt (element `object`) – obsahuje informace o pozici a velikosti hlavního objektu.
- objekty s barvou kůže (element `skin`) – obsahuje informace o pozici a velikosti nalezených objektů s barvou kůže.
- detekovaná tvář (element `face`) – obsahuje informace o poloze nalezené tváře a o spolehlivosti detekce.

Poznámka 4. *Pověštinou pracujeme s rozlišením kamer 640×480 obrazových bodů. Obrazová data (jednotlivé snímky) slouží především k vytvoření výsledné video sekvence a kontrole její správnosti.*

3.1.3 Výstup

Výstupem heuristiky bude řešení popsané v části 2.4, tedy časová posloupnost přepnutí kamer. Na základě této posloupnosti lze již triviálním způsobem vygenerovat z obrazových dat výslednou video sekvenci.

3.2 Předzpracování

Z informací v xml datech vybereme a detailněji popíšeme atributy zajímavé pro naši práci. Data pro nahrávky jednotlivých kamer zpracováváme odděleně.

Atributy pro hlavní objekt (element `object`) :

- *size* – relativní velikost plochy objektu k velikosti celého snímku.
- *xc* a *yc* – souřadnice těžiště objektu s počátkem souřadnicového systému v levém horním rohu.
- *x₁*, *y₁* a *x₂*, *y₂* – souřadnice levého horního a pravého dolního rohu obdélníku (`bounding_box`) ohraničujícího hlavní objekt.

Atributy pro objekty s barvou kůže (element skin) :

- *size* – relativní velikost plochy objektu k velikosti celého snímku.
- *xc* a *yc* – souřadnice těžiště objektu s počátkem souřadnicového systému v levém horním rohu.

Atributy pro objekt lidské tváře (element face) :

- *xc* a *yc* – souřadnice těžiště objektu s počátkem souřadnicového systému v levém horním rohu.
- *conf* – parametr udává správnost detekce tváře[14]. Nabývá hodnot v intervalu $\langle -1, 1 \rangle$.

Většina těchto atributů není použita přímo, na jejich základě lze postavit složitější závislé atributy, díky kterým je možno lépe charakterizovat vlastnosti snímků a celých záběrů podle částí 2.2 a 2.3.

Implementační poznámka 1. Pro každý snímek vytvoříme objekt typu `MyFrameInfo` obsahující výše zmíněné informace, všechny tyto objekty (reprezentující jednotlivé snímky) pak spravuje v kolekci instance objektu třídy `FrameInfoList`.

3.2.1 Vytěžení informací ze vstupních dat

Požadavky na výslednou sekvenci popsané v kapitole 2 vyžadují komplexnější popis snímku, resp. skupin snímků, než nám umožňují atributy získané přímo z xml dat. Při výpočtu takových atributů pro konkrétní snímek budeme vycházet i z informací snímků okolních.

Atribut *movement* – vzdálenost Δ mezi souřadnicemi těžišť hlavního objektu snímku $f_{c,t}$ a snímku $f_{c,t+1}$

$$\Delta = \sqrt{|f_{c,t+1}.xc - f_{c,t}.xc|^2 + |f_{c,t+1}.yc - f_{c,t}.yc|^2} .$$

Poznámka 5. Pokud se na jednom ze snímků $f_{c,t}, f_{c,t+1}$ nevyskytuje hlavní objekt \Rightarrow souřadnice těžiště jsou nulové, bude i pohyb pro snímek $f_{c,t}$ nulový.

Hodnota tohoto atributu vyjadřuje, jak moc se změnila poloha hlavního objektu za čas $t = 1$. Přirozeně předpokládáme, že pokud došlo ke změně polohy objektu, muselo dojít k jeho pohybu.

Atribut *angle* – směr pohybu hlavního objektu počítáme jako úhel α , který svírá spojnice těžišť hlavních objektů na dvou rozdílných snímcích s horizontálním okrajem obrazu. U výpočtu úhlu, narozdíl od výpočtu atributů *movement*, je nutné pracovat se snímky navzájem v čase vzdálenějšími. V této práci volíme čas (vzdálenost) $t = 5$, snímky $f_{c,t}$ a $f_{c,t+5}$. Situaci popisuje obr. 3.2.

Poznámka 6. Pokud se na snímku $f_{c,t+5}$ nevyskytuje hlavní objekt \Rightarrow souřadnice těžiště jsou nulové, bude hodnota směru pohybu pro snímek $f_{c,t}$ rovna -1 .

Implementační poznámka 2. Atribut *angle* je reprezentován v radiánech hodnotou typu `double`. Může nabývat hodnot z rozsahu $\langle 0, 2\pi \rangle \cup \{-1\}$. Speciálním případem je hodnota -1 , která reprezentuje informaci, že směr pohybu není znám.

Obrázek 3.2: Atribut *angle* – směr pohybu hlavního objektu.

Atribut *border* – dotek objektu s okrajem snímku popisuje situaci, kdy objekt teprve vstupuje na scénu nebo už naopak vystupuje ze scény. Uvažujeme toleranci ϵ , s kterou posuzujeme, zda se hlavní objekt vyskytuje na kraji snímku, či nikoliv. Při posuzování uvažujeme rozměry ohraničujícího obdélníku (*bounding_box*). Situace je znázorněna na obr. 3.3

Obrázek 3.3: Atribut *border* – dotek hlavního objektu tří stran snímku.

Empiricky jsme nastavili hodnotu $\epsilon = 10$ (px). Atribut *border* může nabývat hodnot v rozsahu $< 0, 1 >$ podle tabulky 3.1. V budoucnu poslouží jen jako korekční koeficient atributu *size*.

Dotýká-li se objekt	<i>border</i>
jedné strany	0.75
dvou stran	0.50
tří stran	0.25
všech čtyř stran	0.0
nedotýká-li se žádné strany	1.0

Tabulka 3.1: Hodnota atributu *border*

Atribut *head* – vyjadřující, zda byla na snímku nalezena lidská tvář. Jeho hodnota je z rozsahu $< 0, 1 >$ a je definována tabulkou 3.2 na základě následujících dvou pravidel.

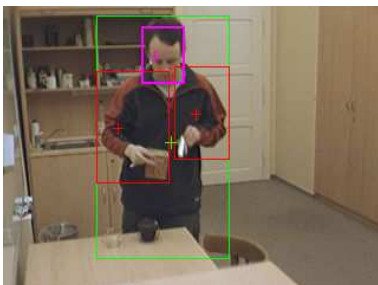
1. Pravidlo – jako objekt s barvou kůže mohou být rozpoznány lidské paže, nebo lidská tvář. Pokud na snímku byly tyto objekty rozpoznány, pokusíme se mezi nimi identifikovat objekt odpovídající lidské tváři. Takový objekt musí splňovat dvě podmínky:

- jeho velikost je menší než 10% velikosti hlavního objektu. Musí tedy platit

$$\text{objekt}_{skin.size} < 0.10 \cdot \text{objekt}_{main.size} .$$

kde $objekt_{skin}$ je testovaný objekt a $objekt_{main}$ je vztahovaný hlavní objekt.

- musí mít ze všech objektů s barvou kůže “nejvýše” položené těžiště a jeho souřadnice musí patřit do horní třetiny ohraničujícího obdelníku (viz obr. 3.4). Tato podmínka je založena na vlastnostech lidské anatomie. Lze konstatovat, že ve většině případů bude hlava nad pažemi.



Obrázek 3.4: Identifikace lidské tváře (fialová barva) ze seznamu objektů s barvou kůže (červená barva).

2. Pravidlo – se řídí parametrem $conf$ získaným metodou rozpoznávání lidské tváře [14]. Atribut $conf$ nabývá hodnot v rozsahu $< -1, 1 >$. Empiricky jsme stanovili minimální použitelnou hodnotu parametru $min(conf) = 0.9$, při které považujeme detekci tváře za spolehlivou.

Výslednou hodnotu atributu $head$ získáme podle tabulky 3.2

Splněné pravidlo	$head$
splněna obě pravidla	1.0
splněno jen jedno pravidlo (1 nebo 2)	0.5
obě pravidla nesplněna	0

Tabulka 3.2: Hodnota atributu $head$

Monotónnost atributů hlavního objektu $movement$ a $size$ – hodnoty mají pouze informativní charakter a dále je v práci neuvažujeme. Udávají, zda v okolí několika snímků (konkrétně ± 2 snímky) hodnota pohybu (popřípadě velikosti objektu) roste, klesá či je konstantní.

Monotónnost atributu $movement$ nese informaci, zda se pohyb hlavního objektu zrychluje či zpomaluje. Tato informace by mohla hrát důležitou roli pro určování tempa jednotlivých záběrů a tím i umožnit dokonalejší popis dění na záběru [13].

Z monotónnosti atributu $size$ lze přibližně určit, zda se hlavní objekt pohybuje směrem ke kameře, nebo naopak směrem od ní. Informace může být například velmi užitečná při složitější realizaci pravidla osy, viz část 2.3.1.

Implementační poznámka 3. *K výpočtu monotónnosti je použita metoda lineární aproximace `linear_aprox` implementovaná v třídě `MyMathFunction`. Výše popsané výpočty z velké části zajišťuje metoda `CalculateFrameInfo` z třídy `FrameInfoList`.*

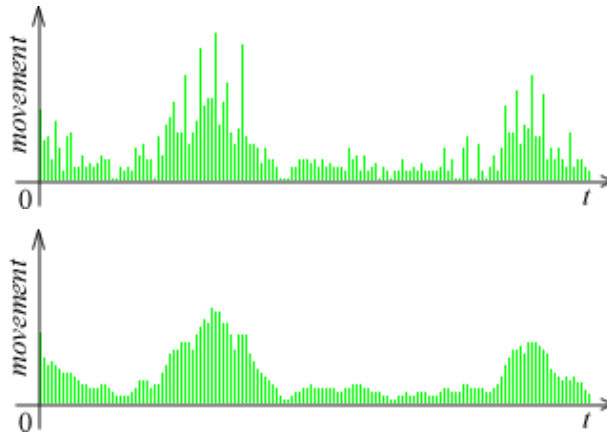
Tyto atributy již dostatečně charakterizují aspekty zmíněné v kapitole 2 a tudíž je pomocí nich možné jednotlivé snímky a záběry (skupiny snímků) vhodně ohodnotit. Systém ohodnocení se úzce vztahuje k použité metodě řešení, a proto bude podrobněji popsán až v sekci 3.4. Je ovšem nezbytné se již v této části zmínit, že výpočet ohodnocení vyžaduje znalost hodnot globálních maxim atributů $movement$ a $size$ pro všechny nahrávky sekvence.

3.2.2 Korekce atributů

Jak jsme již uvedli v části 2.2.2, budou pro hledání záběrů v sekvencích velmi důležité atributy *movement* a *angle*. Jejich získání jsme popsali v předchozí části. Abychom však mohli s těmito daty náležitě pracovat, je ještě nutné na nich provést drobné úpravy.

Normalizace atributu *movement* – provádíme za účelem odstranění extrémů. Hodnota atributu pro snímek $f_{c,t}$ je rovna aritmetickému průměru hodnot okolních snímků (viz obr. 3.5).

$$f_{c,t}.movement = \frac{1}{5} \sum_{t=-2}^{t=2} f_{c,t}.movement$$



Obrázek 3.5: Atribut *movement* před a po provedení normalizace.

Normalizace atributu *angle* – občas nastane situace, kdy na snímku není rozpoznán žádný objekt, ačkoli na okolních snímcích rozpoznán je. Pro tento snímek nemůžeme správně spočítat úhel pohybu ($f_{c,t}.angle = -1$). Výskyt takového snímku, v souvislé skupině snímků se správně vypočteným směrem pohybu, narušuje metodu vytváření záběrů popsanou v části 3.3. Proto, je potřebné upravit hodnotu atributu *angle* pro snímek $f_{c,t}$, jako průměr hodnot atributů okolních snímků. Bohužel, pro tento výpočet nelze použít klasický aritmetický průměr, protože ten by neodpovídal “průměrnému” směru pohybu objektu. Definujme vlastní funkci pro výpočet průměrné hodnoty dvou úhlů.

Definice 3.2.1. *Nechť $angle_avg(a_1, a_2)$ je funkce učující “průměrný” úhel. Tato funkce má dvě vstupní proměnné a_1 a a_2 , hodnoty úhlů v radiánech. Její funkční hodnota je určena následovně:*

Pro $|a_1 - a_2| < \pi$

$$angle_avg(a_1, a_2) = \left(\min(a_1, a_2) + \frac{|a_1 - a_2|}{2} \right) \bmod 2\pi$$

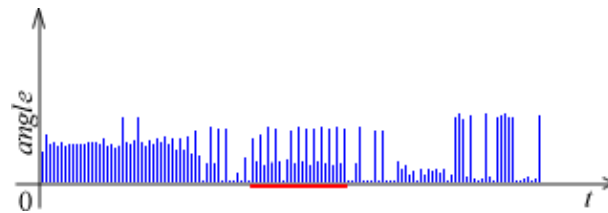
Pro $|a_1 - a_2| = \pi$

$$angle_avg(a_1, a_2) = \{-1\}$$

Pro $|a_1 - a_2| > \pi$

$$angle_avg(a_1, a_2) = \left(\max(a_1, a_2) + \frac{2\pi - |a_1 - a_2|}{2} \right) \bmod 2\pi$$

Korekce atributu *angle* – poměrně často nastane situace, kdy hlavní objekt koná nějakou akci, při které se jako celek nepohybuje. Přesto atribut *movement* není nulový. Souřadnice těžiště se cyklicky mění “tam a zpět”, tudíž, vypočtená hodnota atributu *angle* se neustále střídá se svým přibližným protiúhlem. Účelem této korekce je nastavit atribut $angle = -1$ u snímků, kde byl detekován tento stav s nastavenou přesností δ . Přesností δ myslíme toleranci v posuzování, který úhel je považován za protiúhel, a který ne (viz obr. 3.6).



Obrázek 3.6: Korekce Atributu *angle* - Snímky zachycují část sekvence, kde dochází ke zmíněnému jevu. Červeně podtržená část znázorňuje střídání úhlu.

Vzájemná korekce atributu *movement* a *angle* – vycházíme z logické úvahy, že pokud se hlavní objekt nepohybuje, nelze určit ani směr pohybu. Druhou podmínku lze odvodit analogicky. Upravíme tyto dva atributy následovně:

- pokud je $movement < \epsilon$ pak $angle = -1$ a $movement = 0$
- pokud je $angle = -1$ pak $movement = 0$

Hodnota konstanty ϵ je určena experimentálně a určuje minimální hodnotu atributu *movement*. Jestliže atribut *movement* klesne pod ϵ , bude pohyb objektu považován za nulový.

3.3 Vytvoření záběru

3.3.1 Hledání míst střihu

Pro hledání střihu uvažujeme jen informaci o pohybu a jeho směru. Jinými slovy, tyto dvě informace nám určují, kdy začíná a končí nějaká akce. Zajisté by se daly uvažovat i jiné informace (detekce lidské tváře, monotónost *size* atributu atd.), musíme však zvážit následné zvětšení počtu míst vhodných k střihu a tím rozdělení sekvence do více kratších záběrů způsobující nárůst počtu pásů. Speciálně pro MS s více kamerami by se pak, díky velké fragmentaci, námi navrhovaná metoda přiblížila zmiňovaným metodám uvažujícím jako stavební prvek jeden snímek.

Vycházejme z předpokladu, že záběr reprezentující akci bude mít uvažované vlastnosti (atributy *movement* a *angle*) přibližně konstantní. Naproti tomu, změna akce bude reprezentována náhlou změnou některého z těchto atributů. Nejprve nalezneme místa střihu, zvlášť pro atribut *movement* a zvlášť pro *angle*.

Rozdělení podle atributu *movement* – jako místo vhodné pro střih podle pohybu označíme snímek $f_{c,t}$ pokud

$$f_{c,t}.movement = 0 \wedge f_{c,t+1}.movement \neq 0$$

nebo naopak

$$f_{c,t}.movement \neq 0 \wedge f_{c,t+1}.movement = 0 .$$

Rozdělení podle atributu *angle* – jako místo vhodné pro střih podle směru pohybu označíme snímek $f_{c,t}$ pokud směr pohybu vůči snímku $f_{c,t+1}$ změní více než o $\frac{\pi}{4}$.

$$\min(|f_{c,t}.angle - f_{c,t+1}.angle|, 2\pi - |f_{c,t}.angle - f_{c,t+1}.angle|) \geq \frac{\pi}{4} .$$

Poznámka 7. Pokud jeden z posuzovaných atributů je roven -1, považujeme nerovnost za splněnou. Jsou-li obě hodnoty rovny -1, nerovnost nebude splněna.

Vyloučení příliš krátkých záběrů

Přesto, že na vstupní data jsou použity nejrůznější korekce, díky jejich různorodosti dochází k případům, kdy jsou některá dvě nalezená místa pro střih velmi blízko u sebe. To má za následek vznik velmi krátkých záběrů neodpovídajících požadavkům z části 2.3. Uveďme si pravidlo, kterým se řídíme během rozdělování nahrávky podle obou atributů.

Pravidlo 3: Jestliže uznáme za vhodné označit, jako místo pro střih snímek f_{c,t_1} a dále také snímek f_{c,t_2} kde $t_1 < t_2$ a $t_2 - t_1 < 4$ tak pro

- rozdělení podle atributu *movement* zjistíme podle toho, zda za snímkem f_{c,t_2} následuje střih alespoň $t = 4$ vzdálený, pokud ano, odstraníme místa střihu ze snímků f_{c,t_1} a f_{c,t_2}
- rozdělení podle atributu *angle* porovnáme, zda

$$\min(|f_{c,t_1}.angle - f_{c,t_2+1}.angle|, 2\pi - |f_{c,t_1}.angle - f_{c,t_2+1}.angle|) \geq \frac{\pi}{4} .$$

Pokud je nerovnost splněna, odstraníme místa střihu ze snímků f_{c,t_1} a f_{c,t_2} . Jestliže nerovnost není splněna a následuje-li za snímkem f_{c,t_2} střih alespoň $t = 4$ vzdálený. Následuje-li, odstraníme střih ze snímku f_{c,t_2} , v opačném případě ze snímku f_{c,t_1} . Jinak řečeno, pokud je krátký záběr obklopen záběry s navzájem podobnými atributy *angle*, tak tyto záběry vypustíme.

Implementační poznámka 4. Ve třídě `FrameInfoList` reprezentují tento krok metody `find_movement_shot()` a `find_angle_shot()`.

3.3.2 Rozdělení jednotlivých sekvencí a vytvoření záběrů

Z předchozího kroku jsme získali dvě množiny míst vhodných pro střih (pro *movement* a pro *angle*). Tyto dvě množiny nyní zkombinujeme a vytvoříme finální rozdělení nahrávky do záběrů pro každou kameru. Během kombinace se budeme snažit opět zamezit vzniku velmi krátkých záběrů.

Postupně projdeme všechny snímky dané nahrávky, pokud narazíme v čase t_1 na místo vhodné pro střih z hlediska atributu **A** (*movement* nebo *angle*), zkontrolujeme okolí tohoto snímku $\{f_{c,t_1+1}, \dots, f_{c,t_1+3}\}$, zda se v něm nevyskytuje v čase t_2 místo vhodné pro střih podle atributu opačného k **A**. Pokud ano, vypočteme průměr těchto dvou časů $t_3 = \lceil \frac{t_1+t_2}{2} \rceil$ a snímek f_{c,t_3} označíme jako místo střihu. Místa střihu v časech t_1 a t_2 z hlediska obou atributů odstraníme. Výsledkem tohoto postupu bude seznam koncových míst střihu, rozdělujících nahrávku na záběry.

Poznámka 8. *Tímto postupem zajistíme, že nová místa stříhu budou od sebe vzdálena minimálně $t = 3$, tzn. minimální délka takto vytvořeného záběru bude rovna třem snímkům.*

Sjednocení atributů pro záběry

Abychom mohli dále pracovat se záběry jako s ucelenými prvky, každý záběr ponese informaci o průměrné hodnotě atributů všech snímků, které obsahuje. Pro všechny atributy, kromě *angle*, nám postačí aritmetický průměr. Pro atribut *angle* budeme muset použít funkci *angle_avg()* z definice 3.2.1. Průměrný úhel pohybu získáme aplikací této funkce na první dva snímky záběru, dále pak na výsledek předchozího výpočtu a na další snímek. Takto pokračujeme až do konce daného záběru:

$$angle_avg(\dots angle_avg(angle_avg(f_{c,t}.angle, f_{c,t+1}.angle), f_{c,t+2}.angle), \dots) .$$

3.3.3 Kombinace záběru a fragmentace nahrávky na pásy

Nyní máme všechny nahrávky rozdělené do záběrů. Nalezené záběry v nahrávkách konkrétních kamer nemusí mít stejné délky, mohou začínat nebo končit v odlišném čase. Tento fakt komplikuje vytvoření grafu (viz část 2.3) za předpokladu dodržení časové kontinuity popsané v části 3.1.1. Komplikaci odstraníme kombinací míst stříhu a fragmentací celé sekvence do P pásů podle principu popsaného v kapitole 2. Úpravou se nám záběry jednotlivých sekvencí rozdělí na části odpovídající délkám konkrétních pásů. Tyto části záběrů budou ve vytvořeném grafu přímo zastoupeny uzly, proto můžeme pojmy “uzel” a “část záběru” považovat za totožné.

3.4 Transformace na problém hledání cesty v grafu

Vycházejme z teorie popsané v části 2.4.3. Zde si popíšeme podrobněji strukturu grafu a ohodnocení jeho hran a uzlů. Naší snahou je sestavit graf tak, aby respektoval požadavky na výslednou sekvenci a implicitně obsahoval informace o dění v MS. Na takto sestavený graf pak, za účelem nalezení optimální stříhové skladby, můžeme aplikovat bez větších úprav základní grafové algoritmy [8].

3.4.1 Význam záběru a stříhu v grafu

Kombinací jsme záběry rozdělili na části odpovídající uzlům $v \in V$ kde V je množina všech uzlů grafu. Z principu rozdělení vyplývá, že záběr může být součástí více pásů a musí být součástí alespoň jednoho. Uzel v grafu bude reprezentovat, jak jsme již řekli, část záběru odpovídající délce právě jednoho konkrétního pásu.

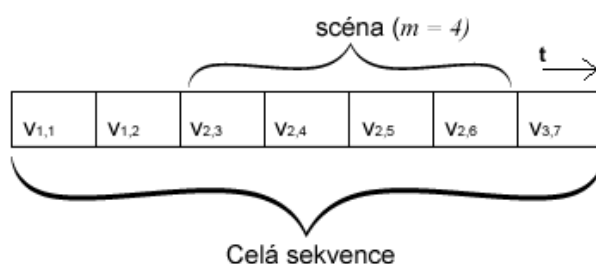
Uzly grafu pak lze uspořádat do obdelníkové mřížky o rozměrech $C \times P$ viz. obr. 3.8. Prozatím struktura odpovídá koncepci popsané v části 2.4.3, ovšem při návrhu hran se částečně od tohoto návrhu odchýlíme. V původním návrhu hrany mezi uzly odpovídaly přepnutím kamery, tudíž přechod po hraně reprezentoval stříh. Abychom mohli popsat přesný význam hrany v nově navrhované struktuře grafu, musíme si nejprve připomenout aspekt výsledné sekvence “délka záběru”, kterému jsme se doposud věnovali jen okrajově. Záběr by měl být tak dlouhý, aby divák stihl pochopit co se na záběru děje. Tato teze se netýká bezprostředně jen záběru podle definice 2.2.1, ale je důležité jí brát v úvahu i při hledání výsledného řešení. Pro použití v kontextu hledání stříhové skladby celé výsledné sekvence je potřeba definovat nový pojem “scéna” a termín “délka záběru” rozšířit na termín “délka scény”.

Definice 3.4.1. *Scéna s je skupina m po sobě jdoucích částí záběrů (uzlů) jedné kamery c vyskytující se ve výsledném řešení (viz obr. 3.7).*

$$s = \{v_{c,i}..v_{c,i+m}\}$$

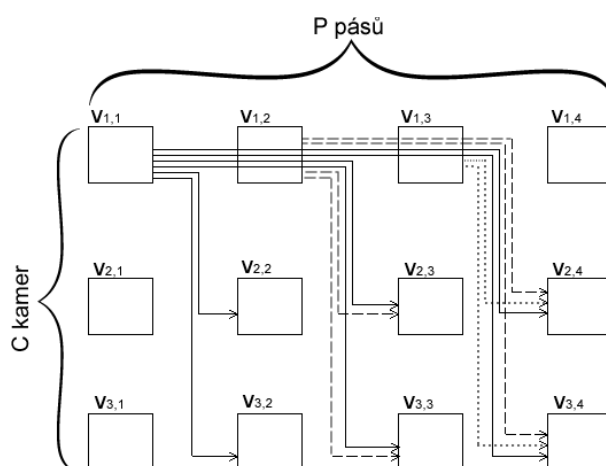
Pak délka scény je rovna součtu délek všech částí záběrů (uzlů), které scéna obsahuje.

$$|s| = \sum_{n=i}^{n=i+m} |v_{c,n}|$$



Obrázek 3.7: Příklad scény (o délce $m = 4$) ve výsledném řešení

Hlavním důvodem, proč zavádíme pojem scény, je možnost určení času, po který bude ve výsledné sekvenci v souvislém úseku vybrána jedna kamera. Otevře se nám tím možnost tyto úseky ohodnotit z hlediska jejich délky, a tím omezit příliš krátké nebo naopak příliš dlouhé scény. Nyní přejdeme k návrhu hran, ty budou v grafu odpovídat právě jednotlivým scénám. Konstrukci celého grafu vysvětlíme na obrázku obr. 3.8. Hrana vycházející z konkrétního uzlu reprezentuje scénu složenou z částí záběrů odpovídajících počátečnímu uzlu a všem dalším uzlům jimiž “prochází”. Ve skutečnosti hrana v grafu uzly procházet nemůže, zobrazení hran na obrázku 3.8 slouží pouze k lepší představě, že přechod po hraně reprezentuje scénu. Přechod ve výsledné cestě znamená zastoupení takto sestavené scény i ve výsledné sestříhané video sekvenci.



Obrázek 3.8: Demonstrace struktury grafu.

Poznámka 9. *Pro zjednodušení jsou na obrázku 3.8 uvedeny jen hrany pokrývající nahrávku kamery pro $c = 0$. Hrany vycházející z jednoho uzlu jsou znázorněny stejnou čarou.*

Z principu konstrukce scény je její minimální délka rovna délce alespoň jednoho pásu. Zprvu se zdá, že v grafu, narozdíl od návrhu v části 2.4.3, ubyly “přímé” hrany z uzlu $v_{c,i}$ do následujícího uzlu $v_{c,i+1}$. Jsou však pouze nahrazeny delšími hranami “procházejícími” uzlem $v_{c,i+1}$. Tato konstrukce umožňuje vnést do grafu informaci o ohodnocení na základě délek jednotlivých scén.

3.4.2 Ohodnocení uzlů

Princip ohodnocení uzlů vychází z požadavků na výslednou sekvenci a z atributů vypočtených v sekci 3.2.1. Každý uzel bude hodnocen na základě atributů popisujících část záběru, které daný záběr odpovídá. Budeme uvažovat atributy (viz tabulka 3.3) popisující vlastnosti záběru dle požadavků zachycení akce, popsanych v části 2.3.1.

Atribut	Minimální hodnota	Maximální hodnota
<i>movement</i>	0	max_movement
<i>size</i>	0	max_size
<i>border</i>	0	1
<i>head</i>	0	1

Tabulka 3.3: Vybrané atributy pro ohodnocení uzlů

Poznámka 10. Hodnoty *max_movement* a *max_size* získané v části 3.2.1 jsou globální maxima atributů *movement* a *size* pro všechny nahrávky dané sekvence.

Tyto atributy nabývají hodnot z odlišných rozsahů. Každý atribut by se pak na výsledném ohodnocení podílel jinou nedefinovanou vahou, takové ohodnocení by nedávalo, při hledání cesty v grafu, smysl. Je nutné atributy nejprve “normalizovat” tak, aby nabývaly hodnot v rozsahu $< 0, 1 >$. Úprava se týká jen *movement* a *size*, *border* a *head* již tuto podmínku implicitně splňují. Využijeme informace o globálních maximech těchto atributů získaných v části předzpracování dat 3.2. Normalizovaný atribut označíme čarou přidanou za názvem atributu původního. Předpokládejme, že $z_{c,t}$ je záběr, jehož částí odpovídá uzel $v_{c,t}$.

- $z_{c,t}.movement' = \frac{z_{c,t}.movement}{max_movement}$
- $z_{c,t}.size' = \frac{z_{c,t}.size}{max_size}$

Atributy *movement'*, *size'*, *border* a *head* jsou postaveny tak, aby z pohledu diváka byla příslušná část záběru (uzel) nejlépe hodnocena, právě když všechny tyto atributy budou rovny své maximální hodnotě. Tudíž, cena průchodu takovým uzlem musí být minimální. Výpočet hodnotící funkce $price.v()$ pro uzel $v_{c,t}$, kde $z_{c,t}$ definujeme takto:

$$price.v(v_{c,t}) = (1 - z_{c,t}.movement') + (1 - z_{c,t}.border \cdot z_{c,t}.size') + (1 - z_{c,t}.head) .$$

Poznámka 11. Atribut *border* jen v malém množství případů nabývá hodnot rozdílných od 1, proto jsme ho použili nepřímou jen jako korekční koeficient atributu *size*.

3.4.3 Ohodnocení hran

Hrany v grafu odpovídají konkrétním scénám složených z částí záběrů, resp. uzlů. Pro použití algoritmů pro hledání nejkratší cesty zastupuje ohodnocení hrany roli vzdálenosti mezi počátečním a koncovým uzlem hrany.

Funkce $price_e()$, pro cenu obecné hrany e procházející uzly $\{v_{c_1,i}..v_{c_1,i+m}\}$ a směřující do uzlu $v_{c_2,i+m+1}$, je složena z těchto částí ($price_e_1..price_e_4$):

- součet cen průchodů jednotlivými uzly $v \in \{v_{c,i}..v_{c,i+m}\}$, kterými hrana e prochází:

$$price_e_1 = \sum_{n=i+m}^{n=i} price_v(v_{c,n}) .$$

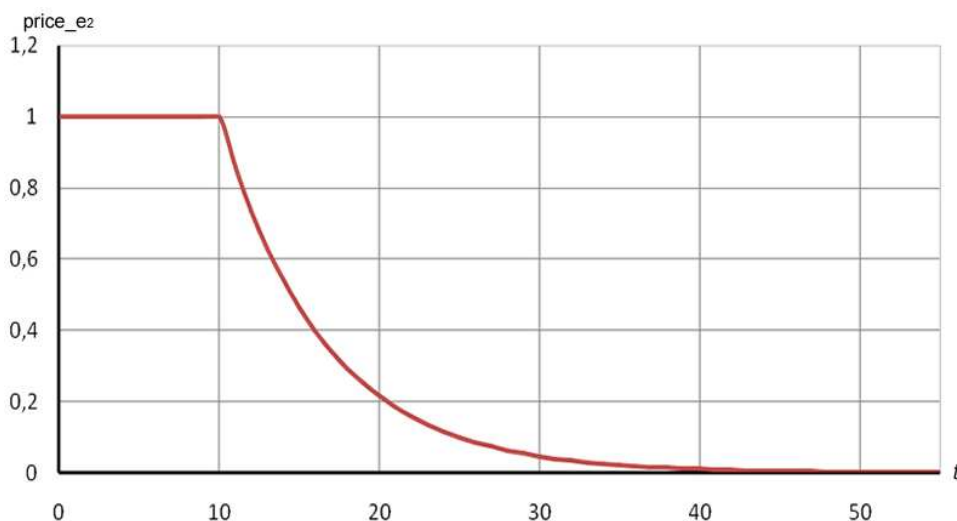
- pro ohodnocení na základě délky scény $|s|$, kterou hrana reprezentuje, jsme použili exponenciální závislost. Scény s délkou $|s| \leq 10$ snímků jsme znevýhodnili nastavením tohoto kritéria na maximální hodnotu ($price_e_2 = 1$). Pro MS s více kamerami, kde dochází k vytvoření většího počtu pásů, je třeba zvážit nastavení vyšší maxima a tím důrazněji omezit krátké scény. Pro scény delší než 10 snímků použijeme následující vztah:

$$price_e_2 = e^{-\lambda(|s|-10)} .$$

konstantu λ určíme na základě předpokladu, že pro stanovenou “ideální” délku scény $|s| = 25$, bude hodnota tohoto kritéria rovna 0.1.

$$\lambda = \frac{\ln(0.1)}{-15} \cong 0.154$$

Zavedením konstant pro minimální a optimální délku scény chceme dosáhnout omezení výskytu krátkých scén ve výsledném řešení. Závislost hodnotícího kritéria na délce scény je znázorněna na obrázku 3.4.3.



Obrázek 3.9: Závislost $price_e$ na délce (počtu snímků) scény.

- ohodnocení vhodnosti navázání poslední části (uzel $v_{c_1,i+m}$) scény, složené z m částí, na část následující (uzel $v_{c_2,i+m+1}$), tj. uzel, na který hrana směřuje. V této práci posuzujeme vhodnost navázání částí ze dvou hledisek, z pohledu směru pohybu a pozice objektu. Připomeňme si, pohybuje-li se objekt v jednom záběru směrem zprava doleva, bylo by rušivé, pohyboval-li by se v následujícím záběru opačně, zleva doprava. Na základě tohoto tvrzení logicky odvodíme vztah pro ohodnocení vhodnosti navázání dvou záběrů, respektive dvou částí záběrů:

$$price_{e_3} = \frac{\min(|v_{c,m}.angle - v_{c,m+1}.angle|, 2\pi - |v_{c,m}.angle - v_{c,m+1}.angle|)}{\pi} .$$

Poznámka 12. *Přirozeně nejhůře hodnocená situace nastane, pokud oba atributy angle budou navzájem protilehlé úhly. Rozdíl takových úhlů bude roven π .*

Dále posuzujeme vhodnost navázání dvou částí podle předpokladu, že pro diváka je nepříjemné, když se přepnutím kamery (stříhem) skokově změní pozice hlavního objektu. Vycházíme ze vzájemné vzdálenosti těžišť objektů na posuzovaných částech záběrů. Vzdálenost vztahujeme k největší možné, kterou je v krajním případě délka uhlopříčky snímku. Při pracovním rozlišení snímku 640×480 obrazových bodů je její hodnota 800px. Hodnotící kritérium spočteme jako poměr vzdáleností objektů a délky uhlopříčky:

$$price_{e_4} = \frac{\sqrt{|v_{c_1,i+m}.xc - v_{c_2,i+m+1}.xc|^2 + |v_{c_1,i+m}.yc - v_{c_2,i+m+1}.yc|^2}}{800} .$$

Cílem zavedení tohto kritéria, je potlačit ve výsledné sekvenci přechody, kde dochází k podstatné změně pozice hlavního objektu. Proto toto kritérium uplatníme, jen když dosáhne hodnoty nad 0.5, v opačném případě bude jeho hodnota nulová.

Funkční hodnota $price_{e}(e)$ se pak pro danou hranu e vypočte takto:

$$price_{e}(e) = \sum_{n=1}^{n=4} price_{e_n} .$$

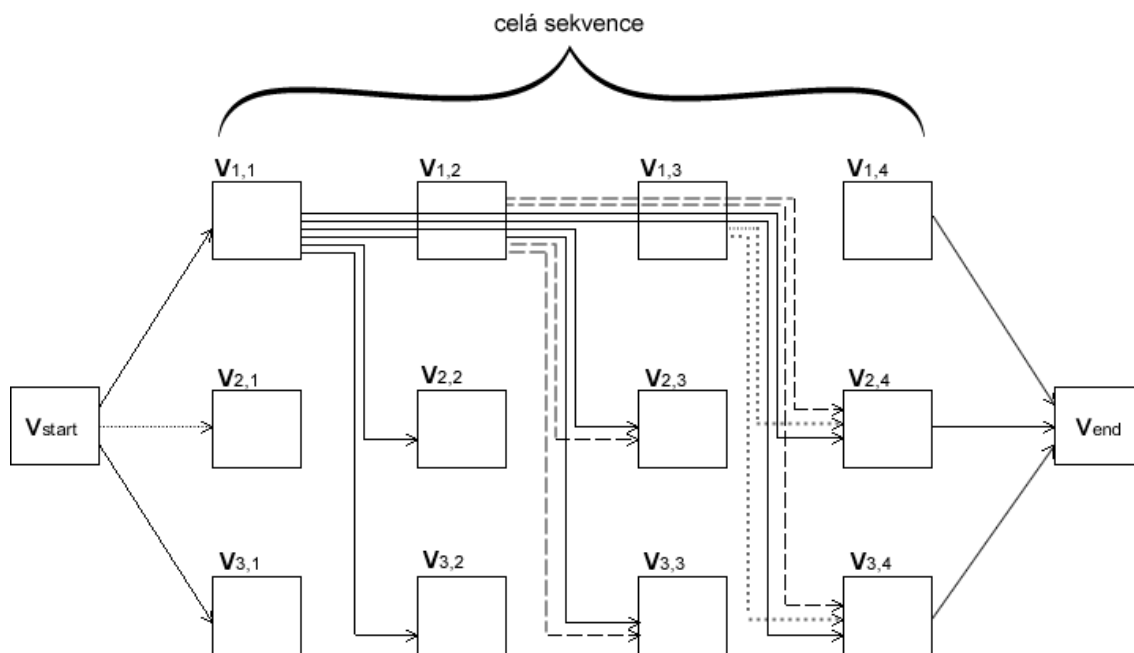
3.5 Algoritmus hledání cesty

V předcházejících částech jsme používali pro ohodnocení hran a uzlů grafu termín cena. Při teoretickém návrhu řešení jsme naopak doposud mluvili o hledání nejkratší cesty v grafu. Je potřeba si vzájemný význam pojmů “cena” a “vzdálenost” v rámci této práce ujasnit. Cena průchodu daného uzlu $price_v(v_{c,i})$ slouží jen jako dílčí část výpočtu ceny konkrétní hrany. Při hledání cesty v grafu hodnocení jednotlivých uzlů neuvažujeme, pracujeme pouze s cenou přechodu po konkrétních hranách grafu. Mluvíme-li o vzdálenosti mezi dvěma uzly, máme na mysli vzdálenost reprezentovanou cenou přechodu po hraně. Nejkratší cesta v grafu mezi uzly v_{c,i_1} a v_{c,i_2} bude zároveň i cesta s nejmenší cenou. Taková cesta složená z hran $\{e_1, e_2 \dots e_n\}$ v daném grafu G pak má součet ohodnocení všech hran $\sum_{i=1}^n price_{e}(e_n)$ minimální.

Základní úlohou při hledání cest v grafu, je nalezení všech nejkratších cest z daného uzlu[8]. Naším cílem je najít pouze jednu jedinou nejkratší cestu mezi počátečním a koncovým uzlem. To ale bohužel nepřináší usnadnění, protože není znám žádný algoritmus, který má asymptotickou složitost v nejhorsím případě lepší než algoritmus pro základní úlohu [8]. Algoritmů pro hledání cest v grafu existuje mnoho. Zvážíme-li, že graf sestavený podle částí 3.4 má jen nezáporné ohodnocení hran a jeho uzly díky časové kontinuitě (viz. část 3.1.1) odpovídají topologickému uspořádání, jedná se tedy o acyklický graf a plně nám postačí dobře známý Dijkstrův Algoritmus hledání cest.

Dijkstrův algoritmus hledání nejkratších cest

Než se pustíme do hledání nejkratší cesty metodou Dijkstrova algoritmu, musíme provést ještě poslední úpravu použitého grafu. Úprava spočívá v přidání počátečního uzlu (v_{start}), a koncového uzlu (v_{end}), a jejich propojení s příslušnými uzly. Graf znázorněn na obrázku 3.10 má již tyto uzly přidány.



Obrázek 3.10: Struktura grafu (pro přehlednost jsou zobrazeny pouze hrany z uzlů $v_{c,t}$ kde $c = 1$).

Poznámka 13. *Doplňené uzly (v_{start} a v_{end}) a hrany spojující tyto uzly se zbytkem grafu mají čistě pomocný charakter a nereprezentují žádnou část akce zachycené v MS. Cena přechodu po těchto hranách je rovna 0.*

Nyní si popíšeme princip Dijkstrova algoritmu pomocí pseudokódu. Zavedeme následující označení:

V ... jako množina všech uzlů v

E ... množina všech hran e

s ... počáteční uzel v_{start}

$d[v]$... vzdálenost (cena cesty) od s pro každý uzel v

$p[v]$... předchůdce uzlu v na doposud nejkratší nalezené cestě od s

$l(u, v)$... funkce vzdálenosti (ceny) pro přechodu mezi uzly u a v .

V našem případě $l(u, v) = price_e(e)$ kde e je hrana z uzlu u do v .

Dijkstra(V, s, l)

1	for každý uzel $v \in V$ do	Inicializace
2	$d[v] := \infty$	Nastavení vzdálenosti z v do s na nekonečno
3	$p[v] := \text{null}$	Předchůdce všech uzlů nastavíme na null
4	$d[s] := 0$	Vzdálenost z s do s je rovna 0
5	$N := V$	Do N všechny dosud nenavštívené uzly
6	while not empty(N) do	Dokud není N prázdné
7	$u := \text{vyber_min}(N)$	Vezměme uzel nejbližší k s
8	for všechny následníky u do	
9	if $d[u] + l(u, v) < d[v]$ then	Pokud je cesta přes hranu mezi u a v kratší
10	$d[v] := d[u] + l(u, v)$	Uprav vzdálenost uzlu v od s
11	$p[v] := u$	Nastav u jako předchůdce v

Výslednou cestu mezi počátečním $v_{start} \equiv s$ a koncovým uzlem v_{end} pak získáme postupným průchodem předchůdců, počínaje koncovým uzlem v_{end} .

1	$S := \{\}$	Výsledná množina uzlů cesty je prázdná
2	$u := v_{end}$	Vycházíme z koncového uzlu u
3	while $p[u] \neq \text{null}$ do	Dokud existuje předchůdce uzlu u
4	$S := u \cup S$	Přidejme uzel u na začátek U
5	$u := p[u]$	Pokročíme k předchůdci uzlu u

Odstraníme-li z této nejkratší cesty počáteční (v_{start}) a koncový (v_{end}) uzel, získáme cestu reprezentující z hlediska ceny optimální řešení stříhové skladby.

3.5.1 Vytvoření výsledné sekvence

Nalezené řešení je posloupnost uzlů, zastupujících části záběrů. Každá část má určenou svou délku, tj. kolik snímků ze záběru obsahuje. Sestříhanou videosekvenci získáme postupným průchodem všech uzlů, a přidáním vždy jen patřičných snímků do výsledku.

Implementační poznámka 5. Z hlediska implementace za sestříhanou sekvenci považujeme množinu snímků. Pro reálné použití by se z těchto snímků sestavil videosoubor (*.avi, *.mov atd..).

Kapitola 4

Složitosti, ověření funkčnosti a experimenty

4.1 Paměťová a časová složitost

Navržená heuristika by se dala z pohledu složitosti rozdělit na části přípravy dat (předzpracování, rozdělení sekvence, atd..) a na aplikaci pokročilejší známé metody řešení.

4.1.1 Fáze předzpracování

Část věnující se přípravě vstupních dat je složena z vzájemně navazujících “sekvenčních” kroků. Podrobný postup je popsán v první polovině 3. kapitoly. Následně použitá metoda řešení (např. hrubá síla, hledání nejkratší cesty), nemá na proces přípravy vliv, ten bude pro všechny metody stejný. V jednotlivých krocích přípravy opakovaně procházíme po snímcích celou sekvenci a vycházíme z informací vypočtených v kroku předchozím.

- **krok 1** – načtení zdrojových xml dokumentů do vnitřního formátu (třída `MyFrameInfo`)
- **krok 2** – výpočet složitějších atributů na základě načtených zdrojových informací
- **krok 3** – rozdělení nahrávek pro jednotlivé kamery z hlediska pohybu a směru pohybu
- **krok 4** – rozdělení sekvence na základě rozdělení jednotlivých nahrávek, vytvoření záběrů.

Asymptotický odhad časové složitosti této fáze je v závislosti na počtu zpracovávaných snímků ($n = \text{délka sekvence} \times \text{počet kamer}$) lineární $O(n)$. Stejně je tomu tak i pro paměťovou složitost. Tyto kroky by zajisté bylo možno alespoň částečně paralelizovat a tak zmenšit jejich časovou složitost. Pro přehlednost návrhu je ale vhodnější předpokládat jejich sekvenční provedení, které i z hlediska implementace značně usnadňuje jejich časté změny během vývoje heuristiky.

4.1.2 Konstrukce grafu a hledání nejkratší cesty

Počet hran a uzlů navrženého grafu

Máme $C \cdot P$ uzlů grafu (počáteční a koncový uzel neuvažujeme), z každého uzlu $v_{c,i}$ vedou hrany do všech následujících uzlů $v_{c',i'}$, kde $c \neq c'$ a $i < i'$. Přidáme-li počáteční a koncový uzel, přibude nám ještě $2C$ hran propojujících tyto uzly se zbytkem grafu. Celkový počet uzlů grafu je tedy:

$$|V| = C \cdot P + 2$$

a celkový počet hran v grafu:

$$|E| = 2C + C(C-1) \sum_{n=1}^{n=P} (P-n) = 2C + \frac{C(C-1)P(P-1)}{2}.$$

Paměťová složitost grafu

Počet kamer C vychází z nastavení konkrétního MS. Navíc lze předpokládat, že hodnota bude nabývat jednotek, maximálně desítek kamer. Proto paměťová složitost bude především závislá na délce zpracovávané sekvence. Nezávisí však přímo na počtu snímků sekvence, ale na počtu pásů P , do kterých je sekvence rozdělena. Odhad paměťové složitosti pro graf reprezentovaný pomocí seznamu sousedů je

$$O(|V| + 2|E|) \cong O(\max(|V|, |E|)) \cong O\left(\frac{C(C-1)P(P-1)}{2}\right) \cong O(C^2P^2).$$

Složitost Dijkstrova algoritmu

Uvažujeme-li hustý graf ($|E| \cong |V|^2$), kde E je množina všech hran grafu a V množina všech uzlů, uložených v prioritní frontě (výběr uzlu má lineární složitost), je asymptotická složitost Dijkstrova algoritmu [8]

$$O(|V|^2 + |E|) \cong O(|V|^2) \cong O(C^2P^2)$$

Obrázek 4.1 slouží pouze k vytvoření obecné představy srovnání časové složitosti řešení hrubou silou a hledáním nejkratší cesty v grafu. Složitost navržené heuristiky roste kvadraticky, zatímco časová složitost řešení pomocí hrubé síly exponenciálně. Závislost potvrzuje předpoklad, že metoda hrubé síly není ani pro krátké nahrávky reálně použitelná. Navržená heuristika vykazuje podstatně lepší časovou složitost.

Ovšem pro dlouhé nahrávky ($P > 40$) by bylo vhodné při konstrukci grafu vytvářet jen hrany do určité maximální délky. Úprava téměř neovlivní kvalitu řešení, jelikož v praxi nebudeme obecně považovat výskyt těchto dlouhých scén (např. desítky minut a více) za vhodný.

4.2 Výsledky dosažené na testovacích sekvencích

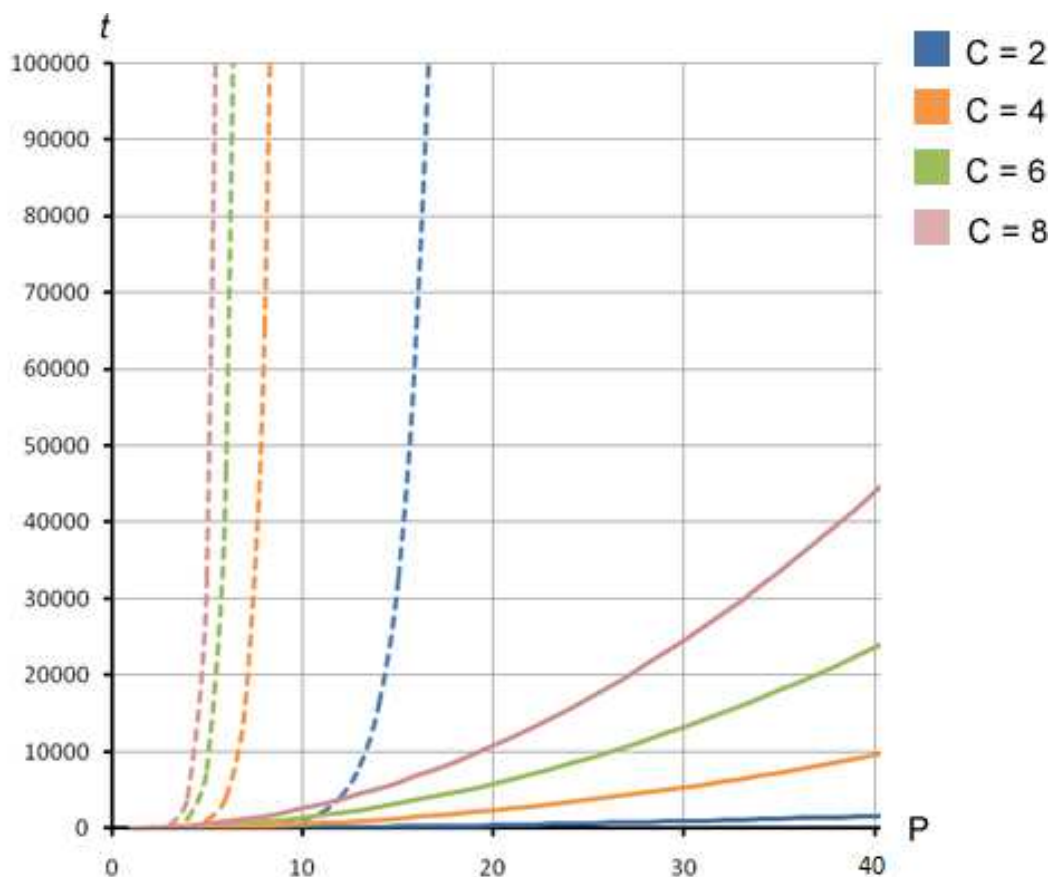
Během vývoje a testování jsme měli k dispozici dvě vzorové sekvence z různých MS. První obsahující nahrávky z 4 kamer a druhá obsahující nahrávky z 6 kamer. Délky obou nahrávek byly 900 snímků.

4.2.1 Posouzení závislosti výstupu na nastavovaných parametrech heuristiky

V klíčových částech přípravy dat bylo nutné rozhodovat na základě předem definovaných konstant. Jejich hodnoty pak více či méně ovlivňují výsledné řešení. V základu by se tyto parametry daly rozdělit podle použití. Uveďme si nyní jejich přehled a posouzení jejich vlivu na výsledné řešení. Konstanty jsou z důvodu přehlednosti uvedeny pod názvem použitým v implementaci.

Hledání záběru

Tyto konstanty významně ovlivňují vyhledávání vhodných míst stříhu a následného rozdělování nahrávek do záběrů. Jejich správným nastavením chceme dosáhnout toho, aby byla nalezena správná místa stříhu a sekvence byla rozdělena do záběrů, splňujících požadavky popsané v kapitole 2. Snažíme se také zabránit vzniku velmi krátkých (velká fragmentace sekvence), nebo naopak velmi dlouhých záběrů (málo stříhů, tudíž význam stříhové skladby přestává mít smysl).

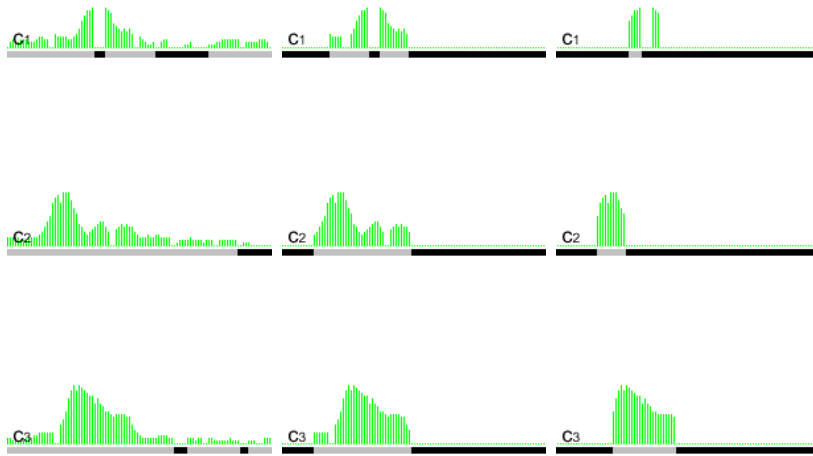


Obrázek 4.1: Závislost časové složitosti řešení na počtu pásů P pro metodu hrubé síly (čárkovaná čára) a pro metodu hledáním nejkratší cesty v grafu (plná čára).

MIN_MOVEMENT 1.23 minimální hodnota atributu *movement*.

Z principu výpočtu atributu je jeho hodnota nenulová i pro situaci, kdy se hlavní objekt nepohybuje. Proto pro $movement < MIN_MOVEMENT$ atribut vynulujeme.

Změnou této konstanty lze ovlivnit množství a délku nalezených záběrů. Pokud bude hodnota příliš vysoká, budou nalezeny jen záběry odpovídající velkému pohybu. Bude-li naopak hodnota nízká, algoritmus zareaguje i v místech, kde k pohybu hlavního objektu nedochází viz obr. 4.2. Na prvním obrázku (zleva) je nastavena velmi konstanta $MIN_MOVEMENT = 0.1$, další obrázek je pro použitou hodnotu 1.23 a poslední obrázek odpovídá nastavení příliš vysoké hodnoty 5. Černé a šedé pruhy značí jednotlivé záběry.



Obrázek 4.2: Rozdělení nahrávek na záběry v závislosti na hodnotě minimálního pohybu.

`DIFFERENT_ANGLE` $\frac{\pi}{4}$ tolerance při posuzování, zda se mezi dvěma snímky dostatečně změnil směr pohybu objektu.

Pokud je rozdíl atributu *angle* těchto dvou snímků větší než `DIFFERENT_ANGLE`, označíme první snímek jako místo střihu.

Konstantou ovlivňujeme citlivost na změnu směru pohybu hlavního objektu. Zvýšením této hodnoty snížíme citlivost a snížíme počet nalezených míst střihu.

`OPPOSITE_ANGLE` $\frac{\pi}{3}$ tolerance při posuzování, zda je úhel α protiúhlem β

Tato konstanta nemá přímý vliv na vyhledávání míst střihu. Je ale důležitá pro správnou korekci atributu *angle*. Tím nepřímým způsobem zabráňuje špatnému určení míst střihu podle směru pohybu hlavního objektu.

`MIN_SHOT_LEN` 3 minimální počet snímků záběru

Z důvodu různorodosti vstupních dat dochází k chybným určení míst střihu, a tím i k vytvoření velmi krátkých záběrů o délce jen několika snímků. Z reálného hlediska nemůže takový záběr zachycovat nějakou ucelenou akci. Navíc větší počet záběrů způsobuje fragmentaci sekvence do více pásů a tedy nárůst časové i paměťové složitosti. Proto se snažíme vznik těchto záběrů zavedením konstanty `MIN_SHOT_LEN` omezit.

Ohodnocení hran a uzlů

Správné řešení je přímo závislé na ohodnocení hran a uzlů v grafu. Naší snahou je, aby princip ohodnocení co nejlépe respektoval dění zachycené v MS. Uvedené konstanty jsou nutnou součástí některých hodnotících pravidel.

BORDER_MARGIN 10 (px) tolerance při posuzování, zda se objekt dotýká okraje snímku.

Hodnota je částečně závislá na MS, a to konkrétně na použitém rozlišení kamer. Hodnota **BORDER_MARGIN** = 10 (px) byla stanovena experimentálně pro rozlišení 640×480. Slouží pouze k znevýhodnění záběrů zachycujících objekt ve fázi vstupu, nebo opouštění scény. Proto ji nemá smysl měnit jinak, než v poměru ke konkrétnímu rozlišení kamer.

OPTIMAL_SCENE_LEN 25 ideální délka (počet snímků) scény

Tato konstanta pomáhá definovat výstup hodnotící funkce délku scény. Její hodnota určuje minimální délku scén, které již nebudou při hodnocení diskriminovány. V souvislosti s následující konstantou značně ovlivňuje délky scén zastoupených v řešení.

MIM_SCENE_LEN 10 minimální délka (počet snímků) scény

Jedná se o opak předchozí konstanty **OPTIMAL_SCENE_LEN**. Určuje velikost scény, která ještě bude v hodnocení maximálně diskriminována. Z požadavků na výslednou sekvenci můžeme usoudit, že tato hodnota by neměla být příliš nízká. Avšak nastavení této konstanty závisí na charakteru zaznamenávaných akcí v MS.

MAX_HEAD_SIZE 0.1 maximální hodnota poměru velikosti lidské tváře k velikosti hlavního objektu

Hodnota **MAX_HEAD_SIZE** = 0.1 byla určena experimentálně. Jelikož se jedná o poměrovou konstantu založenou na předpokladu, že poměr velikostí tváře a celého člověka bude vždy stejný, není do jisté míry závislá na nastavení MS.

MIN_HEAD_CONF 0.9 minimální hodnota parametru *conf*, které musí dosáhnout, aby byla detekce lidské tváře považována za správnou

Na testovaných nahrávkách, z důvodů malého rozlišení a nízké kvality obrazu, byla detekce velmi nespolehlivá. Experimentálně jsme nastavili hodnotu *conf* = 0.9.

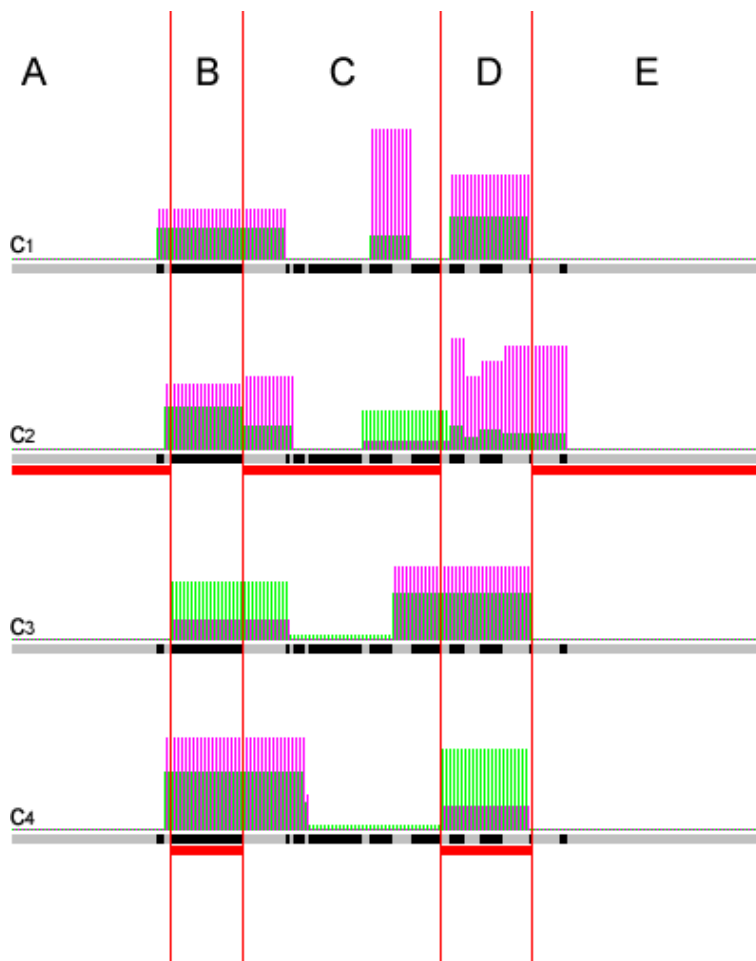
Poznámka 14. *U konstant **MIN_SHOT_LEN**, **MIN_SCENE_LEN** a **OPTIMAL_SCENE_LEN** je jejich hodnota také závislá na snímkové frekvenci použitých nahrávek.*

4.2.2 Ukázka výsledků a shrnutí funkčnosti

Dosažení optimální stříhové skladby se z hlediska působení na diváka velmi špatně hodnotí. Bylo by třeba provést experiment za pomoci odborné asistence a většího počtu diváků, kteří by výsledné sekvence hodnotili. Jediný pro nás dostupný způsob je hodnotit výslednou video sekvenci na základě požadavků vznesených v části 2.3.1.

Na základě obrazových informací můžeme poměrně přesně ověřit správnou funkci fáze, kdy algoritmus hledá místa stříhu a vytváří záběry. Místa stříhu nalezené algoritmem skutečně, s přesností několika snímků, odpovídají začátkům a koncům ucelených akcí. Popišme si nyní dvě vzorové situace vybrané z výsledku heruistiky.

Na obrázku 4.3 jsou znázorněny časové osy a histogramy atributů *movement* (zeleně) a *angle* (fialově) pro část nahrávek první testovací sekvence. Černé a šedé části odpovídají délkám jednotlivých pásů. Červeně jsou pak vyznačeny části zastoupené ve výsledném řešení (časové úseky **A** až **E**). Dále je zobrazen pro každý úsek snímek z kamery zastoupené ve výsledném řešení. V úseku **A** subjekt provádí konkrétní činnost (největší plocha objektu, detekce hlavy), v úseku **B** došlo k pohybu objektu a následně v úseku **C** opět provádění konkrétní činnosti. Část **D** zachycuje návrat objektu do původní polohy (směr a míra pohybu), dále je zvolena opět kamera č. 2 zachycující detail akce.

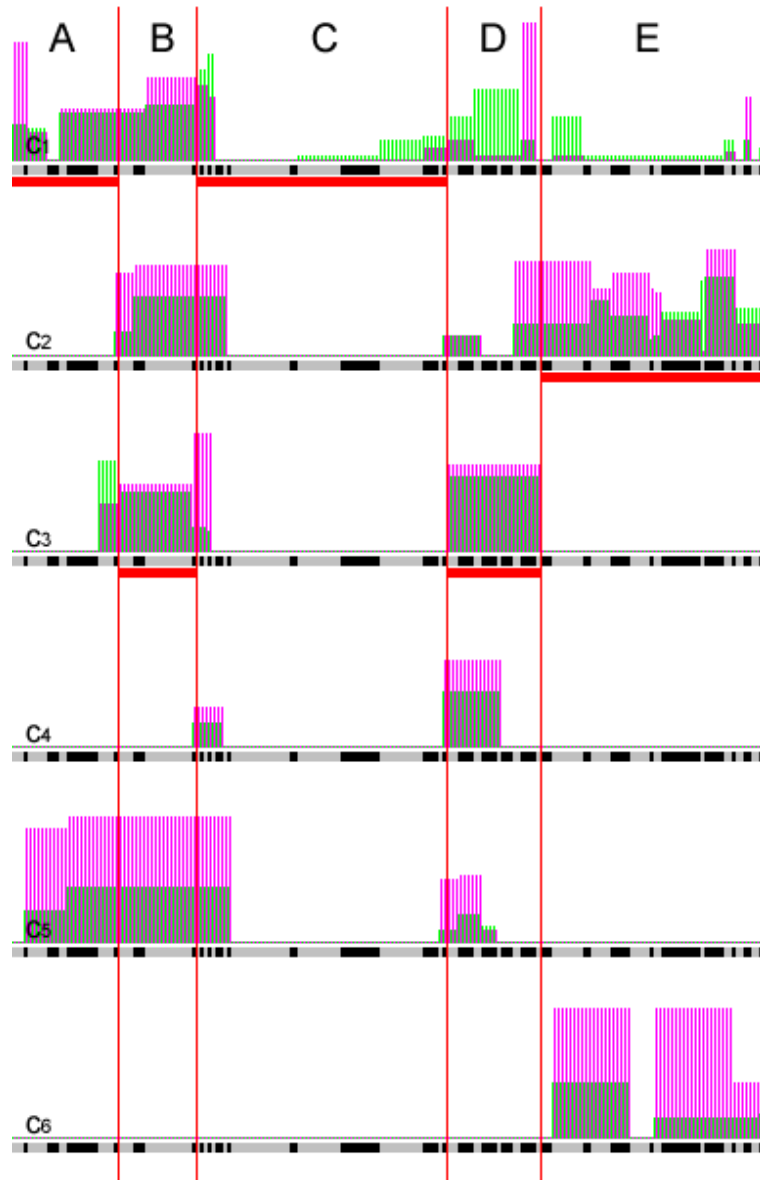


Obrázek 4.3: Rozdělení části první testovací sekvence do záběrů.

Na dalším obrázku 4.4 je zachycena situace z druhé testovací sekvence, kdy se objekt začne pohybovat směrem do středu místnosti (úsek **A** – červená šipka znázorňuje směr pohybu). Během pohybu dojde k přepnutí na kameru č. 3 (úsek **B**), která zabírá větší detail pohybu (důvodem je stejný směr pohybu, větší objekt). Po zastavení objektu je zvolena kamera č. 1. (úsek **C** – detekce hlavy, velikost objektu) podávající obecnější přehled. Jakmile se objekt znovu

rozpohybuje, vybere se opět kamera č. 3, která lépe zachycuje otočení a začátek pohybu. Poté je přepnuto na pohled kamery č. 2 zabírající detail budoucí prováděné činnosti.

Nejdůležitější dopad na celkovou funkci algoritmu má správné nastavení výše popsaných konstant. Některé konstanty (např. `MIN_MOVEMENT`, `DIFERENT_ANGLE`) jsou závislé na konkrétní konfiguraci MS, některé jsou závislé jen částečně (např. `MAX_HEAD_SIZE`, `BORDER_MARGIN`) a nějaké závislé nejsou (např. `OPPOSITE_ANGLE`). Jejich hodnoty byly určeny na základě experimentů s první testovací sekvencí. Se stejným nastavením pak bylo pomocí heuristiky nalezeno přiměřeně kvalitní řešení i pro druhou testovací sekvenci. Na základě tohoto zjištění lze konstatovat, že navržená heuristika je do jisté míry univerzální. Jedná se ovšem pouze o předpoklad na základě jednoho experimentu. K ověření tohoto předpokladu by bylo třeba experimentů dalších. I přes částečnou univerzálnost heuristiky, by k dosažení kvalitních výsledků pro konkrétní MS bylo vhodné nejprve na testovací nahrávce nastavením těchto konstant algoritmus kalibrovat. Na kvalitu finálního řešení má zajisté podstatný vliv také správnost zdrojových dat vzhledem k reálné situaci zachycené v MS.



Obrázek 4.4: Rozdělení části druhé testovací sekvence do záběrů.

Kapitola 5

Návrh Metody statistického učení

Cílem této práce bylo kromě návrhu heuristiky řešící optimální rozvržení střihu, také prozkoumat možnost uplatnění metod statistického učení pro ohodnocení vhodnosti záběru. Podobně jako [3] učí stroj hodnotit krásu lidského obličeje podle vzorových příkladů hodnocení dodaných lidskými učiteli, mohli bychom se pokusit naučit hodnocení kvality záběru a střihu. Přirozeně by se dalo předpokládat využití těchto metod při řešení problému celkové střihové skladby. Na základě trénovacích dat bychom se snažili naučit stroj, jaký střih v kontextu dané situace vhodný je a jaký ne.

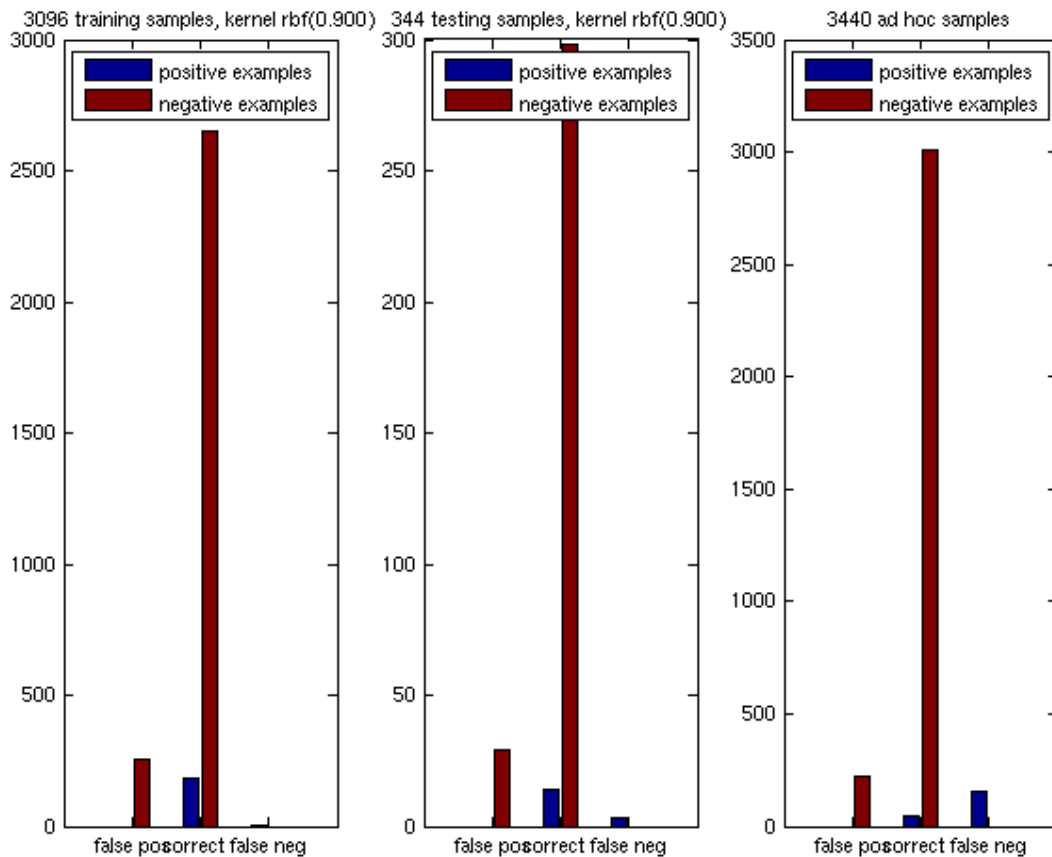
Jednotlivým vzorkům by pak odpovídala konkrétní přepnutí z jedné kamery na druhou. Aby takový vzorek měl smysl, musí obsahovat informace popisující vlastnosti navazujících částí obou nahrávek. Pro získání trénovacích dat resp. pozitivních vzorků, by bylo třeba specialisty z oboru. Ten by na základě svých znalostí určil pozitivní vzorky, tedy vytvořil správnou střihovou skladbu. Získat však potřebná trénovací data a vhodně popsat vlastnosti navazujících částí je velice komplikované a v rámci této práce nerealizovatelné.

Proto jsme se rozhodli aplikovat tyto metody pouze na dílčí část heuristiky určující místa střihu pro jednotlivé nahrávky (dle kritéria pohybu, viz část 3.3). Naší snahou bylo prozkoumat vylepšení principu vyhledávání ucelených akcí a tím částečné odstranění závislosti na nastavené konstantě `MIN_MOVEMENT`.

V tomto případě jednomu vzorku odpovídá jeden snímek, pozitivní vzorek by pak znamenal správný střih oddělující jednotlivé ucelené akce. Stejně jako v předchozím případě je potřebné vhodně popsat okolí snímku v kontextu řešeného úkolu. Nyní je situace o mnoho snazší. Rozhodli jsme se každý vzorek doplnit informacemi o porůměrné hodnotě atributu *movement* několika předcházejících a následujících snímků (deseti, pěti a tří). Jelikož pozitivní vzorek nyní reprezentuje místo oddělující akce pouze z hlediska pohybu sledovaného objektu, nebylo již k jejich získání potřeba odborné asistence. Do jisté míry je sporné, zda daná akce začíná na konkrétním snímku anebo na snímku okolním, proto jsme určovali pozitivní vzorky s přesností \pm jednoho snímku.

Na základě snímků konkrétní nahrávky byla pro každou kameru vytvořena množina vzorků. K jejich klasifikaci jsme používali SVM klasifikátor s radial basis function kernelem [5]. Pro jeho naučení byla použita trénovací sada o velikosti $\frac{9}{10}$ veškerých vzorků ze všech kamer. Do testovací sady byl vždy zahrnut každý desátý vzorek. Data byla zpracovávána ve formátu `arf` pomocí programu Matlab a Weka[15].

Dříve než rozebereme výsledky na obrázku 5.1, je nutné ujasnit význam některých pojmů. Pozitivní vzorky (*positive examples*) spadající do skupiny **false positives** znamenají nalezení střihu v místě, kde by střih být neměl. Naopak tomu vzorky spadající do skupiny **false negatives** respektují situaci, kdy by správně měl být klasifikován střih, avšak klasifikátor tento vzorek označil za negativní (zahozený střih). Z principu dalšího zpracování nalezených míst střihu v kontextu heuristiky není zvýšený počet **false positives** vzorků tak závažný.



Obrázek 5.1: Srovnání výsledku dosaženého použitím klasifikátoru SVM a námi navržené heuristiky (ad hoc).

Protože takto špatně nalezený stříh se ve výsledném řešení nemusí vůbec objevit. Pokud je ale místo správného stříhu klasifikováno negativně (skupina **false negatives**), znamená to, že při hledání výsledného řešení nebude vůbec umožněno provést na tomto místě přechod (stříh). Proto je pro nás větší počet vzorků ve skupině **false negatives** závažnějším problémem, kterému se budeme snažit vyhnout.

První dva histogramy na obrázku 5.1 ukazují srovnání námi vytvořených (pozitivních) vzorků s výsledky klasifikátoru. Z důvodu částečného překrytí pozitivních a negativních vzorků nelze dosáhnout jejich stoprocentního oddělení. Avšak alespoň z hlediska snížení počtu vzorků spadajících do množiny **false negatives** je možné se přiblížit kvalitnímu řešení, tj. jen malá část “zahozených stříhů”.

Jelikož navržená heuristika (ad hoc) nerespektuje možnost nalezení dvou velmi blízkých stříhů, je při uvedeném principu vyhodnocení oproti klasifikátoru částečně v nevýhodě. Klasifikátor se neohlíží na to jak byly klasifikovány okolní snímky a případné sousedící stříhy nepotlačí. Proto při srovnání dochází u heuristiky k vyššímu počtu **false negatives** a naopak nižšímu počtu **false positives**.

Tyto popsané výsledky ovšem nejsou obecným řešením této podúlohy. Pouze poukazují na to, že má smysl se této metodě řešení do budoucna věnovat.

Kapitola 6

Závěr

V této práci jsme navrhli a detailně popsali jednotlivé kroky heuristiky řešící optimální rozvržení střihu z hlediska offline zpracování videa. Jedná se o proces pracující s informacemi, které jsme získali z projektu MultiCam. Vycházeli jsme z reálné situace, kdy filmové dílo je složeno ze za sebou poskládaných záběrů. V tomto aspektu se tato práce částečně odlišuje od prací podobných.

Naším záměrem bylo transformovat problém rozložení střihové skladby tak, aby na něj bylo možné použít již ověřené metody řešení, konkrétně hledání nejkratší cesty v grafu. Vhodnou konstrukcí grafu a výpočtem ohodnocujícím jeho hrany jsme se snažili dosáhnout, aby graf implicitně respektoval dění zachycené v MS. K tomu jsme využili zdrojových dat a zobecnění některých technik a pravidel používaných v kinematografii.

Správná funkce je do jisté míry spojena s počátečním nastavením konfiguračních konstant heuristiky. Jejich hodnoty, uvedené v kapitole 4, byly nastaveny na základě experimentů na první testovací sekvenci tak, aby bylo dosaženo kvalitního řešení. Při jejich stejném nastavení heuristika vykazovala srovnatelně správnou funkci i pro druhou sekvenci. Můžeme tedy konstatovat, že navržená metoda řešení je alespoň částečně nezávislá na vlastnostech instance problému. Ovšem, pro dosažení lepších výsledků, by bylo potřebné heuristiku pro konkrétní problém “kalibrovat”.

Z jednotlivých kroků heuristiky plynou její problémová místa. Prvním je správné nalezení záběrů v jednotlivých nahrávkách, které, dle vznesených požadavků, mají odpovídat uceleným akcím sledovaného objektu. Jelikož máme jen omezené množství zdrojových informací, nelze stoprocentně zaručit nalezení takovýchto záběrů. Přesto k chybnému určení míst střihu (vzhledem k požadavkům) docházelo jen v malém počtu případů. Chybné určení střihu ale vždy nemusí znamenat ještě přímé ovlivnění kvality výsledného řešení, nýbrž jen přidá možnost výskytu takového střihu ve výsledné video sekvenci.

Druhý problém je výpočet vhodného ohodnocení grafu tak, aby co nejlépe popisoval znamenání děj. Ohodnocení je složeno z několika dílčích částí. V této práci jsme nastavili stejné váhy všech těchto částí. Pro konkrétní zadání by se, ovšem, nastavením vah daly diskriminovat, či naopak zvýhodnit potřebné scény.

Při návrhu byl kladen důraz především na požadavek kvality, paměťová a časová náročnost byl požadavek druhotný. Přesto, pro hlavní část heuristiky (konstrukce grafu a hledání nejkratší cesty), vykazuje paměťová i časová složitost pouze kvadratickou závislost na počtu pásů P . Počet pásů je podstatně menší než délka sekvence ve snímcích, tedy i obě složitosti budou nižší. To je jedna z výhod použití záběrů jako základních stavebních prvků namísto snímků.

6.1 Budoucí práce

Za předpokladu dostupnosti více zdrojových informací, jako je například 3D pozice objektu, rozpoznání druh lidské akce[4, 16], atd., by bylo možné vytvořit graf, jenž obsahuje komplexnější popis zaznamenané situace v MS. Otevřela by se nám možnost respektovat během řešení složitější filmařské postupy a tím zvýšit jeho výslednou kvalitu.

Další možnost vylepšení heuristiky je použití metod statistického učení. Z důvodu popsaných v kapitole 5 jsme upustili od aplikace těchto metod na celkové řešení. Naopak vhodné se zdá být užití metod učení při hledání míst stříhu, tedy klasifikaci snímků.

Literatura

- [1] D. Arijon. *Grammar of the Film Language*. Silman-James Press, September 1991 (originally published 1976).
- [2] P. Doubek, I. Geys, T. Svoboda, and L. Van Gool. Cinematographic rules applied to a camera network. In *Omnivis2004*, pages 17–29, 2004.
- [3] Y. Eisenthal, G. Dror, and E. Ruppin. Facial attractiveness: Beauty and the machine. *Neural Computation*, 18(1):119–142, 2006.
- [4] P. F. Felzenschwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.
- [5] V. Franc. Statistical Pattern Recognition Toolbox for Matlab. <http://cmp.felk.cvut.cz/cmp/software/stprtool>.
- [6] W. W. Gaver, A. Sellen, C. Heath, and P. Luff. One is not enough: multiple views in a media space. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1993.
- [7] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.
- [8] J. Kolář. *Teoretická informatika*. ČVUT, Prosinec 2000.
- [9] V. Mařík, O. Štěpánková, and J. Lažanský. *Umělá inteligence (3)*. Academia, 2001.
- [10] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. In *ICCV 2007*, October 2007.
- [11] D. Roth, P. Doubek, and L. Van Gool. Bayesian pixel classification for human tracking. In *7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005)*, volume 2, pages 78–83, 2005.
- [12] S. Sumec. Multi camera automatic video editing. In *Proceedings of ICCVG 2004*, pages 935–945. Kluwer Verlag, 2004.
- [13] J. Valušiak. *Základy střihové skladby*. Akademie muzických umění v Praze, 2005.
- [14] J. Šochman and J. Matas. Waldboost – learning for time constrained sequential detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 150–157, Los Alamitos, USA, June 2005. IEEE Computer Society.
- [15] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.

- [16] A. Yilmaz and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *ICCV*, pages 150–157. IEEE Computer Society, 2005.

Dodatek A

Uživatelská / instalační příručka

Během návrhu byla vyvinuta aplikace implementující postupy zmíněné v této práci. Byl použit programovací jazyk Java a vývojové prostředí NetBeans IDE 5.5.1. Zdrojové kódy a přeložené soubory *.class a *.jar jsou umístěny v projektovém adresáři `implement` na přiloženém CD.

A.1 Spuštění a popis práce s experimentální aplikací

Ke spuštění aplikace je potřeba běžná verze Java Runtime Environment (JRE). Aplikace se pak spustí z příkazové řádky následovně:

```
java -jar expapp.jar
```

Základní rozhraní aplikace (okno `Monitor`) slouží pro navigaci a zobrazení snímků nahrávky. Je zde možné nastavit některé konfigurační konstanty heuristiky a spustit zpracování vybrané sekvence. Po provedení výpočtů jsou zobrazeny přehledy rozdělení nahrávek na záběry (okno `Časová osa`) a nalezené řešení stříhové skladby (okno `Řešení`). Je možné spustit přehrávání sestříhané sekvence. Podrobnější informace naleznete v kořenovém adresáři přiloženého CD uložené v souboru `readme.txt`.

Dodatek B

Obsah příloženého CD

Příložené CD obsahuje tyto soubory a adresáře:

implement– adresář obsahující materiály k implementaci experimentální aplikace (zdrojové kódy, spustitelné soubory, vstupní data).

text– adresář obsahující text této práce ve formátu **pdf** a **ps**.

latex– adresář obsahující text této práce ve zdrojovém formátu **LATEX**.

output– adresář obsahuje sestříhané videosekvence ve formátu **avi** vytvořené na základě výstupu aplikace.

readme.txt– soubor obsahující podrobnější informace o implementaci a pokyny pro obsluhu experimentální aplikace.