

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

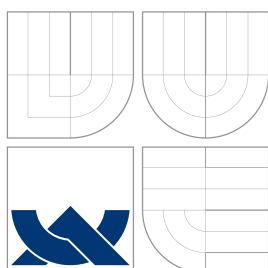
DETEKCE A SLEDOVÁNÍ MALÝCH POHYBUJÍCÍCH SE OBJEKTŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

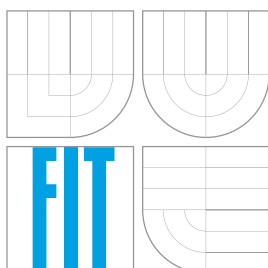
AUTOR PRÁCE
AUTHOR

DAVID ZÁDĚRA

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE A SLEDOVÁNÍ MALÝCH POHYBUJÍCÍCH SE OBJEKTŮ

DETECTION AND TRACKING OF SMALL MOVING OBJECTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DAVID ZÁDĚRA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ROMAN JURÁNEK

BRNO 2011

Abstrakt

Tato bakalářská práce shrnuje a popisuje metody vedoucí k detekci a sledování pohybujících se objektů ve videu. Z detekčních metod jsou v práci zmíněny metody odečítání pozadí, segmentace a detekční metody využívající klasifikátorů. Ze sledovacích metod jsou zde představeny metody Kalmanův filtr, metoda TLD a částicový filtr, který je popsán detailněji v samostatné kapitole. Částicový filtr představuje sledovací metodu, která provádí výpočet odhadu nového stavu objektu na základě sady částic. Tyto částice jsou definovány jejich polohou a ohodnocením. Pro každý následující stav jsou tyto hodnoty iterativně aktualizovány novým ohodnocením a převzorkováním. Ze získaných znalostí a poznatků byl vytvořen program, který demonstruje činnost částicového filtru. Tento program je součástí přiloženého DVD, kde jsou i ukázky a výsledky testů.

Abstract

This thesis summarizes and describes the methods for detecting and tracking moving objects in video. The detection methods discussed in the working are background subtraction, segmentation and detection methods using classifiers. From tracking methods are presented Kalman filter, TLD method and particle filter, which is described in detail in a separate chapter. Particle filter is a tracking method that calculates a new estimate of the state of the object based on a set of particles. These particles are defined by their positions and weights. For each next state are these values iteratively updated by measurement and resampling. From the acquired knowledge and expertise has been created a program that demonstrates the activity of the particle filter. This program is part of the DVD, where are also the samples and test results.

Klíčová slova

Počítačové vidění, Částicový filtr, Sledování objektů

Keywords

Computer vision, Particle filter, Object tracking

Citace

David Záděra: Detekce a sledování malých pohybujících se objektů, bakalářská práce, Brno, FIT VUT v Brně, 2011

Detekce a sledování malých pohybujících se objektů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Romana Juránka. Dále prohlašuji, že jsem řádně uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Záděra
17. května 2011

Poděkování

Tímto bych chtěl poděkovat vedoucímu bakalářské práce Ing. Romanovi Juránkovi za ochotu, odbornou pomoc a cenné rady, které byly poskytnuty během řady konzultací a přispěly k úspěšnému zpracování této bakalářské práce. Dále bych také rád poděkoval Ing. Michalovi Hradišovi za poskytnuté materiály k testování programu.

© David Záděra, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Detekce a reprezentace objektů	3
1.1	Detekce objektů	3
1.2	Reprezentace objektů	5
2	Sledování objektů	7
2.1	Rozdělení sledovacích metod	8
2.2	Kalmanův filtr	10
2.3	Metoda TLD	11
2.4	Použití sledovacích metod	11
3	Částicový filtr	13
3.1	Bayesovský odhad	13
3.2	Popis částicového filtru	14
4	Implementace	16
4.1	Návrh programu	16
4.2	Implementace částicového filtru	17
4.3	Popis programu	19
5	Testování	20
5.1	Implementace a použití testů	20
5.2	Vyhodnocení testů	21
6	Závěr	25

Úvod

Tato bakalářská práce se zabývá detekcí a sledováním objektů ve videu. Tyto činnosti můžeme zařadit do oblasti počítačového vidění, které je jednou z rychle rozvíjejících se oblastí současné informatiky a nachází tak uplatnění v celé řadě odvětví lidské činnosti. Jedná se především o oblasti, kde je zapotřebí získávat a analyzovat informace zachycené v obraze. Zde můžeme jmenovat například mapování provozu v dopravě, sledování letu raket a letadel, využití u kamerových systémů v průmyslových objektech či uplatnění v dalších oborech jako je lékařství nebo vojenství. S tím jak se snižuje cena hardware a zvyšuje se propracovanost detekčních a sledovacích metod roste i počet aplikací a systémů, kde lze tyto metody využít. Cílem této práce je seznámit čtenáře s postupy a vybranými metodami, které jsou teoretickým základem pro pochopení dané problematiky. Jednou z těchto metod je částicový filtr, jemuž je věnována samostatná kapitola, neboť je jednou ze stěžejních částí této bakalářské práce. Mezi další sledovací metody zmíněné v této práci patří Kalmanův filtr a moderní metoda TLD. Tyto metody jsou založeny na rozdílných principech, které jsou v práci popsány a rozebrány. Kromě popisu principů vybraných metod se práce věnuje také uvedení příkladů jejich použití v nejrůznějších aplikacích a systémech.

V první kapitole jsou stručně popsány možnosti, které vedou k detekci a reprezentaci objektů. Poté následují kapitoly dva a tři, věnující se popisu sledovacích metod. Následující druhá polovina práce se zabývá implementací metody částicového filtru a popisem programu, který využívá částicový filtr ke sledování objektů. Testování výsledného programu, budoucí vývoj a shrnutí výsledků jsou obsaženy v páté kapitole a závěru.

Kapitola 1

Detekce a reprezentace objektů

1.1 Detekce objektů

Detekce objektů v obraze představuje jeden ze základních úkolů, který předchází samotnému sledování objektů. Detekcí objektů zjišťujeme zda-li se v obraze vyskytuje pohybující se objekt, dále zjišťujeme jeho případnou pozici, natočení, možný směr pohybu, a některé další vlastnosti, které nás mohou zajímat. Nejčastěji jsou sledovanými objekty různé dopravní prostředky, osoby a nebo zvířata. V obraze může tedy existovat celá řada rozdílných objektů, které mají různé tvary a velikost a jejich detekce často není jednoduchá. Hlavními problémy při detekci objektů v obraze či videu jsou složitost tvaru, vzájemné překrývání dvou a více objektů, jejich natočení, deformace, měnící se velikost během pohybu, aj. Některé aplikace provádí detekci objektů v každém snímku videosekvence, kdy se často používá sekvence snímků k získání rozdílů v jednotlivých snímcích a na jejich základě se následně určí výskyt pohybujícího se objektu. Existují však i metody, které získávají informace vedoucí k detekci objektu z prvního výskytu objektu ve videu, tedy pouze z jednoho snímku. Tyto rozdílné přístupy detekce objektů jsou uvedeny v [12]. K detekci objektů lze využít řady metod, z nichž některé nejčastěji používané jsou popsány v následujícím textu.

Detekce pomocí klasifikátoru

Klasifikátor lze popsat jako algoritmus, který na základě množiny znalostí rozděljuje vstupní data do výstupních skupin (tříd). Úspěšná klasifikace je přitom podmíněna vhodnou volbou klasifikačního algoritmu a odpovídající množinou znalostí. Vstupem pro klasifikátor jsou trénovací data, kterými mohou být obrazová data (hodnoty pixelů) nebo také číselné a symbolické příznaky. Klasifikátory lze rozdělit jak podle použitých metod klasifikace (symbolické, sub-symbolické, statistické, paměťové), tak i podle charakteru učení (dávkové, inkrementální, inkrementální se zapomínáním). Asi nejznámějším příkladem metody využívající klasifikátorů je metoda AdaBoost (Adaptive boosting [16]). Tato metoda patří mezi tzv. boosting metody, jejichž základem je spojení více slabých klasifikátorů do jednoho úspěšnějšího klasifikátoru. Je tak dosaženo lepšího výsledku na základě několika klasifikátorů provádějících učení nad stejnou množinou vstupních (trénovacích) dat. Efektivita klasifikátoru je závislá na počtu a struktuře dat. Vhodný je rovnoměrný výběr několika prvků z každé třídy, které obsahují charakteristické příznaky pro danou třídu. V takovém případě lze dosáhnout velmi dobré úspěšnosti detekce.

Odečítání pozadí

Metoda odečítání pozadí je založena na vytvoření reprezentace scény, která se nazývá model pozadí (anglicky background model). V každém následujícím snímku jsou poté hledány změny, které se liší od původně vytvořeného modelu pozadí. Každá takováto změna značí rozpoznání pohybujícího se objektu. V [12] je uvedeno několik přístupů jak zjistit změny v jednotlivých snímcích. Jedním z nich je modelování barvy každého pixelu gaussovým rozložením $I(x, y) \approx N(\mu(x, y), \Sigma(x, y))$. Parametry střední hodnota $\mu(x, y)$ a rozptyl $\Sigma(x, y)$ jsou získány z měření v několika po sobě jdoucích snímcích. Další z přístupů využívá kromě informací o barvě jednoho pixelu i informace z jeho okolí. Každý pixel ve snímku není porovnáván pouze s odpovídajícím pixelem v modelu pozadí, ale také s okolními pixely. Tím lze snížit závislost na změnách osvětlení. Jiný z přístupů odečítání pozadí reprezentuje intenzitu pixelu jako diskrétní stavy. Za použití Markovových modelů lze zařadit obrazové body do jedné ze tří skupin. Takto reprezentované body se mohou nacházet buď v popředí, pozadí nebo jako stíny.

Metody detekce objektů založené na odečítání pozadí jsou nenáročné na výpočet a snadno implementovatelné. Díky tomu jsou často využívány u řady sledovacích systémů. Ovšem existuje i řada omezení těchto metod. V literatuře [17] jsou uvedeny některé z těchto omezení:

- Rychlé změny v osvětlení mohou výrazně změnit barevnou charakteristiku pozadí, což u většiny metod založených na odečítání pozadí vede k chybným výsledkům.
- Může docházet k detekci objektů, které nejsou zajímavé z hlediska následujícího sledování. K takovýmto objektům patří např. vlající vlajka či pohybující se stromy ve větru.
- Metody založené na odečítání pozadí nejsou schopny rozlišovat mezi objekty a jejich stíny.
- U metod založených na odečítání pozadí je požadavek na použití u systémů se statickou kamerou. Důvodem je to, že pohyb kamery obvykle způsobuje výrazné zkreslení pozadí scény.

Segmentace

Segmentace obrazu má za cíl rozdělit obraz na části, které se vztahují k rozdílným oblastem nacházejícím se v obraze, neboli oddělit objekty od pozadí. Při segmentaci obrazu se vyskytuje několik problémů, mezi kterými jsou nejednoznačnost obrazových dat, různé výsledky pro různé parametry u stejné metody, aj. Segmentace lze rozdělit podle kritérií, které se při segmentaci využívají. V praxi se například používají segmentace prahováním, segmentace na základě detekce hran, segmentace narůstání oblastí, segmentace srovnání se vzorem. Na principu segmentace je založena také metoda Mean-Shift Clustering [12]. Na začátku je algoritmus inicializován náhodně rozloženými shluky. Poté se v iteraci vždy počítá střed dat ležících uvnitř každého shluku a na základě rozdílu nového a starého středu se spočítá Mean-Shift vektor. Tento cyklus se opakuje tak dlouho, dokud se mění střed (vektor) shluků. Mean-shift algoritmus lze použít v řadě aplikací požadující jak detekci hran, tak i rozpoznávání či sledování objektů.

1.2 Reprezentace objektů

Možností jak reprezentovat a popsat objekt existuje velké množství a často záleží na konkrétní situaci a znalosti systému, ve kterém chceme reprezentaci použít. Pomocí reprezentace objektů se často snažíme vyjádřit jsou-li si dva objekty podobné. Proto je důležité zvolit vhodnou reprezentaci danou odpovídající volbou charakteristických vlastností objektů. Objekt lze popsat například na základě jeho interních znaků jako jsou informace z histogramu, barva objektu, rozložení jasu, textury, apod. Kromě uvedených obrazových vlastností lze k popisu objektů použít i externí reprezentaci, která zahrnuje informace o jejich hranici. Této charakteristiky se využívá v případech, kdy je důležitý tvar objektu. Ten může být vyjádřen některým z následujících způsobů:

- Množinou bodů nebo jedním bodem umístěným v těžišti objektu. Reprezentace pomocí bodů je vhodná pro sledování objektů, které zabírají malou oblast.
- Jednoduchými geometrickými tvary, jako je obdélník nebo elipsa.
- Obrysem nebo siluetou objektu.
- Modelováním a popisem jednotlivých částí objektu např. u člověka jednotlivé části těla, u letadla trup a křídla, apod.
- Kostrou objektu, která vytváří spojnici důležitých částí objektu.

Podrobněji jsou tyto a další tvarové charakteristiky objektů rozebrány v [12].

Pro správnou reprezentaci objektů je taktéž důležitá nezávislost (invariantnost) reprezentace objektů vůči posunutí, rotaci a změně velikosti.

Histogram

Histogram představuje jednu z možností, jak vyjádřit obrazové informace o objektu. Histogram udává rozložení hodnot jasu v obraze. Matematicky jej lze vyjádřit jako diskrétní funkci:

$$h(f_k) = n_k \quad (1.1)$$

kde f_k udává k -tou úroveň jasu ($k = 0, \dots, L - 1$ v obraze s L úrovněmi jasu) v obraze a n_k vyjadřuje počet pixelů v obraze odpovídající úrovni jasu f_k . Pro šedý obrazu budou hodnoty jasu v histogramu odpovídat hodnotám získaným z vzorce:

$$f_k = 0,299R + 0,587G + 0,114B \quad (1.2)$$

přičemž f_k udává k -tou úroveň jasu v šedém obraze a R, G, B vyjadřují odpovídající barevné složky v obraze, z něhož byl šedý obraz získán. V různých aplikacích lze pro reprezentaci objektů pomocí histogramu zvolit různý počet binů (např. z důvodů snížení výpočetních nároků). Index binu pro určitou hodnotu jasu z histogramu potom získáme podle vzorce:

$$B_i = N \frac{f_k}{L} \quad (1.3)$$

B_i vyjadřuje index binu do něhož spadá daná hodnota jasu ($i = 0, \dots, N - 1$), N udává celkový počet binů, f_k je k -tá úroveň jasu a L celkový počet úrovní jasu v obraze.

Normalizovaný RG model

RGB model představuje základní barevný model pro zpracování a ukládání obrazových dat. Pro reprezentaci objektů RGB modelem se používá normalizovaný RGB model [3], který lze získat ze tří RGB složek pomocí normalizace:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (1.4)$$

přičemž pro složky r , g , b platí $r + g + b = 1$. Významovou informaci o rozložení barev přitom nesou pouze dvě z těchto složek. Třetí složka se dá v případě potřeby dopočítat jako doplněk do jedné.

RGB model lze použít k reprezentaci objektů v případech, kdy předpokládáme neměnný zdroj osvětlení. Pokud by došlo ke změně jasu osvětlení nebo změně rozložení barev, sledovaný objekt kamerou by byl reprezentován jinými hodnotami než referenční model, přestože by se jednalo o stejný objekt. V takovém případě by došlo k chybnému sledování.

Histogram orientovaných gradientů

Histogram orientovaných gradientů (anglicky Histogram of oriented gradient – HOG) představuje typ reprezentace, která popisuje tvar objektu získaný na základě zachycení rozložení orientací hran v obraze. V [7] je uveden postup, který rozděluje vstupní obraz na malé části a pro každou takto vytvořenou část je tvořen 1-D histogram velikostí gradientů 1.5 nebo jejich orientací 1.6 ze všech pixelů této oblasti. Velikost a orientace gradientů jsou počítány podle následujících rovnic:

$$r = \sqrt{dx(x, y)^2 + dy(x, y)^2} \quad (1.5)$$

$$\varphi = \arctan\left(\frac{dx(x, y)}{dy(x, y)}\right) \quad (1.6)$$

Takto vytvořené oblasti jsou spojovány do větších celků které jsou nazývány HOG deskriptory. Je vhodné provést normalizaci oblastí, čímž lze docílit větší nezávislost na osvětlení a stínech. Výhodou tohoto postupu, který rozděluje vstupní obraz na menší oblasti, je zvýšení nezávislosti změn hodnot histogramu na malých změnách tvaru a natočení v jednotlivých oblastech.

Kapitola 2

Sledování objektů

Objekty ve videu je možné detekovat a reprezentovat různými způsoby, které jsou popsány v kapitole 1. V návaznosti na detekci a reprezentaci objektů se vztahuje úkol jejich sledování. Sledování objektů je ve většině případů založeno na stanovení vztahu mezi instancí objektu (první výskyt objektu) a jeho výskytem v každém následujícím snímku. V [12] jsou uvedeny dva možné přístupy k tomuto problému - zjištění vztahu mezi instancí objektu a jeho následujícím výskytem odděleně pro každý snímek nebo zohlednění informací z předchozích snímků. V prvním případě je oblast výskytu objektu získána na základě detekčních algoritmů v každém snímku a následně je aplikováno sledování. V druhém případě je oblast výskytu a pohyb objektu v této oblasti počítána iterativně na základě aktualizací jeho polohy z předchozího snímku.

Cílem sledování objektů je snaha zjistit jejich aktuální pozici ve sledovaném prostoru a určit informace o jejich výskytu v časovém intervalu, nebo-li určit jejich trajektorii. Tyto informace můžeme dále použít k další analýze chování objektů. Pokud sledujeme objekty, často se také neobejdeme bez následné analýzy pohybu. Díky analýze pohybu můžeme získat řadu informací o sledovaném objektu a odhadnout tak následující pozici objektu a jeho trajektorii. S pojmem analýza pohybu souvisí také tzv. dynamický model. Dynamický model se zabývá pohybem objektů v čase a jeho změnami. Většinou nás zajímají veličiny jako jsou rychlost, zrychlení a směr pohybu. Naproti tomu statický model se zabývá jak je daný objekt v obraze reprezentován, jak vypadá a jak ho rozpoznat. Ve většině případů se analýza pohybu provádí ve 2D obraze, který je reprezentací 3D scény. Takovému prostoru se říká rychlostní pole. Každému bodu rychlostního pole je přiřazen dvousložkový vektor, který určuje směr a rychlost pohybu daného bodu. K vyjádření dynamiky pohybu je zapotřebí nejen informace o poloze objektu (prostorová dimenze), ale i dimenze časová. K tomuto účelu se požaduje sekvence snímků snímáných v časových úsecích Δt dostatečně krátkých na to, aby bylo možno zaznamenat i velmi rychlé pohyby. Dynamický obraz je tedy závislý na časové složce t a lze jej vyjádřit funkcí $f(x, y, t)$, kde x a y určují pozici objektu v obraze a t udává pořadí obrazu ve videosekvenci.

Analýza pohybu je součástí téměř všech aplikací zabývajících se sledováním objektů. Podle [8] existují z praktického pohledu tři hlavní skupiny úloh zabývajících se analýzou pohybu:

Detekce pohybu

V tomto případě je hlavní snahou rozpoznat a zachytit pohyb v obraze. K tomuto účelu se zpravidla používají metody založené na segmentaci obrazu nebo rozdílů snímků.

Rozpoznání a lokalizace pohybujících se objektů

Tento typ úlohy zahrnuje více komplexnější přístup než detekce pohybu. Zahrnuje jak detekci pohybujícího se objektu, tak i sledování jeho trajektorie a odhad následného pohybu. Při řešení tohoto typu úloh se nejčastěji používají metody hledání významných bodů a vztahů mezi nimi.

Odvození vlastností objektů v 3D prostoru

Řešení této skupiny úloh vede ke snaze určit 3D vlastnosti objektů z 2D obrazů získaných z několika snímků v různých časových okamžicích.

Každá z výše uvedených úloh je vhodná k použití u rozdílných systémů. Podle pohybu snímací soustavy a snímané scény lze sledovací systémy rozdělit do čtyř kategorií:

- Statická kamera, nepohybující se objekt
- Pohybující se kamera, nepohybující se objekt
- Statická kamera, pohybující se objekt
- Pohybující se kamera, pohybující se objekt

Úlohy první skupiny (Detekce pohybu [2](#)) je vhodné aplikovat v systémech používající statickou kameru, zatímco druhou skupinu úloh (Rozpoznání a lokalizace pohybujících se objektů [2](#)) lze řešit i v případech, kdy se pohybuje jak kamera tak i objekt.

2.1 Rozdělení sledovacích metod

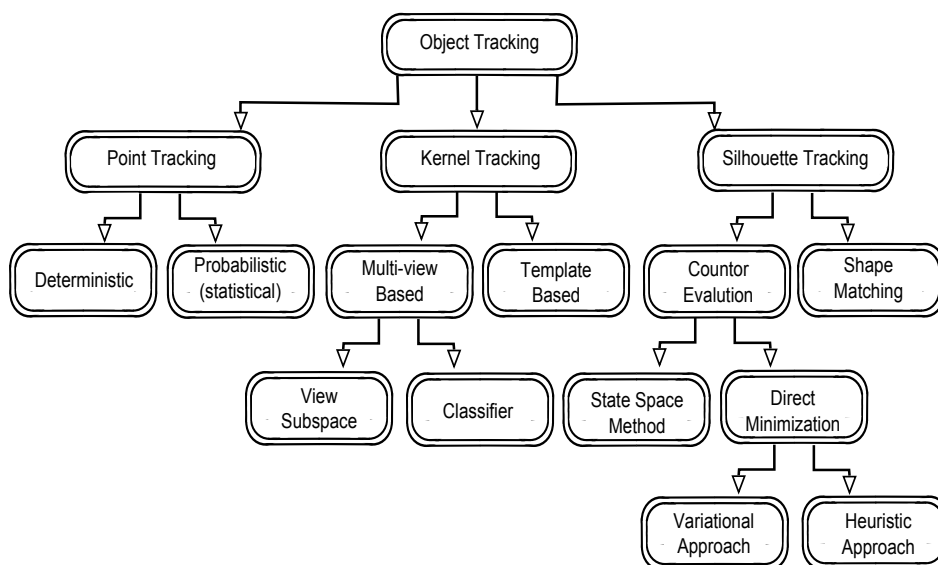
Následující rozdělení sledovacích metod vychází z literatury [\[12\]](#). Jak ukazuje obrázek [2.1](#), sledovací metody lze rozdělit do tří hlavních skupin, kterými jsou Point tracking, Kernel tracking a Silhouette tracking. U většiny zde uvedených metod jsou použity anglické názvy, neboť český překlad buď není používán nebo nevystihuje přesný název dané metody.

Kernel tracking

Kernel tracking metody provádí sledování objektů, které jsou reprezentovány jednoduchými geometrickými tvary a jejich vzhledem. Hlavním prvkem je jádro, které zahrnuje informace o tvaru a histogramu objektu. Objekty jsou sledovány v jednotlivých snímcích počítáním nových hodnot jádra na základě jeho pohybu. Obecně je pohyb objektu vyjádřen parametry pohybu (posun, rotace). Kernel metody se mohou lišit v řadě hledisek, kterými jsou:

- Použitý vzhled reprezentace
- Počet sledovaných objektů
- Metody použité k odhadu pohybu objektů

Na základě těchto parametrů, lze rozdělit tyto metody do dvou podkategorií: templates and density-based appearance models a multiview appearance models.



Obrázek 2.1: Rozdělení sledovacích metod, převzato z [12].

Point tracking

Tato skupina sledovacích metod používá k reprezentaci objektů množinu bodů, které vychází z informací z předcházejícího výskytu objektu a mohou zahrnovat informace o jeho pozici a pohybu. Sledování objektů je založeno na hledání vztahu mezi detekovaným objektem vyjádřeným těmito body v sekvenci snímků. K použití této reprezentace je vyžadováno externího mechanismu pro detekci v jednotlivých snímcích. Point tracking metody mohou být rozděleny do následujících dvou skupin, a to na deterministické a statistické metody. Tyto dvě skupiny se liší v přístupu hledání vztahu mezi body v jednotlivých snímcích.

Silhouette Tracking

Sledované objekty mohou být vyjádřeny různě složitými tvary (části lidského těla, části auta, apod.), které nelze přesně popsat pomocí jednoduchých geometrických tvarů. Silhouette tracking metody se snaží poskytnout přesný popis tvaru těchto objektů. Hledání jednotlivých částí objektů je prováděno prostřednictvím objektového modelu vytvořeného z předchozích snímků. Tento objektový model může být ve formě barevného histogramu, tvaru objektu (tzv. mapy hran) a nebo tvaru obrysu. Silhouette Tracking metody mohou využívat dva přístupy zpracování tvarů objektu, a to shape matching a contour tracking. Shape matching vyhledává v každém aktuálním snímku siluetu objektu odpovídající referenčnímu popisu. Contour tracking přístupu vyvíjí původní obrys objektu na základě jeho nové pozice v aktuálním snímku za použití stavového prostorového modelu nebo přímé minimalizace některých energetických funkcí.

2.2 Kalmanův filtr

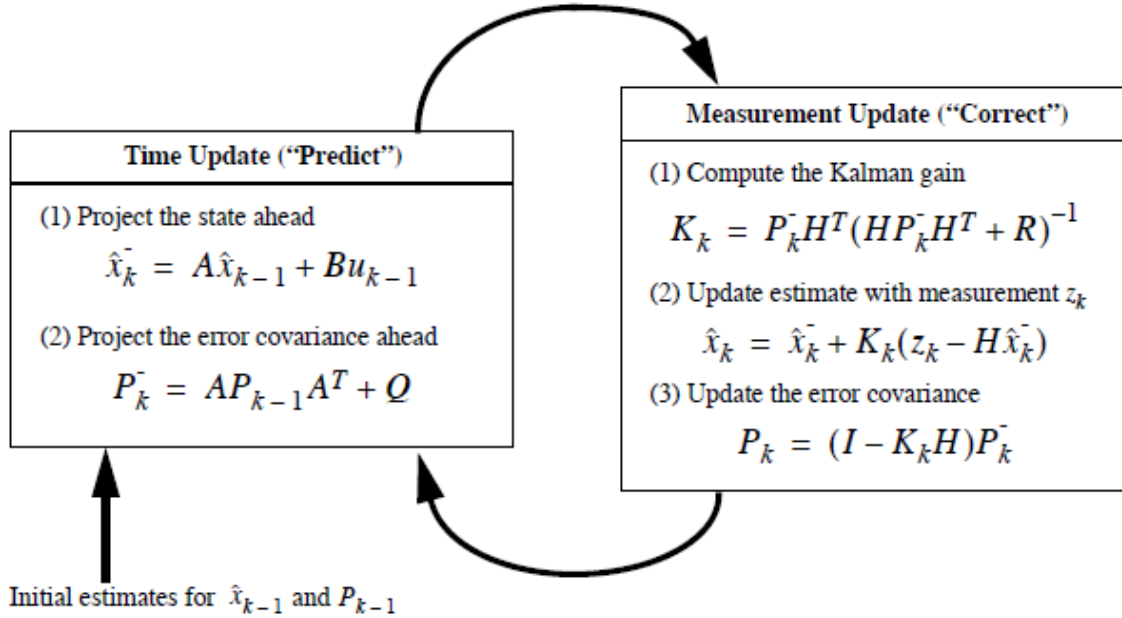
Kalmanův filtr je matematická metoda používaná k výpočtu odhadu stavu lineárních systémů. Použití Kalmanova filtru nachází uplatnění nejen v počítačovém vidění, ale také např. v navigačních a sledovacích systémech, měřicích systémech nebo v ekonomických aplikacích a při zpracování makroekonomických dat. Jeho výhodou je rekurzivní struktura, kdy se v každém kroku upravují jednotlivé údaje pro co nejlepší odhad následujícího stavu. Mezi omezení Kalmanova filtru patří nutnost linearity systému. Pro použití v nelineárních systémech je zapotřebí některého z rozšíření Kalmanova filtru.

Základním cílem Kalmanova filtru při určování polohy objektu je odhad polohy x_k podle lineární stochastické diferenční rovnice na základě měření skutečného stavu z_k :

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (2.1)$$

$$z_k = Hx_k + v_k \quad (2.2)$$

Procesní šum w_k a šum měření v_k odpovídají normálnímu rozložení $p(w) = N(0, Q)$ a $p(v) = N(0, R)$. Matice A v rovnici 2.1 označuje přechodovou matici mezi stavy x_{k-1} a x_k , matice B představuje kontrolní vektor vstupu stavu x .



Obrázek 2.2: Znázornění Kalmanova filtru, převzato z [2].

Samotný algoritmus Kalmanova filtru se skládá ze dvou fází, kterými jsou predikce (odhad stavu) a korekce (aktualizace stavu). Predikční krok obsahuje rovnice, které realizují výpočet následujícího stavu:

$$x_k^- = Ax_{k-1} + Bu_k \quad (2.3)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.4)$$

V korekčním kroku se realizuje začlenění měření do nového odhadu stavu. Tento krok vyjadřují rovnice:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (2.5)$$

$$x_k = x_k^- + K_k(z_k^- - H_k X_k^-) \quad (2.6)$$

$$P_k = (I - K_k H_k) P_k^- \quad (2.7)$$

Nejprve se určí tzv. Kalmanův zisk K_k , který nám říká jaká je váha nové informace oproti té, kterou už známe. Následujícím krokem je aktualizace stavu na základě měření z_k . Nakonec se určí kovariance chyby odhadu P_k .

Celkový Proces fungování Kalmanova filtru shrnuje obrázek 2.2. Podrobnější popis výpočtů a fungování Kalmanova filtru lze nalézt v literatuře [2], [11], odkud byly také převzaty výše uvedené rovnice a obrázek.

2.3 Metoda TLD

TLD lze v současné době zařadit mezi nejrobustnější a nejefektivnější metody sledování. Jak již název napovídá, TLD (Tracking-Learning-Detection) představuje algoritmus, který kromě sledování využívá i učení a detekci objektů [19]. TLD umožňuje dlouhodobé sledování objektů s minimem znalostí o objektu samotném i prostředí, ve které se pohybuje. Objekty jsou sledovány od jejich prvního výskytu a dokonce i poté co zmizí mimo rámec videa a znovu se objeví. Toho je dosaženo díky adaptivnímu sledování a detektoru, který je postupně trénován z negativních a pozitivních výsledků. Detektor tak umožňuje znovu inicializovat sledování v případě jeho selhání. Sledování je prováděno Median Shift algoritmem, který měří počet lokálních posunutí a za pomoci mediánu se odhadne hranice pohybu. Detektor je založen na prohledávání vstupního obrazu a v jednotlivých částech se binárním klasifikátorem rozhodne o přítomnosti objektu. Cílem učení je zavedení inicializačního detektoru, který by umožňoval lokalizovat některý z dříve pozorovaných objektů. V průběhu sledování je detektor aktualizován tzv. PN učením. Místa, která jsou blízko cílové trajektorie jsou označována jako pozitivní, negativní jsou daleko od této pozice. P-aktualizace zlepšují vlastnosti detektoru pozitivními příklady, N-aktualizace zvyšují počet negativních příkladů. Tato kombinace učení ze dvou kategorií tak umožňuje celkové zlepšení metody.

2.4 Použití sledovacích metod

V současné době se objevuje stále více aplikací reagujících na gesta lidské ruky a pohyby lidského těla. Tyto aplikace se stávají součástí téměř všech elektronických zařízení kolem nás jako jsou např. PC, televize, videohry, mobily, apod. Ovládání zařízení pomocí gest a pohybů se stává stále běžnějším způsobem komunikace člověka s nejrůznějšími zařízeními a představuje stále populárnější typ uživatelského rozhraní. Některé oblasti použití gest a pohybů lidské ruky jsou popsány v [18]. Mezi velmi rozšířenou a oblíbenou oblast použití pohybů a gest lze zařadit hraní her, přehrávání muziky a zábavu obecně. V současné době nejsou výjimkou simulace, kdy lze ovládat auto nebo letadlo, aniž byste drželi volant nebo je možné hrát tenis bez rakety. Vše se děje jen na základě snímání polohy a pohybu rukou kamerami. V nedávné době byla představena virtuální počítačová myš, jejíž pohyby jsou tvořeny snímáním tvaru a pohybů lidské ruky. Rozpoznávání pohybů a gest rukou může také pomoci k lepšímu dorozumění se s hluchoněmými lidmi a v komunikaci. Znakovou řeč většinou ovládají právě jen hluchoněmí lidé a speciálně vyškolení pracovníci a většina lidí by jistě přivítala komunikační zařízení pro porozumění nejen znakové řeči, ale i gestům které jsou součástí běžné komunikace. Při komunikaci má každý pohyb těla přiřazen svůj význam a z gramatických pravidel a kontextu lze odvodit jejich význam právě zařízeními,

kteřá tento pohyb sledují. Jiný typ využití gest rukou je představen v [5], kde je ukázáno virtuální prostředí využívající gesta ruky jako vstupního a kontrolního zařízení. Tento systém představuje interaktivní modelování a simulaci biopolymerů, kde lze k jejich tvorbě využít právě pohybů rukou. Zařízení využívající gest rukou a pohybů existuje nepřeberné množství a jejich počet bude v budoucnu zřejmě narůstat.

Použití sledovacích metod v bezpečnostních systémech nachází uplatnění na řadě míst, kterými jsou areály firem případně rodinných domů, zabezpečení veřejných míst policejními kamerami, hlídání v obchodech proti krádežím apod. Před samotným nasazením bezpečnostních systémů by měla předcházet analýza daného prostředí a následná volba vhodného bezpečnostního zařízení a systému. Tomu může odpovídat použití systémů pro venkovní a vnitřní prostory, použití systémů se statickou nebo pohyblivou kamerou, systém s tvorbou záznamů a dalším automatickým vyhodnocením apod.

Další rozsáhlou oblastí použití sledovacích metod je doprava. V současné době se nejen ve velkých městech zvyšuje počet automobilů a s tím roste i potřeba kontroly, sledování a řízení dopravy. Typickými zdroji dopravních dat jsou kamery, které mohou přinášet informace o dopravě v dané oblasti jako celku, případně sledovat a poskytovat údaje o jednotlivých vozidlech. O vozidlech lze např. zjišťovat jejich registrační značku, typ vozidla, rychlost, ale i kdo automobil řídil. Údaje o počtu projetých aut danou lokalitou nebo vytíženosti křižovatek jsou důležité pro plánování nových dopravních cest. Tyto údaje většinou nejsou zpracovávány přímo detekčními a sledovacími zařízeními, kterými byly pořízeny, ale jsou posílána do centra sběru dat k dalšímu zpracování a vyhodnocení. Nemalou roli hrají také bezpečnostní systémy uvnitř aut nebo letadel kontrolující stav řidiče. Většina velkých automobilových společností se podílí na vývoji a nasazení systémů, které sledují pohyb očí řidiče a můžou tak zabránit nehodě způsobené mikrospánkem nebo únavou.

Kromě uvedených oblastí použití sledovacích metod existují i další, kde za zmínku stojí například ještě oblasti jako je lékařství, vojenská technika, navigační a satelitní systémy, apod. Se zvyšující se dostupností sledovacích zařízení a propracovaností sledovacích algoritmů lze očekávat zvyšující se počet oblastí nasazení sledovacích systémů.

Kapitola 3

Částicový filtr

Částicový filtr, označován také jako Sekvenční Monte Carlo Metoda (SMC) [4], patří do skupiny numerických metod řešících problém odhadu stavu systému pro nelineární a ne-gaussovské (non - Gaussian) dynamické modely. Tato metoda byla představena v roce 1993 a od této doby se objevila řada různých podob tohoto algoritmu. Mezi nejčastěji používané implementace částicového filtru patří condensation algoritmus, bootstrap filtering, interacting particle approximation, a survival of the fittest. V současné době nachází využití v aplikacích zabývajících se počítačovým viděním, ekonometrií, robotikou a navigací.

3.1 Bayesovský odhad

Nelineární diskrétní dynamický systém je uváděn jako dvojice - odhad stavu systému x_t a skutečná hodnota (získaná měřením) tohoto stavu z_t v čase t [4], [10]:

$$x_t = f_t(x_{t-1}, v_{t-1}) \quad (3.1)$$

$$z_t = h_t(x_t, u_t) \quad (3.2)$$

kde f je funkce vývoje stavu systému a funkce h je funkcí pro ohodnocení (měření) modelu. v_t označuje procesní šum a u_t šum vzniklý při měření. Z Bayesovského hlediska je sledování založeno na rekurzivním výpočtu stavu x_t , přičemž do tohoto výpočtu jsou zahrnuty data z ohodnocení v časovém intervalu 0 až t . K tomuto účelu se používá funkce pravděpodobnostní hustoty (probability density function *pdf*). Funkce $pdfp(x_t|z_{1:t})$ je počítána rekurzivně ve dvou krocích, a to predikci a aktualizaci. V predikční fázi je použit pohybový model 3.1 a za použití Chapman–Kolmogorovy rovnice lze získat *pdf* ve tvaru:

$$P(x_t|z_{1:t}) = \int P(x_t|x_{t-1})P(x_{t-1}|z_{1:t-1})dx_{t-1} \quad (3.3)$$

Aktualizační část zahrnuje použití rovnice 3.2. V čase $t + 1$ lze aplikovat Bayesovu větu:

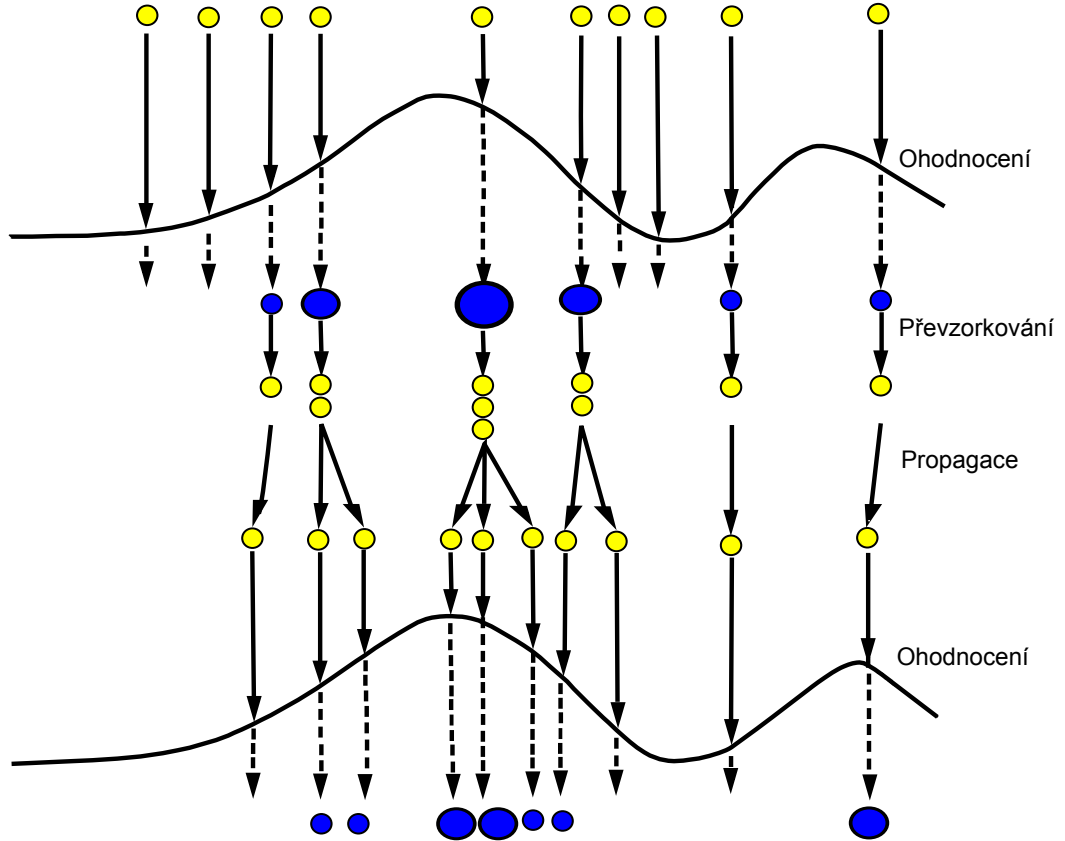
$$P(x_t|z_{1:t}) = \frac{P(z_t|x_t)P(x_t|z_{1:t-1})}{P(z_t|z_{1:t-1})} \quad (3.4)$$

přičemž normalizační konstanta

$$P(z_t|z_{1:t-1}) = \int P(z_t|x_t)P(x_t|z_{1:t-1})dx_t \quad (3.5)$$

závisí na pravděpodobnostní funkci $P(z_t|x_t)$.

Zde uvedený postup nemá analytické řešení a proto je potřeba použít numerických metod. Pro řešení lze použít některou z metod částicový filtr, rozšířený Kalmanův filtr nebo grid-based filtr.



Obrázek 3.1: Fungování částicového filtru, částečně převzato z [14].

3.2 Popis částicového filtru

Částicový filtr představuje implementaci Bayesova filtru vycházející ze simulace Monte Carlo metody [10]. Základní myšlenkou částicového filtru je odhad nové pozice objektu na základě náhodně vybraného souboru vzorků (částic), které reprezentují možný výskyt objektu v čase na dané pozici. Výhodou použití vzorků k odhadu nové pozice je, že neklademe žádné omezení na tvar hustoty pravděpodobnosti výskytu objektu v obraze. V každém následujícím stavu je jednotlivým částicím přiřazována váha w na základě shody reprezentace dané částice se skutečným stavem. Tato váha odpovídá pravděpodobnosti výskytu sledovaného objektu na daném místě v čase t . Z těchto vah a pozic částic je vypočten odhad nové pozice objektu $x_t = \sum_{i=1}^N x_t^{(i)} w_t^{(i)}$ pro normalizované hodnoty vah $\sum_{i=1}^N w_t^{(i)} = 1$. K tomu abychom získali novou hustotu pravděpodobnosti je zapotřebí provést převzorkování částic. Princip převzorkování spočívá v tom, že vytvoříme novou sadu částic, kde jejich hustota rozmístění odpovídá poměru vah částic z předcházejícího stavu. To znamená, že částice

s vyšší váhou mají větší pravděpodobnost, že ve svém okolí budou generovat novou částici. Princip převzorkování je patrný z obrázku 3.1.

Výpočty pro jednotlivé kroky částicového filtru jsou uvedeny v kapitole 4, která se věnuje implementaci programu pro sledování objektů, který je založen na metodě částicového filtru.

Kapitola 4

Implementace

Následující kapitola se zabývá implementací a popisem programu, který sleduje objekty ve videu. Značná část je věnována implementaci částicového filtru, který představuje sledovací algoritmus použitý v programu. Tento program byl vytvořen v programovacím jazyce C++ s využitím knihovny OpenCv, což je open source knihovna pro použití v počítačovém vidění a pro zpracování obrazu (podrobnější popis v literatuře [11]). Program používá objektově orientovaný přístup pro vytvoření tříd modelující sledované objekty a jejich chování. Program byl napsán v operačním systému Windows XP a kód je přenositelný i na systémy s operačním systémem Unix. Název programu po kompilaci je track.

Program track je použitelný jak pro sledování objektů kamerou, tak i pro sledování objektů ve video souborech ve formátu .avi. V jednom videu (vstupu z kamery) lze přitom sledovat více objektů současně. Tyto objekty jsou z implementačního hlediska reprezentovány jedním ze dvou různých modelů. Použitými reprezentacemi jsou barevný RG modelem a reprezentace tvořená histogramem orientovaných gradientů (HOG). Typ reprezentace objektů lze během přehrávání videa měnit, lze tedy pro dva různé objekty v jednom videu použít dvou rozdílných reprezentací.

4.1 Návrh programu

Při návrhu programu lze vycházet z postupů uvedených v literatuře [1], [6], [13]. Zde uvedené algoritmy částicového filtru mohou být rozděleny na dva logické celky:

1. Algoritmus popisující částicový filtr
2. Část popisující implementaci reprezentace objektů

Zatímco implementaci části popisující algoritmus částicového filtr lze vytvořit souhrnně pro použití v různých systémech, vytvoření reprezentace objektů je často závislé na konkrétním použití (např. sledování objektů s měnící se barvou, tvarem). Pro implementaci reprezentací různých modelů je vhodné použít abstraktní model.

Samotný algoritmus částicového filtru představují následující kroky:

1. **Inicializace** zahrnuje vytvoření N částic v okolí objektu, jejichž rozmístění odpovídá normálnímu rozdělení pravděpodobnosti. Sledovanému objektu je na začátku přiřazen referenční histogram q obsahující b binů $q = q^{(u)}, u = 1, \dots, b$.

2. Krok **ohodnocení částic** zahrnuje vytvoření odpovídajícího histogramu pro každou částici $q = p_i^{(u)}$, $u = 1, \dots, b$, $i = 1, \dots, N$ a porovnání tohoto histogramu s referenčním histogramem. Na základě porovnání histogramů je částicím přiřazena váha, která odpovídá pravděpodobnosti výskytu objektu na pozici dané pozicí částice.
3. **Normalizace vah** všech částic je určena vztahem: $\sum_{i=1}^N w_t^{(i)} = 1$.
4. **Odhad nové pozice** objektu je získán z pozic a vah všech částic: $x_t = \sum_{i=1}^N x_t^{(i)} w_t^{(i)}$.
5. Během **převzorkování částic** se vytvoří nová sada částic, která zohledňuje nové ohodnocení a částice s vyšší váhou mají větší pravděpodobnost, že ve svém okolí budou generovat novou částici.
6. Nová pozice objektu je vykreslena ve videu.
7. Výpočet se iterativně opakuje od kroku 2.

V jednotlivých fázích algoritmu je vstupem struktura obsahující informace o stavu objektu v čase $t - 1$, výstupem je opět tatáž struktura ovšem obsahující údaje v čase t . Tato struktura zahrnuje údaje o počtu částic, jejich distribuci tvořenou pozicemi a ohodnocením částic a také pozici objektu v současném a předchozím stavu.

4.2 Implementace částicového filtru

V této části je uveden popis konkrétní implementace výše uvedeného návrhu, jejímž výsledkem je program s názvem track. Implementace je založena na objektovém programování obsahující třídu Object pro modelování vlastností a metod sledovaných objektů a třídu Objectlist, která zahrnuje práci s větším počtem objektů. Atributy třídy Object jsou tvořeny informacemi o objektu a údaji o částicovém filtru. Metody třídy Object odpovídají krokům algoritmu, které jsou uvedeny v návrhu 4.1.

Inicializaci částicového filtru zajišťují metody **InitialiseFilter**(int w, int h, Options o) a **InitialiseModel**(void), přičemž první metoda inicializuje pozice a váhy částic a počáteční stav částicového filtru, druhá metoda se stará o vytvoření referenčního histogramu pro zvolený model reprezentace částic. Jako parametry jsou předávány inicializační hodnoty filtru. Vytvoření histogramu poskytuje funkce **GetHistogram**(Histogram* h, int x, int y). Parametry x , y jsou souřadnice dané částice a parametr h představuje strukturu pro uložení histogramu dané částice. Histogram je vytvářen z několika oblastí, které vyplňují oblast o velikosti objektu v okolí částic. Pro zvýšení spolehlivosti modelu jsou pixely více vzdáleny od středu oblasti započítávány s nižší hodnotou podle váhové funkce:

$$g(r) = 1 - \exp^{-r^2} \quad (4.1)$$

kde r je normalizovaná vzdálenost od středu oblasti. Tento postup je převzat z literatury [1]. Ohodnocení částic je realizováno metodou **Measurement**(IplImage* A, IplImage* B), přičemž parametry této metody jsou matice obsahující aktuální indexy binů přiřazeny každému bodu v obraze 4.2. Tato metoda pro každou částici vytváří histogram a přiřazuje částicím jejich ohodnocení. Ohodnocení částic je tvořeno metodou **CmpHistogram**(void), která porovnává histogram dané částice s referenčním histogramem. Výsledkem porovnání

je reálné číslo z rozmezí 0 až 1, přičemž metrika pro porovnání histogramů vychází z výpočtu Bhattacharyovy vzdálenosti [6], [11]:

$$d = \sqrt{1 - \sum_{u=1}^b \sqrt{p^{(u)}q^{(u)}}} \quad (4.2)$$

Nejlepší shoda dvou histogramů je vyjádřena 0, zvyšující se hodnota značí snižující se shodu histogramů. Váha i -té částice je poté dána ohodnocením:

$$w^{(i)} = \frac{1}{\sqrt{2\pi\sigma}} \exp^{-\frac{d^2}{2\sigma^2}}. \quad (4.3)$$

Odhad nové polohy objektu je počítán v metodě **RunFilter**(void). Výpočet odhadu nové polohy objektu je stanoven z pozic všech částic a jejich vah podle odhadu stavu 4. Metoda **RunFilter**(void) zároveň zahrnuje výpočet rychlosti objektu $v = x_t - x_{t-1}$ pro použití v dynamickém modelu, který má tvar:

$$x_{t+1} = x_t + v + N(0, \Sigma) \quad (4.4)$$

Implementace převzorkování částic je obsažena v metodě **Resampling**(void). Tato metoda vytváří novou sadu částic $(x_t^{(i)}, w_t^{(i)})_{i=1, \dots, N}$ s využitím normalizované kumulativní pravděpodobnosti:

$$c^{(0)} = 0, \quad (4.5)$$

$$c^{(i)} = c^{(i-1)} + w^{(i)} \quad (4.6)$$

Algoritmus je založen na nalezení nejmenšího čísla j pro které $c^{(j)} \geq r$, přičemž r je náhodně vygenerované číslo z intervalu 0 až 1 podle rovnoměrného rozložení. Částice j poté ve svém okolí generuje novou částici i tak, že $x^{(i)} \approx x^{(j)}$. Tento postup je opakován pro všech N částic. Poslední metodou třídy **Object** je metoda **Drawing**(IplImage* img, float sw, float sh, bool v, bool t, bool m), která zajišťuje vykreslení nové pozice objektu ve videu a případné zobrazení trajektorie objektu a rozmístění částic podle nastavení hodnot předávaných parametrů.

K sledování většího počtu objektů ve videu slouží třída **Objectlist**. Tato třída obsahuje seznam všech objektů, které jsou sledovány. Obsahuje metody pro přidání nového objektu **AddObject**(IplImage* img, Options o), zrušení sledovaného objektu **DeleteObject**(int m) a sledování všech objektů **ObjectTracking**(IplImage* img, int c). Metoda **ObjectTracking**(IplImage* img, int c) postupně prochází seznam všech objektů a pro každý objekt volá metody třídy **Objekt** v příslušném pořadí. Tento proces se opakuje pro každý nový snímek videa pro získání nového stavu. Třída **ObjectList** také obsahuje metodu **Event**(IplImage* img, int c), která reaguje na vstup uživatele (zrušení objektu, výběr objektu, sledování trajektorie, zobrazení aktuálního rozmístění částic).

Implementace Modelů pro reprezentaci objektů

V programu track jsou implementovány dva modely pro reprezentaci objektů. Těmito modely jsou barevným RG modelem a HOG modelem. Algoritmus je založen na vytváření a porovnávání histogramů, které reprezentují daný model. Nejprve jsou každému bodu obrazu přiřazeny indexy podle typu modelu. Tyto indexy určují číslo binu histogramu, do něhož

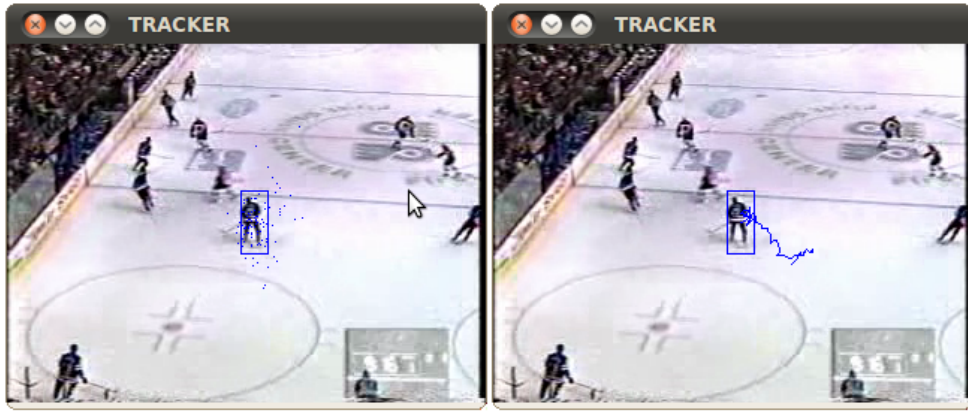
daný bod obrazu spadá. Každý bod obrazu má přiřazený dva indexy I_1, I_2 a do histogramu o indexech I_1 a I_2 se přičte hodnota podle váhové funkce 4.1. Hodnoty indexů I_1, I_2 jsou pro barevný RG model jsou počítány podle rovnic 4.7, pro HOG model podle rovnic 4.8:

$$I_1 = \frac{bnR}{R + G + B}, I_2 = \frac{bnG}{R + G + B} \quad (4.7)$$

$$I_1 = \frac{bn}{255} \sqrt{\frac{dx^2 + dy^2}{2}}, I_2 = \frac{bn}{2\pi} \left(\arctan\left(\frac{dx}{dy}\right) + \frac{\pi}{2} \right) \quad (4.8)$$

přičemž bn značí počet binů ve výsledném histogramu, R, G, B barevné složky pixelu v daném bodě a dx, dy představují změnu intenzity v tomto bodě. Z těchto indexů je potom sestaven histogram pro daný model a to tak, že histogram je tvořen všemi body z okolí částice o velikosti sledovaného objektu.

Výpočet polí indexů pro jednotlivé modely je součástí třídy Objectlist a objektům třídy Object jsou předávány jako parametry. Tím je každé pole indexů pro daný model a snímek počítáno pouze jednou.



Obrázek 4.1: Ukázka distribuce částic a trajektorie objektu.

4.3 Popis programu

Výsledný program track funguje jako konzolová aplikace. Vstupem programu je video ve formátu .avi (spouštěno s parametrem -video název_video souboru) nebo vstup přes kameru (spouštěno s parametrem -camera). Po spuštění programu s příslušným vstupem se očekává označení objektu, který se bude sledovat. Označení objektu lze provést pravým tlačítkem myši. Počet objektů, které lze sledovat v jednom videu, není omezen. S vysokým počtem objektů ovšem rostou výpočetní nároky programu a sledování může být pomalé. Během sledování objektů lze zobrazit rozmístění částic 4.1, sledovat trajektorii objektu 4.1, případně měnit model reprezentace objektů. Zapnutí těchto a dalších vlastností lze zjistit vypsáním nápovědy (parametr -help) nebo v souboru Readme na přiloženém DVD. Pomocí parametrů programu lze nastavit vlastnosti částicového filtru kterými jsou počet částic, počet binů histogramu, velikost oblastí na které je rozdělen sledovaný objekt a typ modelu. Program lze spustit taktéž v testovacím režimu. Tento režim společně s vyhodnocením úspěšnosti sledování programu track jsou uvedeny v kapitole 5.

Kapitola 5

Testování

Úspěšnost sledovacího algoritmu, který je součástí programu track (kapitola 4), byla ověřena sadou testů a experimentů. Testy byly provedeny na několika video souborech se zaměřením na získání výsledků pro různé nastavení parametrů sledovacího algoritmu a celkové stanovení úspěšnosti sledování. Testování programu probíhalo na video souborech poskytnutých ústavem počítačové grafiky a multimédií (UPGM).

5.1 Implementace a použití testů

Součástí vytvořeného programu track je testovací režim pro testování sledovacího algoritmu. Tento testovací režim je implementován přímo jako součást programu track a lze jej spustit přidáním vstupního parametru `-test` při spouštění programu. Za tímto parametrem se očekává název souboru, který obsahuje anotační záznamy pro daný video soubor. Jako soubor s anotačními záznamy lze použít soubor, který obsahuje záznamy v následujícím tvaru:

```
id  čas  x  y  w  h
```

s významem:

- Id je číslo sledovaného objektu
- Čas udává dobu, ve které se má porovnat anotační záznam s pozicí získanou programem, čas je udáván v milisekundách
- X-ová souřadnice levého horního rohu sledovaného objektu
- Y-ová souřadnice levého horního rohu sledovaného objektu
- Šířka sledovaného objektu
- Výška sledovaného objektu

Záznamy jsou odděleny mezerou (případně jiným bílým znakem) a každý záznam je ukončen přechodem na nový řádek. U záznamů se očekává uspořádání podle času.

Samotná metoda provádění testů je založena na porovnávání záznamů uvedených v anotačním souboru s pozicemi, které jsou získány programem. Výsledkem porovnávání jsou

hodnoty true positive (pozice objektu získaná programem se shoduje se záznamem v anotaci) a false positive (pozice objektu získaná programem se neshoduje se záznamem v anotaci). Určení zda je výsledkem true nebo false positiv je dáno procentuální shodou mezi anotačním záznamem a pozicí získanou programem. Pokud je průnik pozic větší než 50% obsahu plochy kterou zabírá objekt, pak je výsledek true positive. V opečném případě je výsledek false positive.

Výstupem programu v testovacím režimu je záznam obsahující údaje, které mají tvar:

id čas : x y w h - > x y w h - > p - > true/false positive

s významem stejným jako u vstupu, kde navíc první čtveřice $x y w h$ označují pozice získané anotacemi a druhá čtveřice pozice získané programem, p je shoda vyjádřená procenty pro daný záznam. Na konci výstupu je uveden souhrn údajů ve tvaru, který shrnuje celkový počet true a false positive výsledků pro každý sledovaný objekt. Ukázku vstupních anotací a výstupů testů lze nalézt na příloženém DVD.

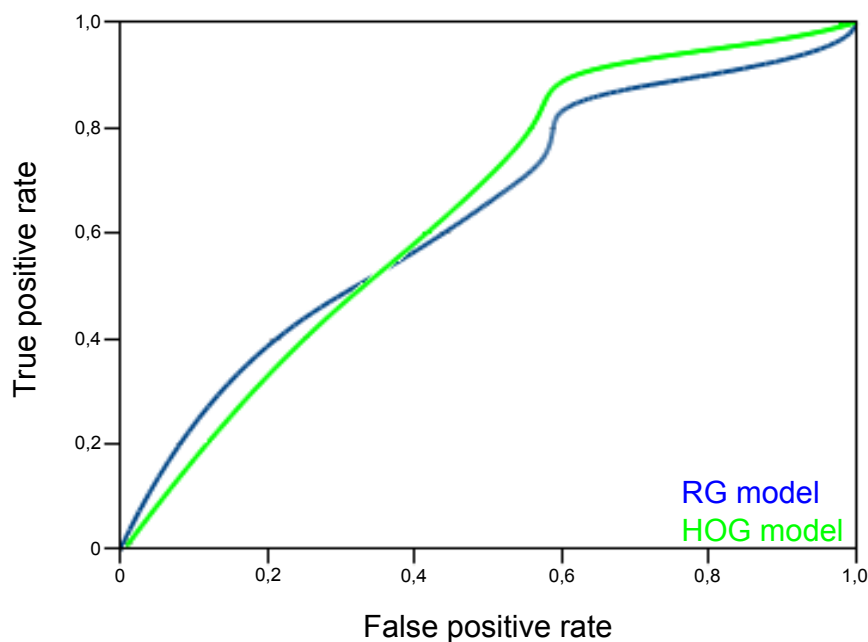
5.2 Vyhodnocení testů

Samotné testování programu bylo rozděleno do dvou skupin testů. V první skupině testů se sledovaly pozice hlav osob ve třech různých video souborech, přičemž v každém videu se vyskytovaly dvě až tři osoby. Délka sledování osob se pohybovala od 1 minuty 43 sekund až po 5 minut 28 sekund. V této skupině testů byly nastaveny parametry sledovacího algoritmu na výchozí hodnoty programu tj. počet částic 100, počet binů 16 a rozdělení objektů na oblasti o velikosti 5×5 pixelů. Tento test byl proveden pro dva modely reprezentace objektů RG model a HOG model. Výsledky testů pro jednotlivá videa a osoby jsou uvedeny v tabulce 5.1.

Číslo testu	Video	Osoba	TP (RG)	FP(RG)	TP(HOG)	FP (HOG)
1	s0_ss2_r2_c1.avi	0	73	3	54	22
2	s0_ss2_r2_c1.avi	1	75	1	22	54
3	s0_ss2_r2_c1.avi	2	86	55	22	119
4	s1_ss1_r1_c1.avi	0	74	272	196	150
5	s1_ss1_r1_c1.avi	1	224	184	214	194
6	s1_ss2_r2_c1.avi	0	389	80	307	162
7	s1_ss2_r2_c1.avi	1	74	13	69	18
8	s1_ss2_r2_c1.avi	2	143	89	40	192
Celkem			1138	697	924	911

Tabulka 5.1: Výsledky testů pro výchozí nastavení programu.

Ze získaných hodnot byla sestavena ROC křivka (obrázek 5.1), která ukazuje výsledky pro použité modely reprezentace. Jak je vidět se zvyšujícím se počtem sledovaných záznamů má lepší úspěšnost HOG model.



Obrázek 5.1: ROC křivka.

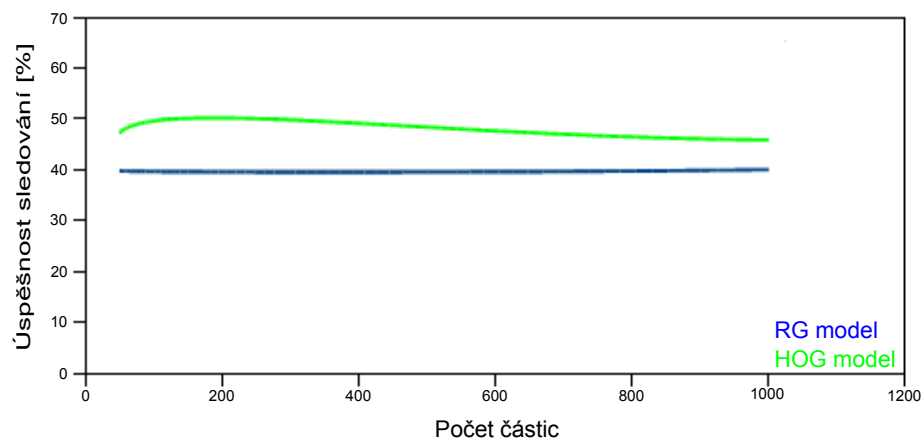
V druhé skupině testů se sledovaly objekty, kterými byly opět osoby. Na rozdíl od první skupiny testů se zkoumal v této sadě testů sledovací algoritmus pro různé nastavení jeho vlastností. Těmito sledovanými vlastnostmi byly:

- Použití různých modelů reprezentace (RG barevný model a HOG model)
- Nastavení různého počtu částic
- Nastavení různého počtu binů histogramů
- Rozdělení sledovaných objektů ne menší oblasti o různé velikosti

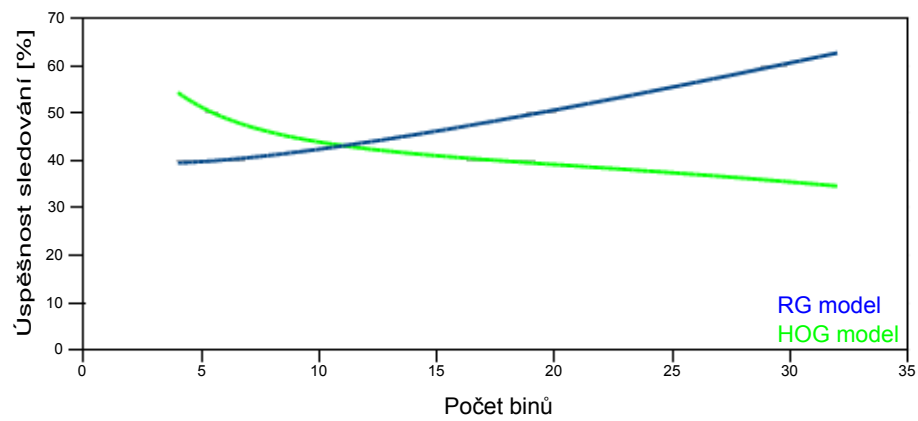
Tento test byl proveden na videu s názvem *s1_ss1_r1_c1.avi* v celkovém počtu 24 testů na 754 vzorcích (tj. záznamů v anotačním souboru) pro každý test. V uvedeném videu byly sledovány dvě osoby, první z nich pod dobu 4 minuty 44 sekund, druhá po dobu 5 minut 28 sekund. Hodnoty true/false positive v tabulkách jsou uvedeny v součtu pro obě osoby. Získané výsledky jsou shrnuty v tabulkách 5.2, 5.3, 5.4 a grafech 5.2, 5.3, 5.4. Grafy byly vytvořeny z data uvedených v tabulkách. Každý graf ukazuje úspěšnost sledování uvedenou v %, přičemž tato hodnota byla získána jako počet true positive výsledků vyděleno celkovým počtem záznamů v anotaci.

Z uvedených grafů je patrné chování částicového filtru implementovaného v programu track pro různé vstupní hodnoty. Jestliže se zvyšuje počet částic, účinnost sledování algoritmu se mění jen nepatrně. V případě změny počtu binů a velikosti oblastí, na které je rozdělen sledovaný objekt se úspěšnost sledování mění výrazněji. Obě tyto nastavení souvisí s reprezentací objektů, přičemž různá nastavení způsobí pro daný model jinou změnu.

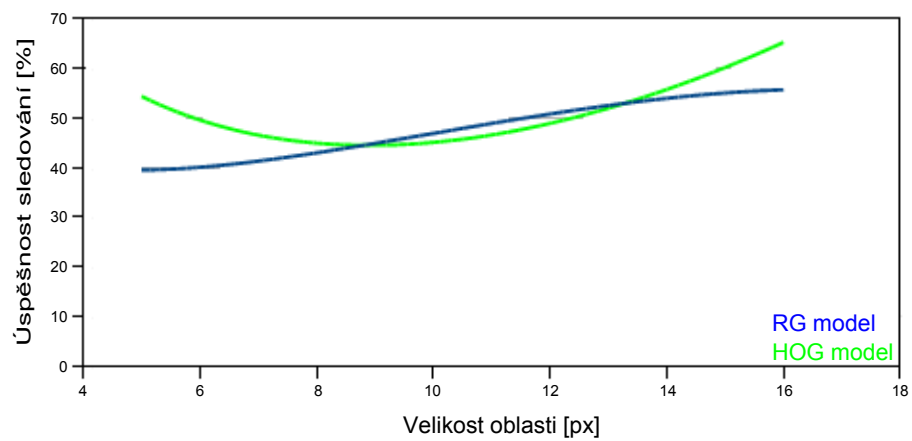
Úspěšnost sledovacího algoritmu lze tedy zvýšit vhodným nastavením parametrů souvisejících s danou reprezentací.



Obrázek 5.2: Úspěšnost částicového filtru v závislosti na počtu částic.



Obrázek 5.3: Úspěšnost částicového filtru v závislosti na počtu binů.



Obrázek 5.4: Úspěšnost částicového filtru v závislosti na velikosti oblastí.

číslo testu	model	počet částic	počet binů	rozdělení na oblasti	TP	FP
1	RG model	50	4	5×5	300	454
2	RG model	100	4	5×5	298	456
3	RG model	500	4	5×5	296	458
4	RG model	1000	4	5×5	302	452
5	HOG model	50	4	5×5	356	398
6	HOG model	100	4	5×5	410	344
7	HOG model	500	4	5×5	350	404
8	HOG model	1000	4	5×5	346	408

Tabulka 5.2: Výsledky testu počet částic.

číslo testu	model	počet částic	počet binů	rozdělení na oblasti	TP	FP
9	RG model	100	4	5×5	298	456
10	RG model	100	8	5×5	303	451
11	RG model	100	16	5×5	345	409
12	RG model	100	32	5×5	473	281
13	HOG model	100	4	5×5	410	344
14	HOG model	100	8	5×5	300	454
15	HOG model	100	16	5×5	315	439
16	HOG model	100	32	5×5	261	493

Tabulka 5.3: Výsledky testu počet binů.

číslo testu	model	počet částic	počet binů	rozdělení na oblasti	TP	FP
17	RG model	100	4	5×5	298	456
18	RG model	100	4	8×8	301	453
19	RG model	100	4	12×12	408	346
20	RG model	100	4	16×16	420	334
21	HOG model	100	4	5×5	410	344
22	HOG model	100	4	8×8	279	475
23	HOG model	100	4	12×12	330	424
24	HOG model	100	4	16×16	492	262

Tabulka 5.4: Výsledky testu velikost oblastí.

Kapitola 6

Závěr

Cílem této bakalářské práce bylo seznámit se s vybranými detekčními a sledovacími metodami a dále ověřit činnost částicového filtru v konkrétní aplikaci a porovnat dosažené výsledky pro různé vstupní hodnoty pomocí testování. Tento cíl se podařilo splnit v podobě a rozsahu popsaném v kapitolách jedna až pět.

V úvodní, první kapitole, jsou uvedeny postupy vedoucí k detekci objektů ve videu a dále jsou zde představeny některé detekční metody. V návaznosti na detekci objektů jsou v druhé kapitole představeny vybrané sledovací metody s popisem jejich využití. Teoretickou část práce uzavírá třetí kapitola věnující se částicovému filtru.

Ve čtvrté kapitole je uveden návrh a postup při vytváření konkrétního programu s názvem track, který umožňuje sledování objektů. Tento program používá ke sledování metodu částicového filtru pro dvě různé reprezentace objektů založené na informacích o barvě a informacích o tvaru objektu. Výsledný kód programu je umístěn na přiloženém DVD a je možné jej použít k vyzkoušení. Celková úspěšnost sledovacího algoritmu byla ověřena sadou testů, jejichž výsledky jsou uvedeny v páté kapitole. Výsledky některých testů poukázaly na vlastnosti vytvořeného programu a na možnosti dalšího zlepšení a vývoje programu. Provedením testů na implementovaných modelech reprezentace objektů byla zjištěna závislost úspěšnosti sledování na vhodné volbě reprezentace a nastavení parametrů této reprezentace.

Do budoucna lze počítat s dalším vývojem a zlepšováním sledovacího algoritmu a programu, který je popsán v kapitole 5. Tento program by mohl být doplněn o automatickou detekci objektů některou z metod uvedených v kapitole 1. Automatická detekce objektů by umožňovala širší použití programu v různých systémech, které by tak nebyly závislé na ručním označení objektů, které se mají sledovat.

Další navrhované vylepšení programu je zaměřeno na přidání adaptivního přizpůsobování se objektům během sledování. V současnosti se v programu track používá model reprezentace bez adaptace. V tomto případě se předpokládá, že změny barev a tvaru objektů jsou omezené, nebo že jsou tyto informace dopředu známy. Ve většině reálných případech je tomu ovšem právě naopak a výrazné změny velikosti, tvaru, natočení nebo jasů znemožní úspěšné sledování objektů programem.

Jedno z navrhovaných vylepšení programu se týká také použití dalšího modelu pro reprezentaci objektů. Současně používané reprezentace jsou založeny na jednoduchých informacích o jasů a tvaru objektu, jejichž přesnost s výraznými změnami tvaru a barvy objektů klesá. Použití některých reprezentací založených na klasifikaci a detekci významných bodů by přineslo celkové zlepšení sledování. Mezi takovéto reprezentace patří např. SIFT [15].

Budoucí vývoj taktéž počítá s vytvořením uživatelského rozhraní, které by přineslo přehlednější a více intuitivní ovládání s možností interaktivního nastavení řady parametrů během sledování. Toto grafické uživatelské rozhraní by mělo také obsahovat možnosti grafického zpracování získaných výsledků například v podobě grafů, porovnávání trajektorií, zobrazování úspěšnosti sledování, apod.

Literatura

- [1] B.Sugandi, H.Kim, J.K.Tan, S.Ishikawa: *A moving object tracking based on color information employing a particle filter algorithm*. Artif Life Robotics (2009), Volume 14, Number 1, 39-42, DOI: 10.1007/s10015-009-0718-6.
- [2] G.Welch, G.Bishop: *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA, 2006, University of North Carolina at Chapel Hill.
- [3] V.Vezhnevets, V.Sazonov, A.Andreeva: *A Survey on Pixel-Based Skin Color Detection Techniques*. In Proceedings of the GraphiCon 2003 (2003), pp. 85-92.
- [4] M.S.Arulampalam, S.Maskell, N.Gordon, T.Clapp: *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*. IEEE Transactions on Signal Processing 50(2): 174-188, 2002. DOI:10.1109/78.978374
- [5] M.Zeller: *A Visual Computing Environment for Very Large Scale Biomolecular Modeling*. Proc. IEEE Int. Conf. on Applicationspecific Systems, Architectures and Processors (ASAP), Zurich, pp. 3- 12, 1997.
- [6] P.Pérez, C.Hue, J.Vermaak, M.Gangnet: *Color-Based Probabilistic Tracking*. ECCV, 2002.
- [7] N.Dalal, B.Triggs: *Histograms of Oriented Gradients for Human Detection*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
- [8] M.Sonka, V.Hlavac, R.Boyle: *Image Processing, Analysis, and Machine Vision*. Thomson Learning, 2008, ISBN 10:0-495-08252-X, 13:978-0-495-08252-1.
- [9] T.Svoboda, J.Kybic, V.Hlavac: *Image Processing, Analysis, and Machine Vision A MATLAB Companion*. Thomson Learning, 2008, ISBN 0-495-29595-7.
- [10] W.Guo, Ch.Han, M.Lei: *Improved Unscented Particle Filter for Nonlinear Bayesian Estimation*. 10th International Conference on Information Fusion: 1-6, 2007, ISBN: 978-0-662-45804-3
- [11] G.Bradsy, A.Kaehler: *Learning OpenCV*. O'Reilly Media, Inc., 2008, ISBN 978-0-596-51613-0.
- [12] A.Yilmaz, O.Javed, M.Shah: *Object tracking: A Survey*. ACM Computing Surveys, ročník 38, č.4, 2006: s.1-45, ISSN 0360030.
- [13] K.Nummiaro, E.Koller-Meier, L.V.Gool: *Object Tracking with an Adaptive Color-Based Particle Filter*. ECCV, 2002, Volume 2449/2002, 353-360, DOI: 10.1007/3-540-45783-643.

- [14] W.Hongling, Y.Bo, T.Guodong, M.Aidong: *Object tracking by applying mean-shift algorithm into particle filtering*. 2nd IEEE International Conference on Broadband Network Multimedia Technology, 2009. IC-BNMT '09.
- [15] P.Wu, L.Kong, F.Zhao, X.Li: *Particle Filter Tracking Based on Color and SIFT Features*. International Conference on Audio, Language and Image Processing, 2008.
- [16] P.Viola, M. J. Jones: *Rapid Object Detection using a Boosted Cascade of Simple Features*. IEEE CVPR, 2001.
- [17] O.Javed, M.Shah: *Tracking and Object Classification for Automated Surveillance*. ECCV, 2006, Volume 2353/2006, 439-443, DOI: 10.1007/3-540-47979-1 23.
- [18] P.Garg, N.Aggarwal, S.Sofat: *Vision Based Hand Gesture Recognition*. World Academy of Science, Engineering and Technology 49, 2009.
- [19] Z.Kalal, J.Matas, K.Mikolajczyk: *Weighted Sampling for Large-Scale Boosting*. British Machine Vision Conference, 2008.