

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020

Bc. Jakub Příbyl



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

SLEDOVÁNÍ OBJEKTU VE VIDEOSEKVENCI POMOCÍ INTEGRÁLNÍHO HISTOGRAMU

OBJECT TRACKING IN VIDEO SEQUENCE USING THE INTEGRAL HISTOGRAM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Příbyl

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Dorazil

BRNO 2020



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Jakub Příbyl

ID: 174384

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Sledování objektu ve videosekvenci pomocí integrálního histogramu

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte principy integrálního histogramu a jeho použití při zpracování obrazu. Optimalizujte dodanou implementaci integrálního histogramu pro práci v reálném čase a použijte ji pro vytvoření aplikace, která dokáže ve videosekvenci sledovat určený objekt. Výsledkem diplomové práce bude aplikace, která během přehrávání videosekvence umožní uživateli přehrávání pozastavit, vybrat objekt, který chce sledovat a opět přehrávání spustit. Aplikace pak v každém následujícím snímku videosekvence vyznačí, kde se sledovaný objekt právě nachází. Výsledná implementace musí řešit také situaci, kdy se vzdálenost objektu od kamery během sledování mění. Nutná omezení pro výběr sledovaného objektu jsou dána použitým mechanismem sledování (integrální histogram).

DOPORUČENÁ LITERATURA:

[1] GONZALEZ, Rafael C. a Richard E. WOODS. Digital image processing. Reading: Addison-Wesley Publishing Company, c1992. World student series. ISBN 0-201-60078-1.

[2] PORIKLI, F. Integral histogram: a fast way to extract histograms in Cartesian spaces. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) [online]. IEEE, 2005, 829-836 vol. 1 [cit. 2017-09-13]. DOI: 10.1109/CVPR.2005.188. ISBN 0-7695-2372-2. Dostupné z: <http://ieeexplore.ieee.org/document/1467353/>

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

Vedoucí práce: Ing. Jan Dorazil

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se věnuje problematice sledování objektu v reálném čase. Sledovaný objekt je definován oblastí ohraničujícího obdélníku. V práci je popsána problematika zpracování obrazu a využití histogramu pro sledování objektu v reálném čase.

Hlavním přínosem práce je rozšíření poskytnutého programu o sledování objektu v reálném čase s měnící se velikostí ohraničujícího obdélníku. Dále je analyzováno chování detekce v různých scénářích. K tomu byly otestovány i různé výpočty vah. Program je napsán v C++ s využitím knihovny OpenCV.

KLÍČOVÁ SLOVA

Sledování objektu, Detekce objektu, Histogram, Integrovaný Histogram, OpenCV, C++

ABSTRACT

This thesis focuses on object tracking in real-time. Tracked object is defined by bounding rectangle. The thesis works on issue of image processing and using histogram for real-time object tracking.

The main contribution of the work is the extension of the provided program to track object in real-time with changing bounding rectangle. Size of the rectangle is changing as the object moves closer or further from camera. Furthermore the detection behavior in different scenarios is analyzed. In addition, various weight calculations were tested. The program is written in C++ using OpenCV library.

KEYWORDS

Object tracking, Object detection, Histogram, Integral Histogram, OpenCV, C++

PŘIBYL, Jakub. *Sledování objektu ve videosekvenci pomocí integrovaného histogramu*. Brno, 2020, 55 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Jan Dorazil

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Sledování objektu ve videosekvenci pomocí integrálního histogramu“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Janu Dorazilovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora

Obsah

Úvod	10
1 Zpracování obrazu a sledování objektu	11
1.1 Reprezentace objektu	13
1.2 Výběr příznaků pro sledování objektu	15
1.3 Detekce objektu	16
1.4 Sledování objektu	17
1.5 Shrnutí	18
2 Způsoby a metody sledování objektů	19
2.1 Bodové sledování	19
2.1.1 Deterministická metoda	19
2.1.2 Statistická metoda	21
2.2 Sledování s jádrem	24
2.2.1 Modely založené na šablonách	25
2.2.2 Sledování s více pohledy	27
2.3 Sledování siluet	27
2.3.1 Shoda tvarů	28
2.3.2 Sledování obrysu	29
2.4 Problémy při sledování	29
2.4.1 Řešení okluze	29
2.4.2 Sledování z více kamer	30
2.5 Shrnutí	30
3 Integrální histogram	32
3.1 Barevný histogram	32
3.2 Vyhledávání v obraze	34
3.3 Matematická formulace integrálního histogramu	36
3.4 Metody procházení oblastí	36
4 Řešení práce	39
4.1 Visual Studio a OpenCV	39
4.2 Testy hypotéz pro porovnání histogramů	40
4.3 Sledování v reálném čase	44
4.3.1 Klouzavé okno s přizpůsobením oblasti	44
4.3.2 Výpočet finální váhy a určení polohy objektu.	45
4.3.3 Shrnutí	46

5 Závěr	47
Literatura	48
Seznam symbolů, veličin a zkratk	55

Seznam obrázků

1.1	Základní diagram sledování objektu.	11
1.2	Reprezentace objektu.	13
1.3	Rozdělení metod pro sledování.	18
2.1	Bodová korelace	20
2.2	Různá omezení pohybu	21
2.3	Sledování pomocí jádra	25
2.4	Iterace středního posunu pomocí více jader	26
2.5	Sledování siluet	28
3.1	Ukázka histogramu.	32
3.2	Barevný histogram snímku.	33
3.3	Pokročilé funkce.	35
3.4	Řetězcová metoda prohledávání obrazu	37
3.5	Prohledávání obrazu pomocí vlnových ploch.	37
4.1	Struktura OpenCV	39
4.2	Šablony pro test výpočtů.	42
4.3	Šablony pro test výpočtů.	43
4.4	Prohledávání pomocí klouzavého okna	45
4.5	Sledování objektu v reálném čase	46

Seznam tabulek

1.1	Jednotlivé metody detekce objektů.	16
2.1	Porovnání sledování.	31
4.1	Tabulka vypočtených vah.	43

Úvod

Sledování a detekce objektů je velmi aktuální a rozšířené odvětví v informatice. Cílem je získávání informací z určitého snímku nebo videa a pomocí těchto informací je poté možné sledovat rozdíly mezi snímky nebo sledovat objekt ve videu. Využití této technologie je například pro sledování změn ve videu, interakcí mezi počítačem a člověkem, ovládání robota apod. Největším problémem tohoto odvětví je jeho výpočetní náročnost a komplexnost. Pokud je využito přesnějších metod jsou většinou velmi náročné na výpočetní techniku a naopak. Je tedy nutné aplikovat správný algoritmus pro specifické situace.

Cílem diplomové práce je prozkoumat možnosti integrálního histogramu a implementace, která byla poskytnuta vedoucím práce a tuto práci rozšířit o sledování objektů v reálném čase s proměnlivou velikostí ohraničujícího obdélníku. Poskytnutý program¹ je OpenCV implementací integrálního histogramu z práce [1]. Program má za úkol detekovat změny pomocí LTP texturálních deskriptorů. Sledování objektu pomocí histogramu má výhodu v nízké výpočetní náročnosti, ale musí se zohlednit situace, ve které se má nasadit.

V rámci této práce byl zprovozněn poskytnutý program a knihovna OpenCV. Praktická část práce byla zhotovena ve vývojovém prostředí Visual Studio s knihovnou OpenCV pro práci s maticemi, výpočtem a porovnání histogramů. V teoretické části byla popsána problematika sledování objektů, různé metody a popsána byla i problematika histogramů.

Hlavním přínosem práce je rozšíření existujícího řešení detekce rozdílů histogramů. Rozšíření spočívalo v aplikaci sledování objektu ve videosekvenci. Za úkol bylo i to, že s měnící se velikostí objektu se mění i jeho ohraničující obdélník. Dále byly analyzovány různé scénáře s metodami výpočtů vah.

Práce je rozdělena do čtyř kapitol. V kapitole 1 je popsán celkový proces pro sledování objektů a jeho dílčí kroky. V kapitole 2 jsou více rozebrány a popsány jednotlivé způsoby a konkrétní metody pro sledování objektů ve videu. Ke konci kapitoly jsou rozebrány i problémy, které se často řeší. V kapitole 3 jsou popsány histogramy a konkrétní řešení detekce pomocí integrálního histogramu [1]. Poslední kapitola 4 je věnována praktické části této práce.

¹<https://github.com/deu439/Integral-Histograms>

1 Zpracování obrazu a sledování objektu

Sledování objektů v reálném čase je definováno jako proces odhadování trajektorie nebo cesty objektu v po sobě jdoucích snímcích. Hlavním cílem sledování objektu je proces segmentace objektu nebo více objektů z videa a sledování jejich pohybu, okluze a orientace. Sledování objektu zájmu pomocí technik zpracování obrazu je náročným a zajímavým odvětvím. Je využito v mnoha aplikacích jako např. řízení provozu [2], navigace robota [3], sledování dopravy [4], identifikace objektů [5] apod. Zpracování obrazu je aktuálně velmi rozšířenou výzkumnou oblastí, do kterého patří i sledování objektů v reálném čase, např. sledování osob a monitorování. To je však spojeno s velkým počtem problémů v jednotlivých krocích (viz 1.1) [6].



Obr. 1.1: Základní kroky při sledování objektu.

Pro analýzu videa jsou hlavní tyto tři kroky: detekce pohybujících se objektů, klasifikace objektů pomocí jejich chování nebo vzhledu a sledování těchto objektů mezi snímky. Sledování objektů je tedy využitelné v těchto oblastech:

- **Rozpoznání na základě pohybu** - Detekce člověka při pohybu, automatická detekce objektů apod. V práci [7] je zajištěna detekce pomocí binárního vzoru, která zajišťuje větší odolnost proti změnám osvětlení scény.
- **Automatizovaný dohled** - Monitorování scény pro detekci podezřelých aktivit a událostí. V [8] jsou detekovány objekty ze snímků s využitím učení. V praxi tato metoda byla použita na detekci pracovníků a strojů na stavbě z databáze pořízených fotografií.
- **Indexování Videá** - Automatická anotace a načítání videí v multimediálních databázích. Jako příklad lze uvést práci [9], kde je řešena problematika nalezení specifické části videa z databáze mnoha videí na dané téma.
- **Interakce člověk-počítač** - Rozpoznávání gest, sledování zrakových pohledů pro vstupní data do počítače apod. Gesta jsou použita například při ovládní herního zařízení Kinect nebo při virtuální realitě, jak lze vidět v práci [10], kde byl využit Kalmánův filtr k rozpoznání gest.
- **Sledování provozu** - Shromažďování dopravních statistik pro řízení provozu. Studie [11] automatizuje sběr dat pomocí sledovacích technik, které pomáhají k implementaci přesnějšího a spolehlivějšího softwaru.
- **Navigace vozidla** - Plánování trasy na základě videa a vyhýbání se překážkám. Použití sledování několika objektů v souvislosti s autonomním řízením vozidla je prezentováno v [12].

Sledování může být definováno jako problematika odhadu trajektorie pohybujícího se objektu v obraze. Při sledování objektu dochází k označování polohy sledovaného objektu v různých snímcích videa. V závislosti na metodě je při sledování možné zjistit informace o objektu jako orientace, oblast nebo tvar objektu. Sledování objektu je komplexní záležitost z těchto důvodů:

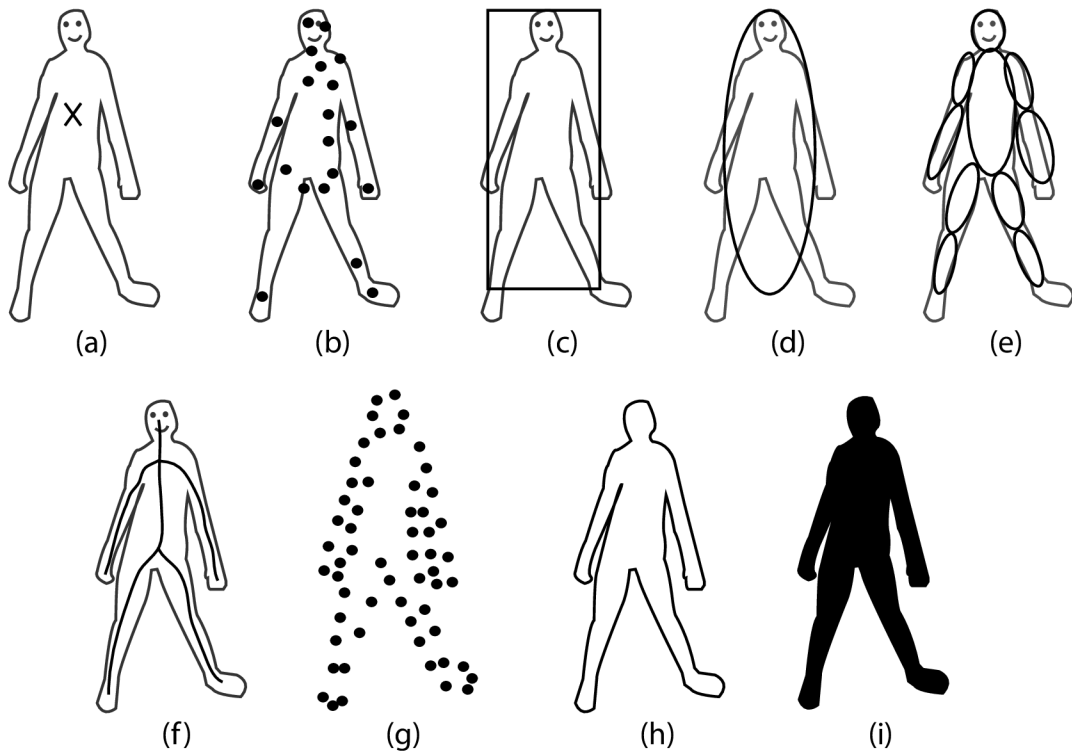
- Ztráta informací při zpracovávání projekce 3D objektu na plochu 2D obrazu,
- šum v obraze,
- složitý pohyb objektu,
- složitý tvar objektu,
- částečná nebo celá okluze,
- změny osvětlení ve scéně,
- požadavky na zpracování v reálném čase.

Ke snazšímu sledování objektů může pomoci např. jednoduchý vzhled objektu bez výrazných změn během sledování, konstantní rychlost pohybu nebo předem daná velikost či počet objektů. Existuje mnoho přístupů pro sledování objektu, které se liší hlavně tím, v jaké oblasti se používají, jaké objekty je potřeba sledovat nebo počet objektů ke sledování. Cílem této kapitoly je zařadit jednotlivé metody do kategorií a popsat metody, které reprezentují tyto kategorie. V sekci 1.1 jsou vysvětleny různé

způsoby reprezentace objektu. V další sekci 1.2 jsou rozebrány různé prvky obrazu jako barva, hrany apod. V sekci 1.3 jsou popsány způsoby detekce objektu ve videu. Dále v sekci 1.4 jsou popsány jednotlivé kategorie sledování. V předposlední sekci 2.4 jsou popsány hlavní problémy, které jsou řešeny při sledování objektu. Na konci kapitoly je shrnutí (viz sekce 2.5).

1.1 Reprezentace objektu

Při implementaci algoritmu pro sledování objektu je třeba zvolit vhodný způsob reprezentace objektu. Objekty jsou reprezentovány pomocí jejich tvarů a vzhledu. Reprezentace pomocí tvaru nebo vzhledu můžou být použity samostatně, ale většinou se používají společně pro větší efektivitu [13].



Obr. 1.2: Reprezentace objektu pomocí různých modelů [13].

- **Bodově** - Objekt je reprezentován jedním umístěným bodem, který je ve středu objektu (1.2(a)), nebo více body (1.2(b)). Použití bodové reprezentace objektu může být problémové při sledování více objektů ve videu, pokud dochází k okluzi mezi těmito objekty. Využívá se spíše pro malé, jednoduché objekty.
- **Jednoduché geometrické útvary** - Tvar objektu je reprezentován jedním z jednoduchých geometrických útvarů (1.2(c) a (d)). Pohyb objektů pro tyto tvary je obvykle modelován rigidní nebo afinní transformací. Je běžné, že části

objektu jsou mimo geometrický útvar popř. pozadí je uvnitř, což negativně ovlivňuje přesnost sledování. Jednoduché geometrické tvary jsou vhodnější pro objekty, které nemění svůj tvar v závislosti na čase.

- **Siluetu a obrys** - Objekt je reprezentován pomocí obrysu (1.2(g) a (h)) či siluetou, tedy oblastí, ohraničenou obrysem. (1.2(i)). Tyto reprezentace objektů jsou vhodné pro sledování komplexních objektů, které mění tvar v čase [14].
- **Kloubové tvarové modely** - Reprezentace objektu se skládá z částí těla, které jsou mezi sebou spojeny klouby. Například lidské tělo má spojenou hlavu, trup, ruce, nohy a chodila pomocí kloubů. Vztah mezi částmi je řízen pomocí pohybových modelů, např. úhel kloubů apod. Tyto části jsou pak zobrazené pomocí elips (1.2(e)).
- **Modely kostry** - Kostru objektu je možné získat aplikováním medial axis transformace na siluetu objektu [15]. Tento model se běžně používá pro reprezentaci a rozpoznání objektu a lze využít i pro reprezentaci rigidních objektů (1.2(f)).

Podobně jako u tvarů je mnoho způsobů, jak reprezentovat vzhledové vlastnosti objektu. Jednotlivé modely tvarů a vzhledů se můžou kombinovat pro sledování [13]. Některé běžné vzhledové reprezentace v souvislosti se sledováním objektů jsou:

- **Hustota pravděpodobnosti vzhledu objektu** - Hustota pravděpodobnosti je funkce, která popisuje relativní pravděpodobnost, že tato náhodná proměnná zaujme danou hodnotu [16]. Použitím vnitřní oblasti snímku určeného tvarovým modelem (např. kontury nebo siluety) lze vypočítat odhad hustoty pravděpodobnosti příznaků odvozených z obrazu. Tímto příznakem může být například barva nebo textura. Odhady hustoty pravděpodobností vzhledu objektu mohou být buď parametrické (Gaussián a směs Gaussiánů), nebo neparametrické (Parzenova okna a histogramy).
- **Šablony** - Šablony jsou tvořeny s použitím jednoduchých geometrických tvarů a siluet. Výhoda šablony je ta, že obsahuje prostorové i vzhledové informace. Do šablon je kódován pouze vzhled objektu generovaný z jednoho pohledu. Tato reprezentace je tedy vhodná pouze pro sledování objektů, jejichž vzhled se během sledování výrazně nemění.
- **Aktivní modely vzhledu** - Jsou generovány modelováním tvaru a vzhledu objektu zároveň. Obecně lze říci, že tvar objektu je definován sadou orientačních bodů, které tvoří tvar objektu. Podobně jako u reprezentace pomocí obrysu, orientační body mohou ležet na krajích objektu nebo uvnitř objektu. Aktivní modely vzhledu vyžadují tréninkovou fázi, během které je tvar i vzhled získán ze sady vzorků pomocí například analýzy hlavních komponent (PCA).
- **Více pohledové modely vzhledu** - Tyto modely jsou kódovány z různých pohledů na objekt. Jeden způsob reprezentující různé pohledy na objekt je

generování podprostoru z daných pohledů. Subprostorové přístupy jako např. PCA a analýza nezávislých složek (ICA) jsou použity pro reprezentaci tvaru i vzhledu [17].

Dalším způsobem jak zjistit různé pohledy na objekt je trénováním sady klasifikátorů (metoda podpůrných vektorů [18] nebo Bayesovské sítě [19]). Limitací je však to, že trénovací sada musí pokrýt všechny variace vzhledu objektu. Obecně lze říci, že existuje silný vztah mezi reprezentací objektu a sledovacím algoritmem. Reprezentace objektu je většinou zvolena podle oblasti aplikace sledování. Pro sledování objektů, které jsou v obraze velmi malé je většinou vhodné použít bodovou reprezentaci. Pro objekty, které spíše připomínají základní geometrické útvary je vhodné použít zase reprezentaci modelů jednoduchých geometrických tvarů. Pro sledování obtížnějších tvarů, např. lidí, je vhodné použít reprezentaci pomocí obrysu a siluety.

1.2 Výběr příznaků pro sledování objektu

Zvolení správného příznaku hraje kritickou roli pro sledování objektů. Nejžádanější vlastností vizuálního příznaku je jeho unikátnost, pomocí které lze objekty v prostředí rozlišit. Výběr funkce je úzce spjat s reprezentací objektu. Například barva je použita jako příznak pro sledování pomocí histogramu a pro sledování s využitím obrysů jsou důležité příznaky hrany objektu. Mnoho sledovacích algoritmů využívá kombinaci těchto příznaků. Běžné vizuální příznaky jsou:

- **Barva** - Barva je jedním z nejjednodušeji aplikovatelných a nejspolehlivějších vizuálních příznaků. Je robustní vůči problémům s pozadím a je nezávislý na velikosti a orientaci snímku. Nejpoužívanější technikou pro získání barev ze snímku je barevný histogram, který ukazuje distribuci barev v obraze. Barevný histogram má dva hlavní problémy a to, že nezohledňuje prostorové informace a není unikátní. Dva jiné snímky mohou mít stejnou barevnou distribuci a tím pádem stejné histogramy. Zároveň mohou vzniknout dva rozdílné histogramy při dvou stejných snímcích s rozdílným nasvětlením scény [20].
- **Hrany** - Rozdíl mezi objektem a pozadím je většinou jednoduše rozpoznatelný. Hrany objektů obvykle generují silné změny v intenzitě obrazu. Důležitou vlastností hran je to, že jsou méně náchylné na změny osvětlení scény než barevné prvky.
- **Optický tok** - Algoritmus optického toku (OF) vypočítává posun vzorů jasu z jednoho snímku na druhý. To se provádí pomocí hodnot intenzity sousedních pixelů. Algoritmy, které vypočítávají posunutí pro všechny obrazové pixely se označují jako algoritmy hustého OF, zatímco algoritmy řídkého OF vypočítávají posunutí pro určitý počet pixelů ve snímku. Výpočet tohoto zjevného

pohybu se provádí identifikací pohybu každého pixelu mezi snímky. Tato metoda je časově náročnější pro detekci složitějšího pohybu objektu a hodí se více pro detekci pohybu více objektů zároveň. [21].

- **Textura** - Textura je změna intenzity povrchu, která využívá prvky jako plynulost a pravidelnost. V porovnání s barvou je vyžadováno předzpracování pro generování deskriptorů. Podobně jako u hran je textura méně náchylná na změny osvětlení scény [6].

Ve většině případů si uživatel vybere, které příznaky chce využít pro svou aplikaci. Dá se však využít i automatická selekce příznaků a to pomocí metody filtrování a metody obtékání. Metody filtrace se snaží zvolit příznak na základě obecného kritéria (např. příznaky by neměly spolu souviset). Metoda obtékání se snaží zvolit příznaky na základě užitečnosti Příznaků v konkrétní problematice (např. klasifikace pomocí podmnožin příznaků). PCA je příkladem metody filtrování k redukci příznaků.

1.3 Detekce objektu

Detekce objektu je část, která detekuje instance objektů třídy v obraze. Každá metoda pro sledování objektů vyžaduje detekci objektu buď v každém snímku, nebo v době, kdy se sledovaný objekt objeví poprvé ve videu. Běžným přístupem je to, že se využívá informace v jednom snímku. Některé metody však využívají dočasné informace vypočtené ze sekvence snímků k redukci špatných detekcí. V tabulce 1.1 jsou uvedené běžně používané metody pro detekci objektu.

Tab. 1.1: Jednotlivé metody detekce objektů.

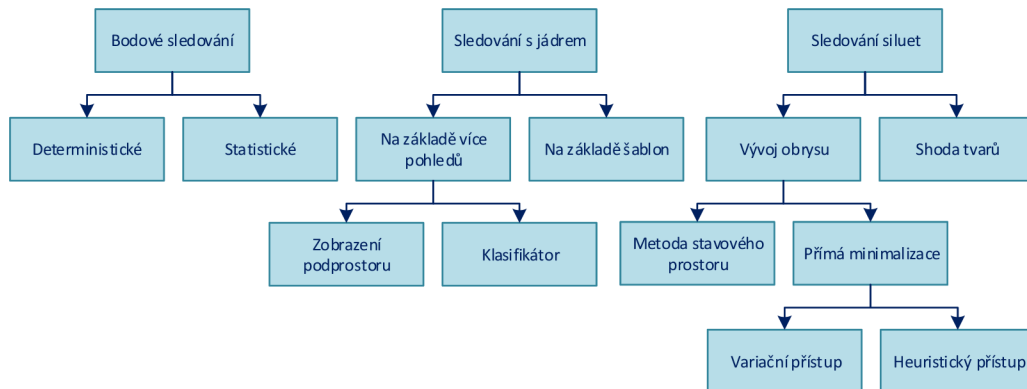
Kategorie	Odpovídající práce
Bodové detektory	Harrisova detektoru rohů [22] SIFT [23]
Segmentace	Střední posun [24] Aktivní obrysy [25]
Odčítání pozadí	Adaptivní modely [26] Dynamické scény [27]
Učení s učitelem	SVM [18] Neuronové sítě [28] AdaBoost [29]

- **Bodové detektory** - Využívají se pro nalezení bodů zájmu v obraze. Bod zájmu je definován jako bod ve snímku, který je lehce detekovatelný. To znamená, že bodem zájmu může být např. konec čáry, bod maxima na křivce nebo roh objektu. Velkou výhodou těchto detektorů je to, že tyto příznaky jsou do značné míry nezávislé na osvětlení a na pohledu kamery [30].
- **Segmentace** - Cílem segmentace obrazu je rozdělit snímek na několik menších homogenních oblastí. Díky tomu dochází ke zjednodušení a změně reprezentace obrazu do pochopitelnější formy, která je lépe analyzovatelná. Segmentace obrazu se obvykle používá k nalezení objektů a hran v obraze. Segmentace obrazu je proces přiřazování označení každému v pixelu v obraze tak, že pixely se stejným označením sdílejí určité charakteristiky [31].
- **Odčítání pozadí** - Odčítání pozadí se používá k oddělení pohyblivého objektu od pozadí. Při používání této metody je nutné vybudovat model pozadí pro danou scénu. K detekci objektu pak dochází jakmile dojde ve scéně ke změně, která není konzistentní s modelem pozadí. Pixely podléhající této změně vyznačují oblast sledovaného objektu. Velkým omezením těchto přístupů je to, že vyžaduje statické pozadí [32].
- **Učení s učitelem** - Učení s učitelem se používá ke klasifikaci a zpracování dat pomocí strojového jazyka. Při učení s učitelem se využívají označená data k odvození učícího algoritmu. Soubor dat se používá jako základ k predikci klasifikace jiných neoznačených dat pomocí strojového učení [33].

1.4 Sledování objektu

Sledování objektu je problematika použití měření ze sensorů k určení polohy, cesty a charakteristik objektu zájmu. Senzorem může být jakékoliv měřící zařízení, které se používá pro shromažďování informací o objektech v prostředí (kamera, radar apod.). Cílem sledování objektu je odhad trajektorie objektu v čase. Toho je docíleno pomocí lokalizace polohy objektu v každém snímku videa. V prvním případě je oblast objektu v každém snímku získána pomocí algoritmu detekce objektu a v každém dalším snímku je sledován odpovídající objekt. V druhém případě je oblast objektu odhadována iterativní aktualizací pozice a oblasti objektu z předchozích snímků. V obou případech sledování jsou objekty reprezentovány tvarem nebo vzhledovým modelem (viz sekce 1.1) [34]. Při sledování objektu existuje celá řada nejistot. Např. může dojít k částečné okluzi objektu, objekt může být špatně detekován (popř. špatný počet, pokud je detekovaných objektů více.) nebo může změnit svůj tvar. Měření ze sensorů mohou podléhat náhodnému šumu. Pokud je více sensorů, je komplexnější a časově náročnější sledovat jeden objekt z více pohledů. Objekty také mohou být

blízko sebe a sledování objektu může změnit cíl sledování. Senzory také velice často poskytují data, bez toho, aniž by byl objekt v obraze. [35]



Obr. 1.3: Rozdělení metod pro sledování [13].

Rozdělení sledovacích metod lze vidět na obrázku 1.3. Dále budou popsány jednotlivé metody a ke každé uvedené stručně příklad.

1.5 Shrnutí

V této kapitole byla obecně popsána problematika spjatá s klasifikací, detekcí a sledováním objektu. V úvodní části kapitoly 1 je stručně popsáno, v čem spočívá zpracování obrazu a sledování objektů. V sekci 1.1 je sepsáno, jakým způsobem se objekt klasifikuje. Klasifikace objektu je důležitá např. v odvětví dolování dat, kde pomocí automatického prohledávání datové sady lze rychle vyhledávat klasifikované objekty. Dále je důležité pro část detekce, kde se může rozlišovat např. mezi osobou a zvířetem. V sekci 1.3 je popsáno, jakým způsobem se detekují objekty v obraze. Pro správně sledování objektu je detekce důležitou součástí kvůli plynulému a bezchybnému sledování. V poslední sekci 1.4 kapitoly je popsáno a rozděleno sledování objektu. Této části je věnovaná další kapitola 2.

2 Způsoby a metody sledování objektů

V této kapitole budou rozebrány různé kategorie pro sledování objektů ve videu a to bodové sledování 2.1, sledování s jádrem 2.2 a sledování siluet 2.3. Jednotlivé kategorie jsou poté dále rozděleny na jednotlivé metody, které jsou popsány.

2.1 Bodové sledování

Detekce objektu v po sobě jdoucích snímcích je reprezentována body a asociace těchto bodů je založena na předešlých stavech objektu, které zahrnují jeho pozici a pohyb. Určení korelujících bodů mezi snímky je komplikovaný problém především z důvodu okluze a chybné detekce. Obecně lze rozdělit tyto metody do dvou skupin - deterministické a statistické metody. Deterministické metody používají kvalitativní pohybovou heuristiku k omezení problémů korelace. Na druhou stranu, statistické metody modelují nejistoty měření a berou je v potaz při vytváření korelace. Tato korelace je definovaná jako pravděpodobnost asociací objektu ve snímku $t - 1$ k specifickému objektu ve snímku t .

Metody sledování bodů lze porovnat na základě vytváření správné trajektorie bodů. Vzhledem k ground truth může být výkon vyhodnocen výpočtem přesnosti a odvolání. U bodového sledování mohou být definovány takto:

$$\text{přesnost} = \frac{SK}{VK}, \quad (2.1)$$

$$\text{odvolání} = \frac{SK}{AK}, \quad (2.2)$$

kde SK je počet správných korelací, VK je počet vytvořených korelací a AK je počet aktuálních korelací. Aktuální korelace označuje korelaci dostupnou v ground truth. Kvalitau sledovacích metod lze porovnat i podle těchto vlastností:

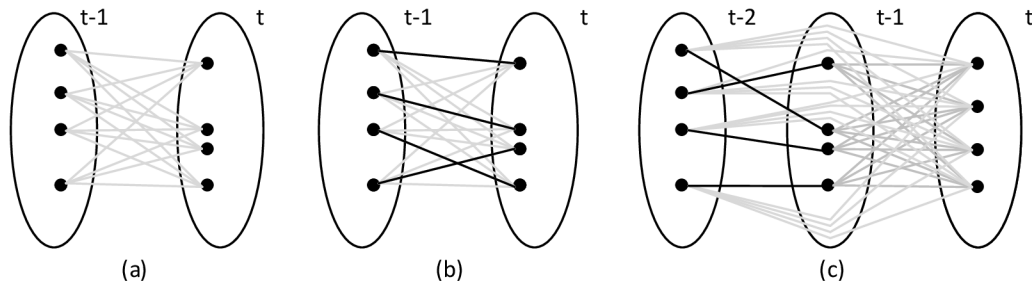
- Schopnost vypořádání se s novými objekty a objekty opouštějící scénu.
- Zacházení s okluzí.
- Vytvoření optimálního řešení problému výpočetní náročnosti funkcí pro stanovení korelace.

Bodové sledování je vhodné pro sledování velmi malých objektů reprezentovaných jedním bodem. Ke sledování větších objektů je nutno použít více bodů.

2.1.1 Deterministická metoda

U bodového sledování tato metoda pracuje na propojení každého objektu v předešlém snímku s jediným objektem v aktuálním snímku. Deterministická metoda pro korelaci bodů definuje cenu asociace každého objektu v čase $t - 1$ jedinému objektu

v čase t pomocí sady pohybových omezení. Minimalizace korelace je formulováno jako problém kombinatorické optimalizace. Řešení vzájemných korelací (obr. 2.1(a)) mezi všemi možnými asociacemi (obr. 2.1(b)) lze získat pomocí optimálních metod přiřazování (např. [36]).

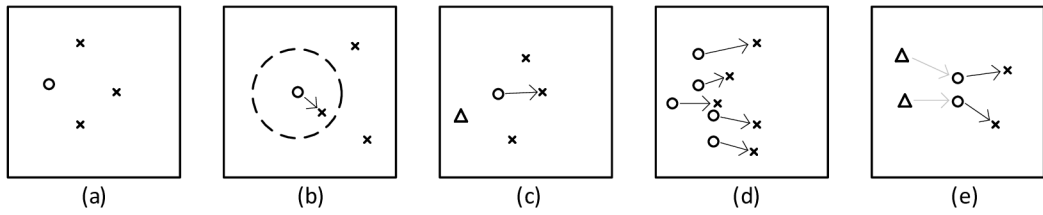


Obr. 2.1: Bodová korelace. (a) Všechny možné asociace bodu (objektu) ve snímku $t - 1$ s body (objekty) ve snímku t , (b) jedinečná sada asociací vykreslená čarami, (c) vícesnímková korelace [13].

Omezení hodnoty korelace je definováno pomocí kombinace následujících omezení:

- **Podobnost** předpokládá, že objekt nezmění razantně polohu mezi snímky (viz 2.2(a)).
- **Omezení rychlosti** definuje hranici rychlosti objektu a vymezí kruhovou oblast, ve které se nachází objekt (viz 2.2(b)).
- **Plynulost pohybu** předpokládá, že se směr a rychlost objektu drasticky nezmění (viz 2.2(c)).
- **Společný pohyb** omezuje rychlost objektů ve vzájemně blízkosti tak, aby byla podobná. Toto omezení je vhodné pro objekty reprezentované více body (viz 2.2(d)).
- **Rigidita** předpokládá, že objekty ve 3D světě jsou rigidní, proto vzdálenost mezi dvěma body objektu se nezmění (viz 2.2(e)).
- **Podobná jednotvárnost** kombinuje podobnost a omezení rychlosti.

Příkladem deterministické metody je řešení od Shafique a Shah [37], kteří navrhnou víceúčelový přístup k zachování časové koherence rychlosti a polohy (viz 2.2(c)). Formulují korelaci jako teoretický problém grafu. Korelace více snímků slouží k nalezení nejlepší unikátní cesty $P_i = x^0, \dots, x^k$ pro každý bod (horní index je číslo snímku). U chybně detekovaných bude cesta obsahovat chybějící pozice v příslušných snímcích. Graf generovaný pomocí bodů v k snímcích je převeden na bipartitní graf rozdělením každého objektu na dva (+ a -) uzly. Korelace je pak získána pomocí hladového algoritmu.



Obr. 2.2: Různá omezení pohybu. (a) blízkost, (b) omezení rychlosti, (c) plynulost pohybu, (d) společný pohyb, (e) rigidita. Trojúhelník znázorňuje pozici objektu ve snímku $t-2$, kruh označuje snímek v čase $t-1$ a křížek označuje snímek v čase t [13].

2.1.2 Statistická metoda

Není možné se úplně vyhnout chybám jako šumu v měření nebo náhodným chybám v pohybu. Statistická metoda tyto chyby bere v úvahu při odhadu stavu objektu, díky čemuž je robustní. Statistická metoda je však více komplexní než deterministická metoda. Metody statistické korelace používají stavový prostor k modelování vlastnosti objektu jako je poloha, rychlost a zrychlení. Bodové sledování není tolik ovlivněno změnou osvětlení. Vyžaduje však další mechanismy pro detekci objektu v každém snímku. Má také problémy s okluží a není dobré pro rozlišení více objektů od objektu s pozadím.

Informace o objektu ve scéně jsou reprezentovány stavovými hodnotami. Např. poloha objektu je reprezentována jako sekvence stavů $X^t : t = 1, 2, \dots$. Změna stavu za čas je pak dána dynamickou rovnicí

$$X^t = f^t(X^{t-1}) + W^t, \quad (2.3)$$

kde $W^t : t = 1, 2, \dots$ je šum. Vztah mezi měřením a stavem objektu je dán jako rovnice měření $Z^t = h^t(X^t, N^t)$, kde N^t je bílý šum, který je nezávislý na W^t . Cílem sledování je určit stav X^t pomocí všech měření do daného momentu, popřípadě zkonstruovat funkci hustoty pravděpodobnosti (PDF) $p(X^t|Z^{1,\dots,t})$. Teoretické optimální řešení je použití rekurzivního Bayesovského filtru, který řeší problém ve dvou krocích. Krok predikce využívá dynamické rovnice a vypočtené PDF stavu v čase $t-1$ k odvození původního PDF aktuálního stavu, který je $p(X^t|Z^{1,\dots,t-1})$. Krok korekce využívá PDF $p(X^t|Z^t)$ aktuálního měření pro výpočet předešlého PDF $p(X^t|Z^{1,\dots,t-1})$. V případě, že měření vznikají pouze v důsledku přítomnosti jednoho objektu ve scéně může být stav odhadnut těmito dvěma kroky. Pokud se však ve scéně objevuje více objektů, musí být měření spojena s odpovídajícími stavy objektů.

Odhad stavu jednoho objektu

Pokud jsou f^t a h^t lineární funkce, počáteční stav X^1 a šum má Gaussovo rozdělení, pak je odhad optimálního stavu určen Kalmanovým filtrem. V opačném případě je použit filtr částic.

Kalmanův filtr se skládá z kroků predikce a korekce [13].

$$\begin{aligned}\bar{X}^t &= \mathbf{D}X^{t-1} + W, \\ \bar{\Sigma}^t &= \mathbf{D}\Sigma^{t-1}\mathbf{D}^T + Q^t,\end{aligned}$$

kde \bar{X}^t a $\bar{\Sigma}^t$ jsou stav a predikce kovariance v čase t . \mathbf{D} je stav matice přechodu, která definuje vztah mezi stavovými proměnnými v čase t a $t - 1$. Q je kovariance šumu W . Podobně krok korekce využívá aktuální pozorování Z^t k aktualizaci stavu objektu:

$$K^t = \bar{\Sigma}^t \mathbf{M}^T [\mathbf{M} \bar{\Sigma}^t \mathbf{M}^T + R^t]^{-1}, \quad (2.4)$$

$$X^t = \bar{X}^t + K^t \underbrace{[Z^t - \mathbf{M} \bar{X}^t]}_v, \quad (2.5)$$

$$\Sigma^t = \bar{\Sigma}^t - K^t \mathbf{M} \bar{\Sigma}^t,$$

kde v se nazývá inovace, \mathbf{M} je měřicí matice, K je Kalmanův zisk.

Filtr částic se využívá, když stavové proměnné nemají Gaussovo rozložení. Hustota podmíněného stavu $p(X_t|Z_t)$ v čase t je reprezentována sadou vzorků $s_t^{(n)} : n = 1, \dots, N$ (částice) s hmotností $\pi_t^{(n)}$ (vzorkovací pravděpodobnost), kde váhy definují důležitost vzorku. Ke snížení výpočetní složitosti je nutné uložit kumulativní hmotnost $c^{(n)}$ ($c^{(N)}=1$) pro každou n -tici $(n^{(n)}, \pi^{(n)})$. Nové vzorky jsou vytvářeny v čase t z $S_{t-1} = \{(s_{t-1}^{(n)}, \pi_{t-1}^{(n)}, c_{t-1}^{(n)}) : n = 1, \dots, N\}$ pomocí různých schémat odběrů vzorků. Nejběžnějším schématem vzorkování je *vzorkování podle důležitosti*, které funguje následovně:

1. Selekce. Je vybráno N náhodných vzorků $\hat{s}_t^{(n)}$ z \mathbf{S}_{t-1} generováním náhodného čísla $r \in [0, 1]$, nalezením nejmenšího j tak, že $c_{t-1}^{(j)} > r$ a nastavení $\hat{s}_t^{(n)} = s_{t-1}^{(j)}$.
2. Predikce. Pro každý zvolený vzorek $\hat{s}_t^{(n)}$ je generován $\hat{s}_t^{(n)} = f(\hat{s}_t^{(n)}, W_t^{(n)})$, kde $W_t^{(n)}$ je nulová střední Gaussova chyba a f je kladná funkce, tj. $f(s) = s$.
3. Korekce. Váhy $\pi_t^{(n)}$ odpovídající novým vzorkům $s_t^{(n)}$ jsou vypočítány pomocí měření z_t podle $\pi_t^{(n)} = p(z_t|x_t = s_t^{(n)})$, kde p lze modelovat jako Gaussovu hodnotu.

Využitím nových vzorků \mathbf{S}_t lze odhadnout polohu nového objektu pomocí $\varepsilon_t = \sum_{(n=1)}^N \pi_t^{(n)} f(s_t^{(n)}, W)$. Sledovače založené na filtru částic mohou být inicializovány

pomocí prvních měření, $s_0^{(n)} \sim X_0$, s váhou $\pi_0^{(n)} = \frac{1}{N}$ nebo trénováním systému pomocí vzorkovacích sekvencí. Kromě sledování nejlepších částic se obvykle používá další převzorkování k eliminaci vzorků s velmi nízkou váhou [38].

Asociace dat a odhad stavu více objektů.

Při sledování více objektů pomocí Kalmanova filtru nebo filtru částic je potřeba před použitím těchto filtrů vyřešit problém korelace. Nejjednodušší metodou je využití přístupu nejbližšího souseda. Pokud jsou však objekty blízko sebe, je šance, že korelace nebude správná. Špatně asociovaná měření můžou způsobit chybu konvergence. K řešení tohoto problému existuje několik metod pro přiřazování statistických dat (např. JPDAF nebo MHT), které jsou popsány níže.

Filtr asociace dat pravděpodobnosti (JPDAF) využívá trasu, což je sekvence měření, u nichž se předpokládá, že pocházejí ze stejného objektu. Pokud máme N tras a $Z(t) = z_1(t), \dots, z_{m_t}(t)$ jsou m měření v čase t . Je potřeba přiřadit tato měření k existujícím trasám. Předpokládá se, že počet tras zůstává konstantní v čase. $v_{i,l}$ je inovace (viz rovnice 2.5) přiřazena trase l díky měření z_i . JPDAF spojuje všechna měření s každou trasou. Kombinovaná váhová inovace je dána:

$$v^l = \sum_{i=1}^{m_k} \beta_i^l v_{i,l}, \quad (2.6)$$

kde β_i^l je pravděpodobnost, že měření i je vytvořeno z objektu asociovaného k trase l a je dána jako:

$$B_i^l = \sum_{\eta} P[\eta_l(k) | Z^k] \tau_{i,l}(\eta), \quad (2.7)$$

kde $\eta_{i,l}$ je indikátor proměnné, $i = 1, \dots, m_k$ a $l = 1, \dots, N$. $\eta_{i,l}$ se rovná jedné, pokud je měření $z_i(k)$ asociované k trase l , jinak se rovná nule. Velkým omezením JPDAF algoritmu je jeho neschopnost zpracovávat nové objekty, které se objevují ve scéně a sledované objekty, které scénu opouštějí. Dalším problémem je, že data jsou asociovaná pevnému počtu objektů mezi dvěma snímky. Pokud dojde ke změně počtu objektů, dochází k chybám [39].

Sledování více hypotéz (MHT) řeší problém týkající se změny počtu objektů u JPDAF. Pokud je pohybová korelace stanovena pouze pomocí dvou snímků, může vzniknout nesprávná korelace. Lepších výsledků je dosaženo, pokud je korelace odložena, dokud není analyzováno více rámců. MHT algoritmus uchovává několik korelačních hypotéz pro každý objekt v každém časovém rámci. Algoritmus má schopnost vytvářet nové trasy pro objekty vstupující do scény a ukončovat trasy pro objekty opouštějící scénu a díky tomu řeší i okluzy [40].

2.2 Sledování s jádrem

Sledování s jádrem vypočítává pohyb objektu. Pohyb objektu můžeme modelovat pomocí parametrického modelu (translace, konformita, afinita). Při sledování objektu s určeným místem výběru lze pomocí výpočtu pohybu definovat jeho oblast a orientaci v následujícím snímku, protože pro každý bod objektu v aktuálním snímku lze jeho umístění v následujícím snímku určit pomocí odhadovaného pohybového modelu. V závislosti na situaci, při které jsou sledovače použity, může být důležitá jedna z následujících tří charakteristik. Například při analýze chování objektu na základě jeho trajektorie je postačující pouze pohyb. Pro identifikaci objektu je však také důležitá oblast, ve které se nachází. Za účelem vyhodnocení výkonnosti sledovacích zařízení v této kategorii je možné definovat podmínku, na jejímž základě bude zařízení dělat to, co se od něj očekává. Pokud se očekává, že se bude sledovat pouze pohyb, může být při vyhodnocování proveden výpočet míry vzdálenosti mezi odhadovanými a skutečnými parametry pohybu. Příkladem měření vzdálenosti může být úhlová vzdálenost,

$$d = \frac{\mathbf{AB}}{|\mathbf{A}||\mathbf{B}|},$$

kde \mathbf{A} a \mathbf{B} jsou pohybové vektory. Pokud je požadováno, aby sledovače poskytly kromě své trajektorie i správné určení oblasti objektu, lze tyto údaje získat vyhodnocením výpočtu přesnosti a odvolání. Přesnost i míra odvolání jsou označovány jako průsečík předpokládané a správné oblasti objektu. Přesnost je určena poměrem průniku předpokládaných oblastí. Odvolání je poměr průniku k ground truth. Kvalitativní srovnání sledovacích zařízení lze dělit na základě:

- schopnost sledování jednoho nebo více objektů,
- schopnost zvládnout okluzi,
- implementace učení,
- typ pohybového modelu,
- požadavek na ruční inicializaci.

Velmi běžnou reprezentací objektů v reálném čase jsou geometrické tvary (viz 2.3. Vzhledem k omezení rigidity, počítají sledovací metody v této kategorii parametrický pohyb objektu. Pohyb objektu lze odhadnout porovnáním vzhledu objektu mezi předchozím a současným rámcem. Proces odhadu může být ve formě přímého vyhledávání nebo pomocí postupného přibližování a oddalování. Sledování objektu založené na principu přiblížení a oddálení vyžaduje, aby alespoň část objektu byla viditelná uvnitř oblasti, jehož pozice je definovaná pozicí objektu v předešlém snímku. V takovýchto případech se používá Kalmanova filtrace nebo filtrace částic (viz 2.1.2).

Nevýhodou nahrazování objektů jednoduchými geometrickými obrazci je, že může dojít k vynechání částí objektu mimo oblast obrazce, nebo naopak zahrnutí části

pozadí dovnitř. Tyto jevy lze pozorovat jak u rigidních objektů (měnících pouze svoji polohu), tak u nerigidních (u nichž pohybem dochází ke změně vzhledu). V takových případech nemusí být odhad polohy objektu pomocí zvětšení podobnosti modelu správný. K překonání tohoto omezení je jedním z možných řešení zachytit uvnitř objektu jádro, spíše než do něj zahrnout kompletní tvar. Další možností je modelování vzhledu objektu pomocí funkce pravděpodobnosti hustoty barvy nebo textury a přiřazením váhy pixelům nacházejícím se uvnitř jednoduchého obrazce na základě podmíněné pravděpodobnosti pozorované barvy nebo textury.

Metody sledování pomocí jádra se liší reprezentací vzhledu, počtu sledovaných objektů a v použité metodě odhadu pohybu objektu. Tyto metody sledování rozdělujeme do dvou kategorií na základě reprezentace vzhledu objektu, a to na sledování na základě více pohledů, a na sledování s využitím šablon [42].



Obr. 2.3: Sledování pomocí jádra.

2.2.1 Modely založené na šablonách

Modely založené na šablonách prohledávají snímek a hledají šablonu objektu, která je definovaná z předchozího snímku. Vzhledem ke své jednoduchosti a nízké výpočetní náročnosti se používají šablony a modely vzhledu založené na hustotě. Díky šablonám lze sledovat jak jeden objekt, tak i více objektů zároveň [43].

Sledování jednoho objektu

Nejběžnější přístup v této kategorii je shoda šablon. Shoda šablon vyhledává ve snímku I_w pomocí hrubé síly oblast, která je podobná šabloně objektu O_t z předchozího snímku. Pozice šablony v aktuálním snímku je vypočítaná mírou podobnosti, např. metodou křížové korelace:

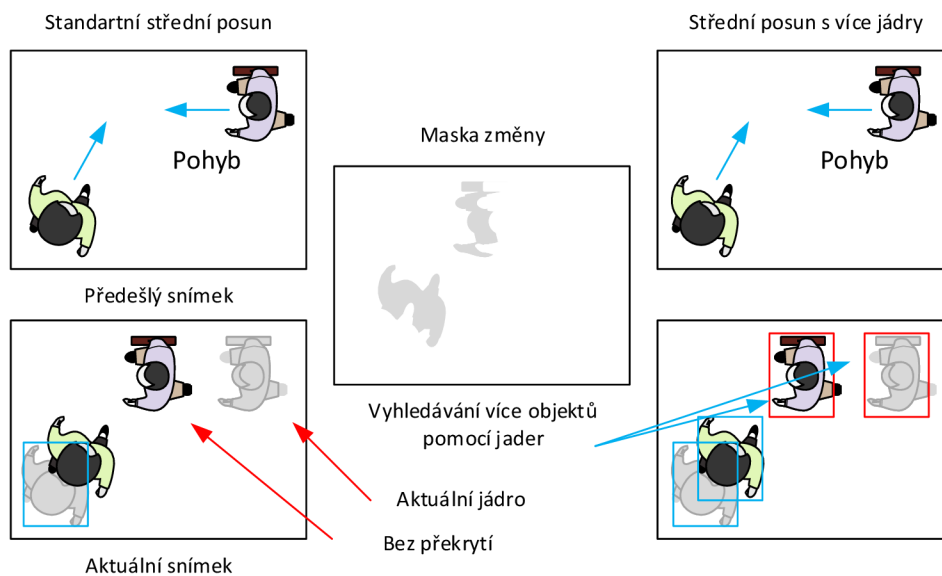
$$\arg \max_{dx, dy} = \frac{\sum_x \sum_y (O_t(x, y) * I_w(x + dx, y + dy))}{\sqrt{\sum_x \sum_y O_t^2(x, y)}}, \quad (2.8)$$

kde (dx, dy) je pozice šablony. Pro vytvoření šablony je možné využít barvu nebo texturu snímku. Zmíněná metoda se používá kvůli své flexibilitě a nekomplikovanému přístupu. Jelikož se využívá hrubé síly pro prohledávání snímku, je tato metoda výpočetně náročná. Ke snížení výpočetní náročnosti se vymezuje oblast vyhledávání podle nalezeného objektu z předešlého snímku.

Kromě šablon se dá využít k reprezentaci objektu např. barevný histogram nebo modely směsí, které lze vypočítat pomocí hodnot pixelů uvnitř čtverce nebo elipsy. V práci [44] jsou generovány objektové modely nalezením střední barvy pixelů uvnitř obdélníkové oblasti. Aby se snížila výpočetní složitost, prohledávají objekt v osmi sousedních oblastech. Podobnost mezi modelem objektu M a hypotetickou pozicí H je vypočtena vyhodnocením poměru barev u M a H . Pozice, která má největší poměr je vybrána jako aktuální pozice objektu.

Sledování více objektů

Modelování objektů individuálně nezohledňuje interakci mezi více objekty ani interakci mezi pozadím a objekty během sledování. Příkladem interakce mezi objekty může být jeden objekt částečně nebo úplně překrývající druhý. Pro vyřešení tohoto problému bylo navrženo několik metod. Jedna z metod je popsána v práci [45], kde je odhadován statistický model pozadí sestavený z více vrstev pomocí Bayesovského mechanismu aktualizace, který je pak porovnán s aktuálním snímkem pro určení pixelů v popředí. Měřením vzdálenosti mezi barvou pixelů a odpovídajícími modely se získá mapa vzdálenosti. Aktuální objekty jsou sledovány několika jádry se středním posunem (viz obr. 2.4) a jejich tvary jsou upravovány pomocí mapy vzdálenosti.



Obr. 2.4: Iterace středního posunu pomocí více jader [45]

2.2.2 Sledování s více pohledy

V předchozích metodách sledování jsou histogramy, šablony, aj. obvykle generovány online - tedy představují informace shromážděné o objektu z nejnovějších pozorování. Objekty mohou vypadat jinak z různých úhlů pohledu a pokud se během sledování výrazně změní pohled na objekt, tak vzhled objektu již nemusí být platný a trasa objektu může být ztracena. Tento problém řeší např. [46], kde jsou prvky z různých pohledů efektivně zpracovány pomocí korelačních filtrů. Poté jsou jednotlivé pohledy sloučeny, aby se získal přesnější výsledek v rámci pravděpodobnostního rámce.

2.3 Sledování siluet

Objekty mohou mít komplexní tvary např. ruce, hlava, ramena apod., které nemohou být popsány pomocí jednoduchých geometrických tvarů. Metody sledování s využitím siluet poskytují přesný popis tvaru těchto objektů. Cílem je najít oblast objektu v každém snímku pomocí objektového modelu vygenerovaného pomocí předchozích rámců. Tento model může být ve formě barevného histogramu, hrany objektu nebo obrysu objektu (viz 2.5b).

Sledování siluet se využívá, když je vyžadováno sledování celé oblasti objektu. V souvislosti se sledováním oblastí jsou přesnost a odvolání definovány jako průsečík předpokládaných a správných objektů. Přesnost je poměr průniku k předpokládané oblasti a odvolání je poměr průniku k původní pozici. Důležité faktory pro rozlišení různých sledovačů siluety jsou použití prvků, práce s okluzí, trénování. Některé algoritmy navíc používají pro sledování pouze informace o hranici siluety, zatímco jiné používají celou oblast uvnitř siluety (viz 2.3). Obecně jsou přístupy oblastí odolnější vůči obrazovému šumu. Kvalitativní srovnání přístupů ke sledování. Nejdůležitější výhodou sledování pomocí siluet je jejich flexibilita při manipulaci s celou řadou tvarů objektů. Siluety mohou být zastoupeny různými způsoby. Nejběžnější reprezentace siluety je ve formě binární indikační funkce, která označuje oblast objektu jedničkami a zbytkové oblasti nulami. U metod založených na konturách je silueta znázorněna explicitně nebo implicitně. Explicitní metoda definuje hranici siluety sadou kontrolních bodů. Implicitní metoda definuje siluetu pomocí funkce definované na mřížce. Nejběžnějším implicitním znázorněním kontury je reprezentace množin.

Sledování pomocí siluet je rozděleno do dvou kategorií a to shoda tvarů a sledování obrysu. Metody shody tvarů hledají siluetu objektu v aktuálním snímku. Oproti tomu sledování obrysu posouvá výchozí obrys do nové pozice v aktuálním snímku buď na základě modelů stavu a prostoru nebo přímou minimalizací energetické funkce [6]. Sledování siluet je flexibilní při reprezentaci objektu, zvládá velkou variaci tvarů a dokonce i komplexní tvary s klouby. Detekce objektu je potřeba jen ve snímku, kdy se objekt objeví poprvé na scéně. Problémy jsou při okluzi.



(a) Objekt zájmu

(b) Obrys objektu

(c) Silueta objektu

Obr. 2.5: Sledování siluet.

2.3.1 Shoda tvarů

Shoda tvarů je podobná metoda jako sledování založené na porovnávání šablon, kde silueta objektu a jeho asociovaný model se hledají v aktuální snímku. Hledání se provádí výpočtem podobnosti objektu s modelem předpokládané siluety objektu zjištěné z předchozího rámce. Model objektu, který je ve formě mapy hran, je znovu inicializován, aby zvládl změny pohledu v každém snímku.

Přístupem ke shodě tvarů je použití určité míry podobnosti k měření míry podobnosti mezi dvěma tvary. Metody měření podobnosti jsou tedy důležitou součástí této metody. Algoritmus pro využití metody tvarů záleží na specifických problémech shod. Tato metoda pracuje podobně jako metoda šablon. Model je generován na základě siluety z předchozího snímku a ta je využita k výpočtu podobnosti v aktuálním snímku. Model objektu je potom ve formě histogramu barvy nebo hran, hrany objektu nebo kombinací těchto modelů.

Dalším přístupem k porovnání tvarů je najít odpovídající siluety detekované ve dvou po sobě jdoucích snímcích. Vytvoření korespondence siluety je podobné jako v sekci 2.1. Hlavní rozdíl mezi porovnáním siluety a porovnáním bodů je reprezentace objektů a použité objektové modely. Siluety využívají prvky pohledu objektu oproti bodovému sledování, kde se používají pouze pohybové prvky. Detekce siluety se obvykle provádí pomocí odčítání pozadí. Jakmile jsou siluety objektu extrahovány, shoda proběhne pomocí výpočtu určité vzdálenosti mezi objektovými modely spojenými s každou siluetou. Objektové modely jsou obvykle ve formě funkcí hustoty (histogramy hrany nebo barvy), hranice siluety (zavřené nebo otevřené objektové kontury) nebo kombinace těchto modelů [48].

2.3.2 Sledování obrysu

Metody sledování obrysů, v kontrastu s metodami porovnávání tvarů, iterativně rozvíjejí vstupní obrys z předchozího snímku do nového obrysu na aktuálním snímku. Tento rozvoj vyžaduje, aby nějaká část objektu na aktuálním snímku překrývala oblast pozice objektu na předchozím snímku. Sledováním rozvíjením obrysu může být provedeno dvěma různými přístupy. První přístup používá modely stavu a prostoru k vymodelování tvaru a pohybu objektu. Druhý přístup přímo rozvíjí obrys minimalizací (obrysové energie?) za použití technik přímé minimalizace, jako je například gradient klesání [49].

Sledování za použití modelu stavu a prostoru

Stav objektu je definován z hlediska tvaru a pohybových parametrů obrysu. Tento stav je aktualizován v každém okamžiku tak, aby maximalizována pravděpodobnost kontury. Pravděpodobnost kontury závisí na předchozím stavu a aktuální pravděpodobnosti, která je obvykle definovaná z hlediska vzdálenosti obrysu od pozorovaných hran.

Sledování přímou minimalizací funkční energie kontury

V kontextu vývoje kontury existuje analogie mezi metodami segmentace diskutovanými v části 4.3 a metodami sledování kontury v této kategorii. Metody segmentace i sledování minimalizují energetickou funkčnost buď chamtivými metodami nebo klesáním. Energie obrysu je definována jako časová informace ve formě buď časového gradientu (optický tok)

2.4 Problémy při sledování

V této sekci jsou popsány problémy spjaté se sledováním objektu v reálných situacích. Mezi dva hlavní problémy patří okluze a sledování jednoho objektu při sledování z více kamer.

2.4.1 Řešení okluze

Okluze může být rozdělena do tří kategorií: vlastní okluze, okluze mezi dvěma objekty a okluze způsobená strukturou pozadí scény. Vlastní okluze se děje, když jedna část objektu překrývá druhou. Tato situace se nejčastěji stává při sledování objektu s klouby (ruce, nohy apod.). Okluze mezi dvěma objekty nastává, když dva objekty, které jsou sledovány, se překrývají. Okluze způsobená pozadím nastává v případě, když objekt v pozadí překrývá sledovaný objekt. Částečnou okluzi objektu ve scéně

je těžké předpovědět, protože je obtížně rozlišit mezi změnou tvaru objektu a okluzí objektu. Běžným přístupem k zvládnutí kompletní okluze během sledování je modelováním pohybu objektu lineárními dynamickými modely nebo nelineární dynamikou a v případě okluze odhadovat polohu objektu, dokud se znovu neobjeví. Pro odhad umístění pohybu objektů lze využít např. Kalmanův filtr a pro nelineární dynamické modely lze použít filtr částic.

Pravděpodobnost okluze se dá také redukovat zvolením vhodné pozice kamery. Menší šance k okluzi dochází, pokud je kamera ve výšce a vidí na scénu z ptačího pohledu. K častější okluzi dochází pokud je kamera v úrovni s objektem, poté je potřeba implementovat mechanismy pro řešení okluzí [50].

2.4.2 Sledování z více kamer

Využívání více kamer pro sledování je důležité hlavně ze dvou důvodů. Prvním důvodem je použití hloubky informace pro sledování a řešení okluze. Druhým důvodem je zvětšení pozorované oblasti, protože není možné, aby jediná kamera sledovala velké oblasti kvůli konečnému zornému poli senzoru. Důležitým problémem při používání více kamer je vztah mezi různými pohledy kamer, které lze ručně definovat, či vypočítat automaticky na základě pozorování objektů pohybujících se v obraze. U sledovacích algoritmů, které vyžadují odhad hloubky je nutný velký výpočetní výkon. Metody zajišťující korelaci mezi nepřekrývajícími se senzory předpokládají, že jsou nehybné, a že jsou k dispozici trasy objektů každé kamery. Výkon těchto algoritmů hodně závisí na tom, jak moc objekty následují zavedené cesty a na vzájemné synchronizaci obrazu z vícero kamer. V případě, že nelze využít časoprostorové omezení, např. u objektů, které se libovolně pohybují v oblasti bez okluze, je možné použít pouze přístup sledování podle rozpoznávání, který využívá vzhled a tvar objektu k rozeznání při znovu objevení na scéně [51].

2.5 Shrnutí

V této kapitole byly probrány jednotlivé metody a kategorie pro sledování objektů. U přístupů ke sledování objektu je nejdůležitější věcí zajistit pohyb objektu. Metody sledování objektu, které jsou v kapitole vysvětlené pomocí nastudované literatury, jsou vhodnými příklady i díky tomu, že jsou snadno implementovatelné. Pro statické pozadí funguje velmi dobře metoda rozlišování rámců, která zajišťuje vysokou přesnost a malý výpočetní výkon. Objekty mohou být kategorizovány podle pohybu, tvaru, barvy a textury. Metody založené na barvách a texturách jsou nejvíce využívány díky vysoké přesnosti. U sledování založeném na obrysech a sledování s využitím jádra je požadováno, aby byl objekt přítomen v prvním snímku. Ten je

poté detekován v každém dalším. Z tab. 2.1 lze vidět, že určité metody je možné využít pouze jen v některých situacích. Obecně lze říci, že čím přesnější je sledování, tím je výpočetně náročnější [6].

Tab. 2.1: Porovnání sledovacích metod.

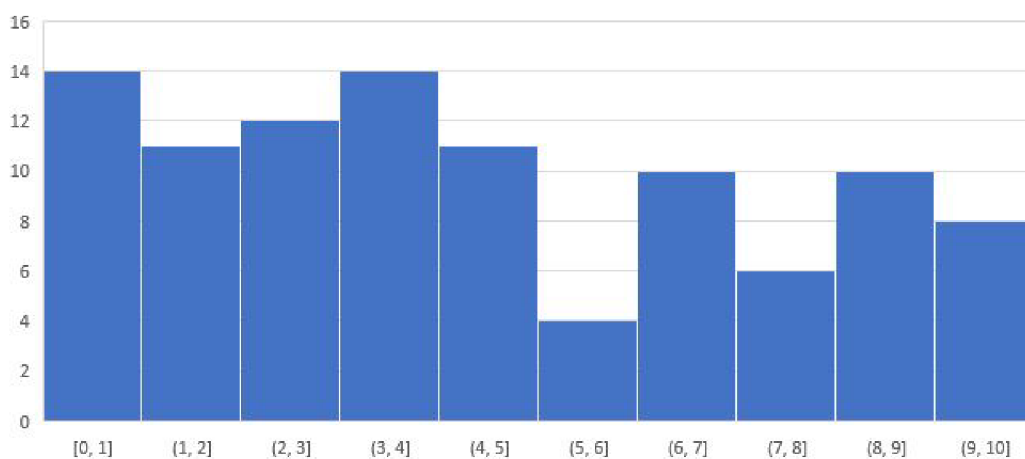
Metoda	Okluze	Trénovací data	Počet objektů ke sledování
Kalmanův Filtr	Ne	Ne	Jeden
Filtr částic	Ano	Ne	Několik
Sledování více hypotéz	Ano	Ne	Několik
Porovnání šablon	Částečná	Ne	Jeden
Průměrný posun	Částečná	Ne	Jeden
Podpora vektorového stroje	Částečná	Ano	Jeden
Sledování založené na vrstvách	Ano	Ne	Několik
Shoda obrysu	Ano	Ano	Několik
Shoda tvaru	Ne	Ne	Jeden

3 Integrální histogram

Na úvod je vhodné vysvětlit, co je to vůbec histogram, co je barevný histogram a proč se používá pro sledování objektů. Dále bude vysvětleno, jaké metody se využívají pro prohledávání obrázku a porovnávání jednotlivých histogramů mezi sebou.

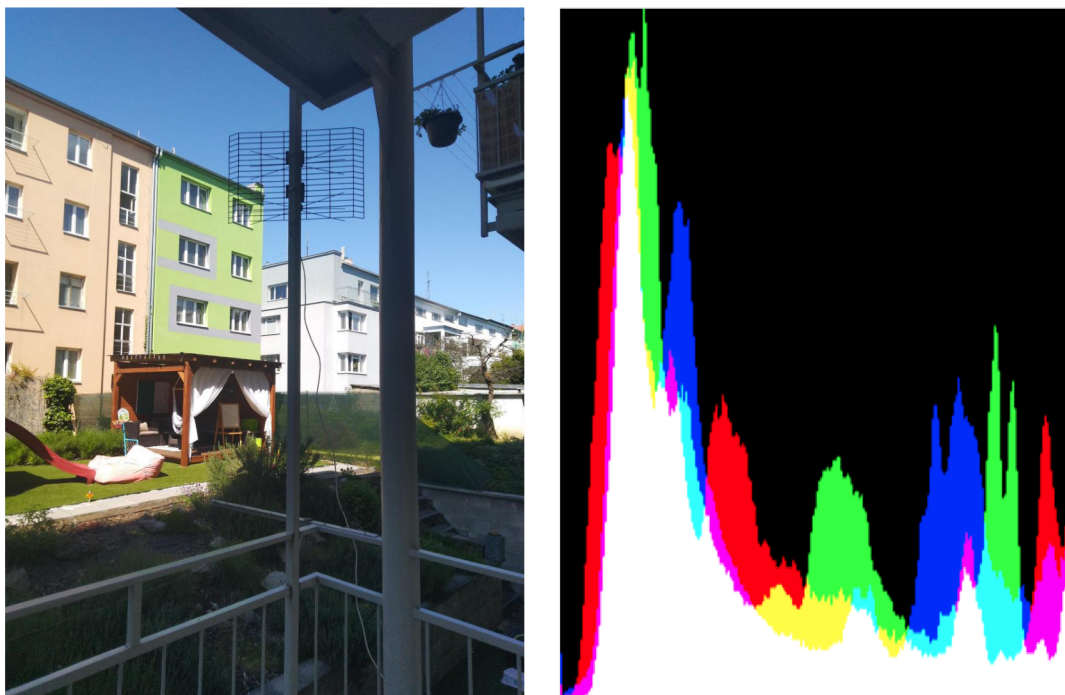
3.1 Barevný histogram

Histogram je graf, který zobrazuje distribuci číselných proměnných hodnot jako řadu sloupců. Každý sloupec obvykle pokrývá rozsah číselných hodnot, které jsou nazvané jako zásobníky. Výška daných sloupců označuje výskyt datových bodů s hodnotou odpovídajícího zásobníku. Histogramy jsou dobré pro zobrazení obecných distribučních funkcí proměnných datových sad. V histogramu lze vidět, kde jsou vrcholy distribuce, zda je distribuce zkosená, nebo symetrická nebo jestli se v grafu nacházejí nějaké odlehlé hodnoty. Aby bylo možné použít histogram, je nutné mít nějakou proměnnou, která využívá po sobě jdoucí číselné hodnoty. To znamená, že rozdíly mezi hodnotami jsou konstantní bez ohledu na jejich absolutní hodnoty. Nastavení zásobníku je věc, která musí být zohledněna při sestavování histogramu. Způsob, jak jsou zvoleny zásobníky má velký efekt na to, jak jsou histogramy interpretovány. Pokud je hodnota na hranici zásobníku, bude trvale přiřazena k zásobníku pravé nebo levé straně. Která strana je určena záleží na vizualizačním nástroji. Některé nástroje mají možnost přepsat výchozí nastavení [52]. Příklad histogramu lze vidět na obrázku 3.1. Na ose X jsou zásobníky rozsahu náhodných čísel, a na ose Y je četnost čísel v zásobníku.



Obr. 3.1: Ukázka histogramu.

V odvětví zpracování obrazu je barevný histogram reprezentací distribuce barev ve snímku. Barevný histogram lze vytvořit pro jakýkoliv druh barevného prostoru, ale nejčastěji se používají trojrozměrné prostory RGB nebo HSV (viz obr. 3.2). Pro monochromatické obrázky se používá histogram intenzity. Pro multispektrální obrázky, kde je každý pixel reprezentován libovolným počtem měření (např. nad rámce RGB), je barevný histogram N -rozměrný, kde N je počet měření. Každé měření má svůj vlastní rozsah vlnových délek světelného spektra, z nichž některé mohou být mimo viditelné spektrum. Pokud je sada hodnot barev malá, může být každá z těchto barev umístěna do určitého rozsahu. Pak histogram obsahuje pouze počet pixelů, které mají každou možnou barvu. Nejčastěji je prostor rozdělen do vhodného počtu rozsahů (zásobníků), uspořádaných jako pravidelná mřížka, z nichž každá obsahuje mnoho podobných hodnot barev. Barevný histogram může být také znázorněn a zobrazen jako hladká funkce definovaná v barevném prostoru, který se blíží počtu pixelů. Stejně jako jiné druhy histogramů je i barevný histogram statistikou, na kterou lze nahlížet jako na aproximaci základní kontinuální distribuce hodnot barev [53].



Obr. 3.2: Barevný histogram snímku.

Hlavní nevýhodou histogramů pro klasifikaci je, že reprezentace závisí na barvě sledovaného objektu, bez toho, aniž by byl brán v potaz jeho tvar nebo textura. Barevné histogramy mohou být teoreticky identické pro dva snímky s různým obsahem

jen díky tomu, že náhodou sdílejí stejné barevné informace. Dalším problémem je to, že barevný histogram má vysokou citlivost na šum, jako je např. změna intenzity osvětlení. Dalším problémem je vysoký rozměr zásobníků. Tento problém lze vyřešit např. zavedením průniku barevných histogramů, výpočtem kvadratické vzdálenosti nebo indexování barevných konstant. Použití barvy v systému, v reálném čase, má několik výhod. Jednou z nich je, že informace o barvách se počítají rychleji ve srovnání s jinými invariantami. V určitých případech je to velmi efektivní metoda pro identifikaci objektů se známým umístěním a vzhledem. Pro výpočet podobnosti dvou snímků se používá např. průnik histogramů nebo euklidova vzdálenost. Výsledku se říká váha a sama o sobě nemá vypovídající hodnotu, využívá se pouze v případě, že je porovnána s jinými váhami. Omezením je tedy to, že při porovnávání snímku s objektem je nutnost, aby předpokládaný objekt prošel celý snímek a vypočítal váhy v celém snímku [54].

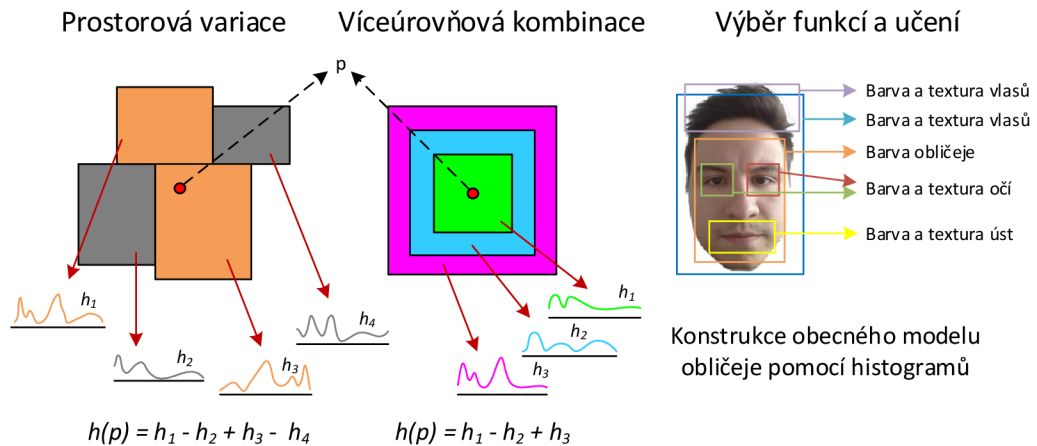
Dalším přístupem k vyjádření obsahu barevného snímku je dvourozměrný barevný histogram. 2D barevný histogram zvažuje vztah mezi barvami párů pixelů. 2D barevný histogram je dvourozměrné pole, tedy matice, přičemž každý prvek ukládá normalizovaný počet párů pixelů, kde každá barva odpovídá indexu prvku v každém sousedství pixelů. Pro výpočet podobnosti dvou obrazů lze využít např. chi-kvadrát test [55] nebo Bhattacharyya vzdálenost [56] [57].

3.2 Vyhledávání v obraze

Histogramy patří mezi nejběžnější funkce používané v mnoha úlohách počítačového vidění jako např. vyhledávání objektů v obraze [58], segmentaci [59], detekci [60] a sledování [61]. Porikli [1] zobecnil koncept integrálního obrazu a představil výpočetně velmi rychlou metodu extrahování histogramu z jakéhokoliv obrazu v konstantním čase. Integrální histogram zajišťuje optimální a kompletní řešení problematiky hledání objektu na základě histogramu. Tato metoda má tři hlavní výhody:

1. Výpočetně rychlejší než konvenční přístupy. Integrální histogram je schopen zpracovat i složité vyhledávací procesy v reálném čase, což dříve bylo velmi nepraktické
2. Může být rozšířena na vyšší datové dimenze, stejnosměrné i nestejněměrné zásobníky a více cílů bez ztráty výpočetního výkonu.
3. Umožňuje popis prvků vyšší úrovně histogramu (viz 3.3).

Aby byly splněny výpočetní požadavky pro detekci, byla vyvinuta metoda pro rychlé výpočty histogramů ve všech možných oblastech objektů v daných datech. Je



Obr. 3.3: Pokročilé prvky lze vypočítat pomocí integrálního histogramu. [1]

využito prostorové rozmístění dat v kartézské soustavě souřadnic a je šířena agregovaná funkce, která se nazývá integrální histogram. Ta začíná v počátečním bodě a prochází všemi zbývajících body díky skenovací linii. Integrální histogram v daném bodě je iterován na základě histogramů, dříve zpracovaných datových bodů. V každém kroku se navyšuje hodnota zásobníku daného bodu. Poté, co je získán integrální histogram pro každý datový bod, je možné jednoduše dopočítat histogramy cílových oblastí s využitím integrálních hodnot histogramu v rohových bodech těchto oblastí bez nutnosti rekonstrukce histogramů pro každou oblast. V 2D datech, jako je třeba obrázek, se převádí integrální histogram na extrakci obdélníkových oblastí histogramů, které se počítají průnikem integrálního histogramu ve čtyřech rohových bodech.

Je možné vypočítat součet hodnot v pravoúhlých oblastech v lineárním čase bez opakování operátoru sumace pro každou možnou oblast [29]. Kumulativní obrazová funkce je definovaná tak, že každý prvek funkce zadržuje hodnotu součtu hodnot nalevo a nad pixelem, včetně hodnoty samotného pixelu. Kumulativní obraz lze vypočítat pro všechny pixely se čtyřmi aritmetickými operacemi na pixel. Začíná se od levého horního rohu přecházejícím nejprve doprava a poté dolů, je hodnota kumulativního obrazu na aktuálním pixelu rovna součtu levého a horního pixelu a odečtení kumulativních hodnot levého horního pixelu. Po výpočtu kumulativního obrazu lze součet obrazové funkce v obdélníku vypočítat pomocí dalších čtyř aritmetických operací s příslušnými úpravami na okraji. S lineárním výpočtem lze tedy v lineárním čase vypočítat součet obrazové funkce na jakémkoliv obdélníku.

Konvenční přístup k měření vzdáleností mezi daným histogramem a histogramem všech možných cílových oblastí je složitý. Tento proces vyžaduje generování

histogramů pro regiony vycentrované v každém možném bodu. V případě, že hledání by bylo provedeno pomocí různých měřítek, neboli různých velikostech cílové oblasti, tak by se proces opakoval tolikrát, kolik je různých měřítek.

3.3 Matematická formulace integrálního histogramu

Integrální histogram je rekurzivní metoda nacházející se v kartézské soustavě a může být rozšířena do libovolného dimenzionálního datového prostoru a reprezentována jakýmkoli tenzorem. Jedná se o nadmnožinu kumulativní formulace zmíněné v předchozí sekci. K porovnání histogramu je potřeba prvně vygenerovat integrální histogram a poté vypočítat histogram cílových oblastí průniku.

Předpokládejme, že funkce f je definována v d -dimenzionální prostředí a reálná hodnota v kartézském prostředí R^d jako $x \rightarrow f(x)$, kde $x = [x_1, \dots, x_d]$ je bod v prostoru. Tato funkce mapuje do k -rozměrného tenzoru, tj. $f([x_1, \dots, x_d]) = [y_1, \dots, y_k]$. Předpokládá se, že d -dimenzionální datový prostor je v rozsahu N_1, \dots, N_d , tj. $0 \leq x_i \leq N_i$.

Integrální histogram je definován jako $H(x^p, b)$ v datovém bodě, v řádu p^{th} společně se sekvencí bodů x^0, x^1, \dots, x^p a je vyjádřen jako:

$$H(x^p, b) = \cup_{j=0}^p Q(f(x^j)). \quad (3.1)$$

$Q(\cdot)$ je odpovídající zásobník v daném bodě a \cup je operátor sjednocení, který je definován: hodnota zásobníku b z histogramu $H(x^p, b)$ je roven součtu všech předtím navštívených bodů zásobníků histogramu, který je součtem $Q(f(x^j))$ kde $j < p$. $H(x^p, b)$ je histogram oblasti mezi původním a aktuálním bodem: $0 \leq x_1^j \leq x_1^p$, $0 \leq x_2^j \leq x_2^p$, ..., atd. Integrální histogram může být napsán rekurzivně jako

$$H(x^p, b) = H(x^{j-1}, b) \cup Q(f(x^j)). \quad (3.2)$$

S počáteční podmínkou $H(0, B) = 0$, což znamená, že v původním bodě jsou všechny zásobníky prázdné.

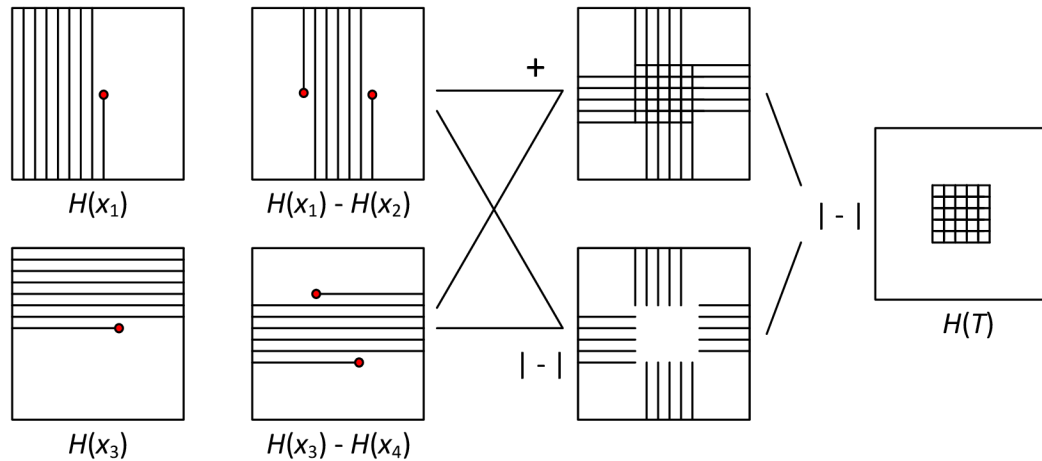
3.4 Metody procházení oblastí

Je více různých přístupů k procházení a skenování oblastí. Porikli [1] popisuje dvě hlavní metody, které se používají nejvíce.

Řetězcová metoda prohledávání obrazu

Jedna z metod je řetězcová metoda, která prochází datovou oblast podél každé dimenze, např. zleva doprava a shora dolů pro obrazová data. Integrální histogram

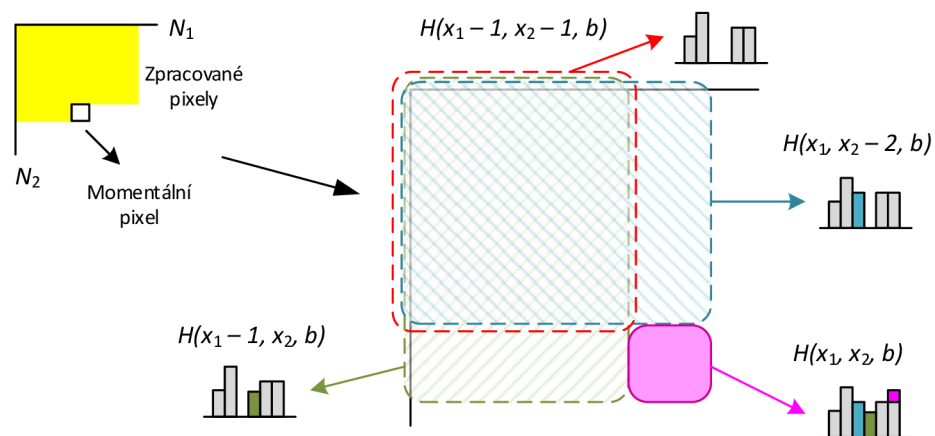
v aktuálním bodě je získán zkopírováním předchozích hodnot a zvětšením odpovídajícího zásobníku vzhledem k aktuální hodnotě. Řetězcová metoda se aktualizuje v každém kroku. Pro získání histogramu z oblasti polytopu v d-dimenzionálních datech by mělo být provedeno skenování d řetězců v různých rozměrech, jak je znázorněno na 3.4.



Obr. 3.4: Znázornění řetězcové metody prohledávání obrazu.

Prohledávání obrazu pomocí vlnových ploch

Je také možné pro skenování bodů využít aktivní sady bodů tzv. Vlnoplochy.



Obr. 3.5: Prohledávání obrazu pomocí vlnových ploch.

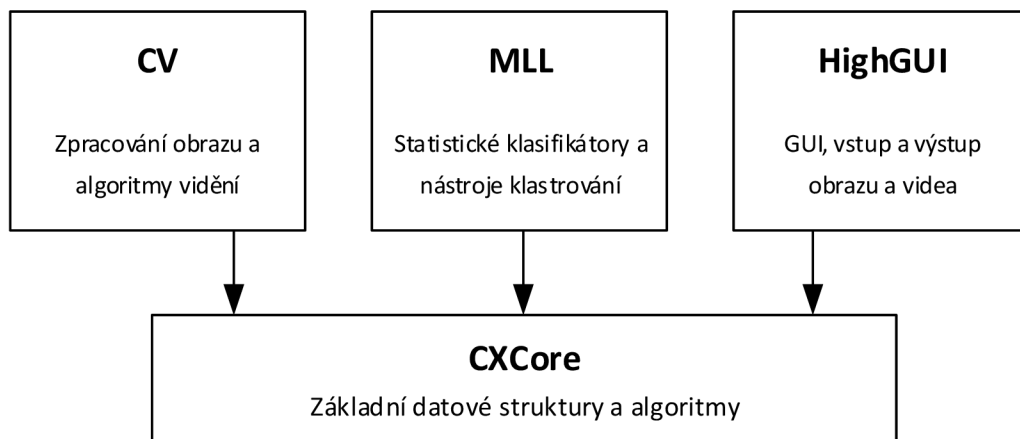
Prohledávání pomocí vlnoploch vyžaduje aktualizaci integrálního histogramu pro takové datové body, kde jejich sousední levý, horní a levý horní jsou už skenovány pro obrazová data. Integrální histogram v bodě je získán pomocí tří aritmetických operací pro každý zásobník s využitím hodnot integrálního histogramu, pro tři sousedy, jak je vidět na obrázku 3.5. Hodnoty integrálního histogramu předešlých bodů jsou kopírovány do daného bodu před další propagací. Buď se aktualizovaný zásobník zkopíruje do všech ostatních zásobníků zbývajících bodů (Celkem $\frac{N^2 - 2}{2}$ kopírovacích operací) nebo všechny předešlé zásobníky jsou zkopírovány do aktuálního (BN operace), což lze provést pomocí rychlého kopírování paměti na úrovni hardwaru, nebo tabulkami ukazatelů.

4 Řešení práce

V této kapitole je popsána praktická část práce. Cílem práce je optimalizace integrálního histogramu pro práci v reálném čase a vytvoření funkční aplikace, která je schopna sledovat uživatelem určený objekt. Uživatel může během přehrávání zastavit video, zvolit si objekt zájmu a znovu spustit video se sledovaným objektem. Tento objekt je pak vyznačený v dalších snímcích. Hlavním problémem je řešení vzdálenosti objektu od kamery a změna velikosti sledovacího obdélníku. Práce je realizována přes konzolovou aplikaci s využitím prostředí Visual Studio a knihovny OpenCV. Celá práce je pak napsána v programovacím jazyce C++.

4.1 Visual Studio a OpenCV

Pro práci je zvoleno vývojové prostředí Visual Studio, s využitím knihovny OpenCV¹, která se používá pro zpracování obrazu. Visual Studio je integrované vývojové prostředí, díky kterému je možné upravovat, ladit a vytvářet kód. OpenCV je volně dostupná knihovna v C++ pro zpracování obrazu a počítačové vidění. Je to knihovna zaměřená hlavně na zpracování videa v reálném čase, což je pro naši aplikaci žádoucí. Má několik stovek funkcí, které implementují algoritmy zpracování obrazu a usnadňují vývoj aplikací. Knihovna je strukturovaná do 4 hlavních částí (viz obr. 4.1).



Obr. 4.1: Struktura knihovny OpenCV.

¹<https://opencv.org/>

4.2 Testy hypotéz pro porovnání histogramů

Pro výpočet váhy dvou histogramů je potřeba vybrat metriku ($d(H_1, H_2)$) k vyjádření váhy. Váha je v tomto případě míra shody. V první části je práce jsou porovnány výpočty, které nabízí knihovna OpenCV pro výpočet váhy dvou histogramů. OpenCV knihovna nabízí tyto metody výpočtu:

- **Korelace** - CV_COMP_CORREL:

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}}, \quad (4.1)$$

kde

$$\bar{H}_k = \frac{1}{N} \sum_J H_k(J),$$

a N je celkový počet zásobníků histogramu.

- **Chi-kvadrát** - CV_COMP_CHISQR:

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)} \quad (4.2)$$

- **Intersekcce** - CV_COMP_INTERSECT:

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I)) \quad (4.3)$$

- **Bhattacharyya vzdálenost** - CV_COMP_BHATTACHARYYA

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}} \quad (4.4)$$

Pro otestování výpočtů a detekce objektu ve snímku pomocí šablony je nutné mít snímky ve správném barevném formátu (HSV). Tento barevný formát vyžaduje OpenCV funkce `calcHist()`, která slouží k výpočtu histogramu. Vstupní parametry jsou následující:

- **Snímek**, ze kterého se počítá histogram.
- **Počet snímků**
- **Počet kanálů** - rozsah (pro tři kanály `channels[] = {0, 2}`).
- **Výstupní histogram**
- **Počet dimenzí**
- **Velikost histogramu** - počet zásobníků v každém směru.
- **Rozsah hodnot** - rozsahy hodnot zásobníků.
- **Uniformnost** - zjišťuje, jestli je histogram unimodální nebo ne.
- **Akumulace** - pokud je nastavena, histogram není při alokaci vymazán.

```

1 // Vstupni snimek
2 Mat img = imread("surf.jpg", IMREAD_COLOR);
3 // Pocet barevnych kanalu
4 int channels[] = {0, 2};
5 // Vypocitany histogram
6 Mat output;
7 // velikost histogramu, odstín = 30, saturace = 32
8 int histSize[] = {30, 32};
9 // rozsahy hodnot
10 float hranges[] = {0, 180}; // Odstín
11 float sranges[] = {0, 256}; // Saturace
12 const float* ranges[] = {hranges, sranges};
13 bool uniform = true;
14 bool accumulate = false;
15
16 // vypocet histogramu
17 calcHist(&img, 1, channels, Mat(), output, 2, histSize,
           ranges, uniform, accumulate);

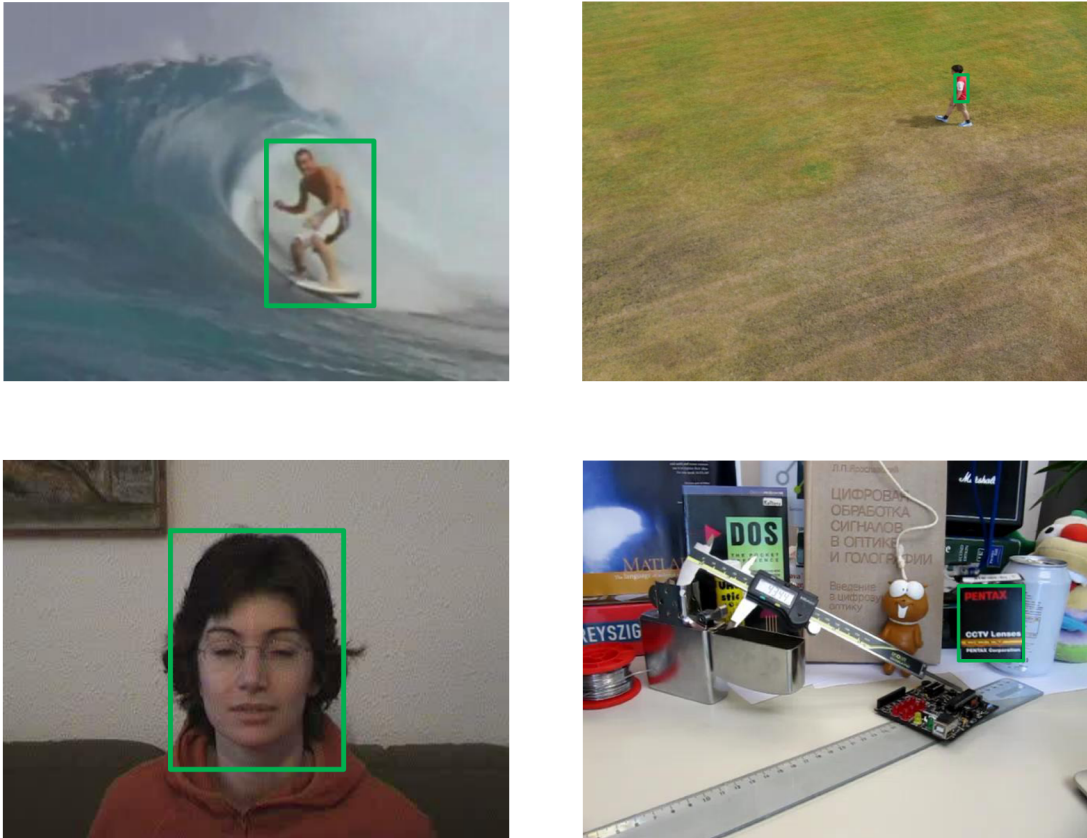
```

Využití této funkce je výpočetně náročné a proto není vhodné ji použít pro real-time sledování. Postup pro porovnání dvou snímků je následující:

1. Uživatel vytvoří šablonu z prvního obrázku (viz 4.3).
2. Aplikování výpočtu integrálního histogramu na celý snímek a na šablonu.
3. Porovnávání histogramu šablony a histogramu stejné velikosti ve druhém snímku pomocí `compareHist()`.
4. S využitím metody klouzavého okna je ve druhém snímku nalezena oblast s nejlepší vahou 4.4.
5. Vykreslení oblasti nalezeného objektu.

V prvním kroku je vybrána šablona uživatelem. To se dělá před spuštěním programu oříznutím šablony z původního snímku. V druhém kroku je poté vypočítán integrální histogram prvního snímku a oblasti zájmu. Výpočtem integrálního histogramu je zjištěna intenzita barvy jednotlivých pixelů. Jelikož je nutné porovnávat histogramy o stejné velikosti, musí se porovnávat histogram šablony a stejně velká oblast z druhého snímku. To je dosaženo pomocí funkce `regionHistogram()`, která z vypočteného integrálního histogramu z prvního snímku zjišťuje hodnoty intenzity barev pixelů pouze z dané oblasti. Tato oblast má stejnou velikost jako šablona. Poté se provede třetí krok a oblast se porovnává se šablonou. Takhle se pomocí klouzavého okna porovnají všechny oblasti v druhém snímku a oblast, kde je nejlepší výsledek váhy je objekt zájmu. V této oblasti je pak vykreslen obdélník. Uživatelem zvolená

šablona musí být co nejpřesněji okolo objektu zájmu, jinak může dojít k chybnému sledování objektu. Pokud je zvolena větší oblast, než je objekt, může být sledování ovlivněno barvami v pozadí. Pokud mezi snímky nedojde k výrazné transformaci hledaného objektu, je objekt nalezen. Tento postup je aplikován na několik situací, které se liší transformací objektu a celého snímku.



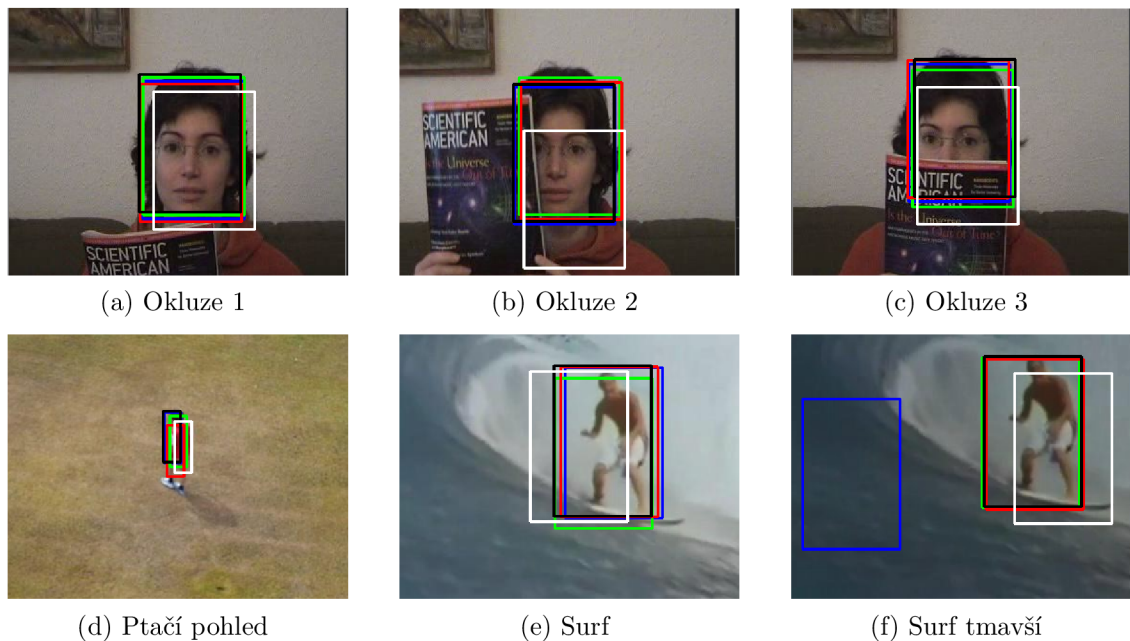
Obr. 4.2: Vybrané šablony pro porovnání metod výpočtu.

Pro demonstraci detekce objektu byly vybrány tři scénáře. V prvním případě je pozorována detekce při okluz. (viz 4.3a, 4.3b, 4.3c). V druhém případě dochází ke změně vzhledu objektu (dochází ke změně polohy kamery, viz 4.3d). Ve třetím případě se změnila poloha objektu a změnil jas obrazu (viz 4.3e, 4.3f). Každá metoda výpočtu má přiřazenou svoji barvu ohraničujícího obdélníku. Metodě chi-kvadrátu patří červený ohraničující obdélník, průniku zelený, Bhattacharyya vzdálenosti modrý, alternativnímu chi-kvadrátu černý a KL divergenci bílý. Kromě výpočtu metodou průniku platí, že čím menší je vypočtená váha, tím větší přesnost.

Tab. 4.1: Tabulka vypočtených vah.

	Chi-kvadrát (Červená)	Průník (Zelená)	Bhattach. (Modrá)	X^2_{alt} . (Černá)	KL_div (Bílá)
Ptačí pohled	2200,71	317	0,71469	2219,03	12675,6
Okluze 1	2827,33	12813	0,218036	5661,67	6239,8
Okluze 2	10089,2	11113	0,32527	9773,41	13407,7*
Okluze 3	30783,5	9523	0,401322	14950,3	9725,01
Surf	5817,41	10267	0,336751	9681,78	29755,1*
Surf tmavší	14473,3	4417	0,744665	39058,7	233375

Z výsledků lze vidět, že sledování objektu pomocí histogramu pracuje velmi dobře s okluzí. Dobrých výsledků bylo dosaženo i při změně pohledu. Dobrých výsledků nebylo dosaženo pro zmenšení jasu snímku, jelikož již při malém zmenšení u metody výpočtu Bhattacharyya je výsledek špatně. KL divergence měla ze všech metod nejhorší přesnost detekce. Objekt je ve všech případech nalezen, ale obdélník neohraničuje celý objekt podle šablony a tyto případy jsou vyznačeny v tab. 4.1 .



Obr. 4.3: Vybrané šablony pro porovnání metod výpočtu.

4.3 Sledování v reálném čase

V následující části je řešena problematika sledování objektu ve videu (v reálném čase). Jelikož funkce použita v sekci 4.2 je výpočetně náročnější, je potřeba vytvořit funkci pro výpočet váhy. Nejsnadněji implementovaná metoda výpočtu je chi-kvadrát, který je implementován v další části.

Před samotnou prací s histogramy je nutné načíst snímek, ve kterém je objekt vyhledáván. Šablonu je po zmáčknutí klávesy **N** zvolena zvolením ohraničujícího obdélníku okolo objektu. Zvolení nové šablony je možné díky metodě `selectNewTemplate()`, které se předává video a matice snímku. V aplikaci pro sledování v reálném čase, která je v této práci, je použit výpočet pomocí chi-kvadrátu. Výpočet je aplikován ve funkci `compare_X2()`, kde jako vstupní data jsou dva histogramy o stejné velikosti a vrácená hodnota je váha porovnání. V případě chi-kvadrátu platí, že čím menší je vrácená hodnota, tím víc jsou si oba histogramy podobné. Funkce je zobrazena níže:

```
1 float compare_X2(uint16_t *h1, uint16_t *h2, int len)
2 {
3     float sim = 0;
4     float val;
5     for (int i = 0; i < len; i++)
6     {
7         val = h1[i] - h2[i];
8         if (val != 0)
9             sim += (val*val) / (h1[i] +
10                h2[i]); // Chi-square
11     }
12     return sim;
13 }
```

Po zvolení šablony a výpočtu integrálního histogramu šablony i snímku je nutné vypočítat histogramy obdélníkové oblasti o velikosti šablony. Toho je docíleno pomocí metody klouzavého okna.

4.3.1 Klouzavé okno s přizpůsobením oblasti

Pro nalezení objektu v druhém snímku ze šablony prvního snímku je nutné využít metodu klouzavého okna. Nejprve je nutné vypočítat integrální histogram pro šablonu a pro snímek, ve kterém je hledán objekt pomocí šablony. Poté pomocí funkce

`regionHistogram()` je vypočítán histogram obdélníku o velikosti šablony a postupným posouváním po snímku se vyhledává objekt zájmu (viz obr. 4.4). Každý tento obdélník se porovná se šablonou pomocí funkce `compareHist()`. Všechny hodnoty jsou uloženy do vektoru. Po dokončení prohledávání je vektor seřazen od nejmenších vah. Nejmenší váhy mají největší pravděpodobnost nalezení objektu ve snímku.



Obr. 4.4: Prohledávání pomocí klouzavého okna.

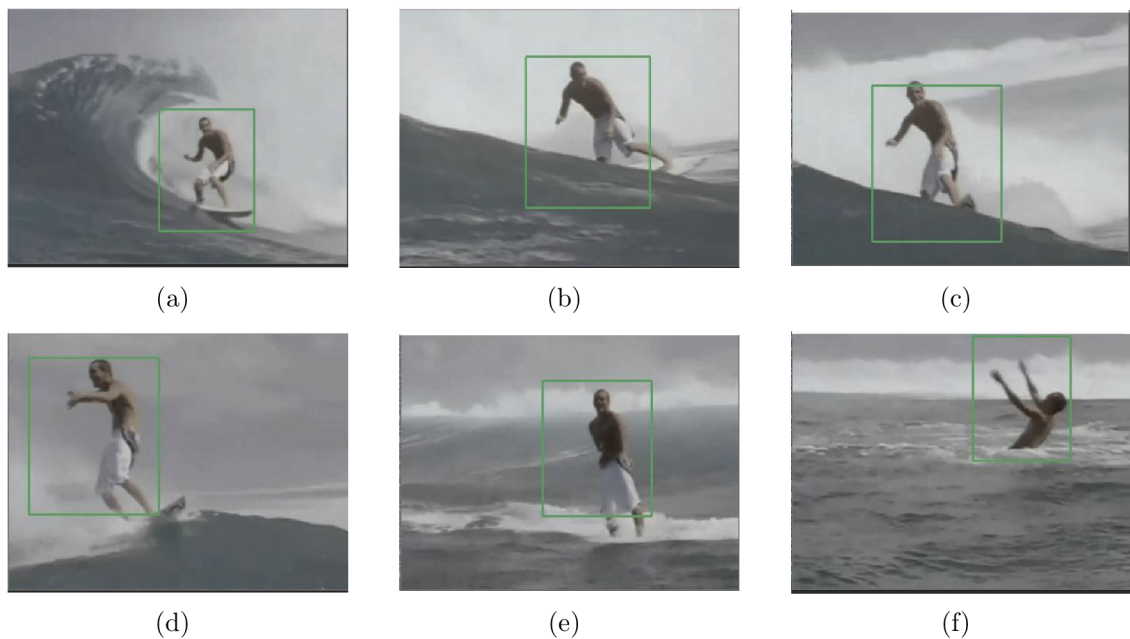
Jelikož je žádoucí i změna velikosti ohraničujícího obdélníku podle změny velikosti objektu, je při procházení snímku implementováno i zvětšování a zmenšování prohledávajícího obdélníku. Toho je docíleno prohledáváním snímku pomocí obdélníku, který má v jednom cyklu menší velikost, v druhém stejnou a v třetím větší velikost než šablona. Prohledávání snímku tedy proběhne třikrát pro tři různé velikosti obdélníku. Tyto obdélníky jsou také porovnávány funkcí `compareHist()` se šablonou a vyhodnocená váha je ukládána do příslušných vektorů.

4.3.2 Výpočet finální váhy a určení polohy objektu.

Jakmile prohledávání skončí, musí se vektory vytřídit tak, aby se pracovalo jen s hodnotami, které mají co nejmenší váhu. To je dosaženo funkcí `sort()`, která vytřídí podle váhy jednotlivé obdélníky. Funkce `resizeArray()` pak vyselektuje na základě váhy několik nejmenších vah. Tohle udělá pro všechny vektory. Ze všech zbylých vah se poté vypočítá průměr pomocí funkce `averageTracking()` a tím se získá hledaný obdélník. Podle něho je poté aktualizována šablona a celý proces probíhá znovu pro další snímek.

V dalším snímku už však není prohledávaná celá oblast, jelikož se neočekává, že by objekt se přesunul přes celou scénu během jednoho snímku. K prohledávání celé oblasti je potřeba mnohem větší výkon než když se vymezí oblast okolo sledovaného objektu. V práci je momentálně vymezená oblast prohledávání nastavena na ± 50 pixelů. Tato hodnota zajišťuje plynulý chod videa a bezproblémovou detekci a sledování objektu. Tím, že se omezí oblast sledování se předchází i potenciálním chybám, které by mohli vzniknout výpočtem histogramů objektů mimo námi sledovaný objekt zájmu.

Sledování objektu a měnící se velikost objektu lze pozorovat na obrázcích 4.5.



Obr. 4.5: Sledování objektu v reálném čase [63].

4.3.3 Shrnutí

Sledování bylo plynule pro video [63] o rozlišení 480x360 na Laptopu ThinkPad T490. Pro dané video fungovalo sledování v reálném čase s měnící se velikostí ohraničujícího obdélníku dobře. Problém by mohl nastat, kdyby se měnilo osvětlení scény nebo pokud by se objevil objekt u objektu zájmu. Omezením vyhledávané oblasti se značně ušetří počet početných operací a sníží se nároky na výpočetní výkon. U vyšších rozlišení je však potřeba více výpočetního výkonu. Sledování by se dalo použít u dohledových kamer, kde není nutný velký počet snímků za vteřinu.

5 Závěr

Tato práce se zabývala sledováním objektu ve videu pomocí porovnávání histogramů. Cílem bylo zprovoznit dodaný program, seznámit se s problematikou sledování objektu a integrálního histogramu a rozšířit existující řešení detekce změn mezi dvěma snímky. Rozšíření spočívalo v aplikaci sledování objektu ve videu, v reálném čase s měnící se velikostí ohraničujícího obdélníku. V 1. kapitole byly popsány jednotlivé kroky související se zpracováním obrazu a sledováním objektu. V 2. kapitole jsou podrobněji popsány způsoby a metody sledování objektů. Kapitola je rozdělena do sekcí podle způsobů sledování. Ke konci kapitoly jsou popsány časté problémy u sledování objektů. V kapitole 3 je obecně histogram a následovaný detailním popisem integrálního histogramu vycházejícího z práce [1]. Poslední kapitola č. 4 popisuje řešení práce. V první části 4. kapitoly je popsáno vybrané vývojové prostředí, následované otestováním metod výpočtu vah pro porovnání histogramů.

V této práci byla zprovozněn program poskytnutý vedoucím práce pro detekci změn v obraze pomocí knihovny pro zpracování obrazu OpenCV, vývojového prostředí Visual Studia a programovacího jazyka C++. V práci je prostudována problematika sledování objektů a integrálního histogramu, ze kterého vychází poskytnutý program. Dále jsou porovnány metody výpočtu pro porovnání dvou histogramů. Správnou implementací těchto metod je možné zlepšit přesnost sledování.

Hlavním přínosem práce byl rozšíření dosavadního řešení o sledování objektu v reálném čase. Tento objekt je zvolen uživatelem. Správně zvolená šablona zvyšuje přesnost sledování. Dalším z přínosů je upravování velikosti ohraničujícího obdélníku podle velikosti sledovaného objektu.

Návrh na budoucí rozšíření práce je možné vést více směry. Práce by se dala rozšířit o sledování více objektů zároveň. Další z možností by byla implementace strojového učení pro zlepšení detekce a sledování objektu např. i při opuštění scény. Další možností by byla implementace EDM, což je další z metod pro výpočet váhy dvou histogramů.

Literatura

- [1] F. PORIKLI *Integral histogram: a fast way to extract histograms in Cartesian spaces* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), June 2005, p829-836 vol. 1 [cit 2019-10-27] Dostupné z <<https://ieeexplore.ieee.org/document/1467353/citations#citations>>
- [2] MATHUR, Garima, Devendra SOMWANSHI a Mahesh m BUNDELE. *Intelligent Video Surveillance based on Object Tracking*. In: 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE) [online]. IEEE, 2018, s. 1-6 [cit. 2019-12-03]. DOI: 10.1109/ICRAIE.2018.8710421.
- [3] KUNII, Y., G. KOVACS a N. HOSHI. *Mobile robot navigation in natural environments using robust object tracking* In: IEEE International Symposium on Industrial Electronics [online]. Institute of Electrical and Electronics Engineers, 2017, s. 1747-1752 [cit. 2019-12-03]. DOI: 10.1109/ISIE.2017.8001512. ISBN 9781509014125.
- [4] MUHAMMAD HAMDAN, H.g. a O.o. KHALIFAH. *Traffic intensity monitoring using multiple object detection with traffic surveillance cameras* In: IOP Conference Series: Materials Science and Engineering [online]. Institute of Physics Publishing, 2017, 260(1), [cit. 2019-12-03]. DOI: 10.1088/1757-899X/260/1/012009. ISSN 17578981.
- [5] FAROOQ, Javeria. *Object detection and identification using SURF and BoW model*. In: 2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube) [online]. IEEE, 2016, s. 318-323 [cit. 2019-12-03]. DOI: 10.1109/ICECUBE.2016.7495245.
- [6] Hatwar B. Rupali, Kamble D. Shailesh *A Review on Moving Object Detection and Tracking Methods in Video* International Journal of Pure and Applied Mathematics. Volume 118 No. 16 2018, 511-526 [cit. 2019-12-03]. ISSN 1314-3395. Dostupné z: <<https://acadpub1.eu/jsi/2018-118-16-17/articles/16/33.pdf>>
- [7] PENG, Shanshan. *Motion Object Detection Research Based on Texture Feature*. Sensors & Transducers [online]. Toronto: IFSA Publishing, S.L, 2014, 175(7), 279-283 [cit. 2020-04-13]. ISSN 23068515. Dostupné z: <<http://search.proquest.com/docview/1553183958/>>

- [8] YABUKI, N. ; NISHIMURA, N. ; FUKUDA, T. *Automatic object detection from digital images by deep learning with transfer learning*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, Vol.10863, pp.3-15.
- [9] FURINI, M. *On introducing timed tag-clouds in video lectures indexing*. Multimedia Tools and Applications, 2018, Vol.77(1), pp.967-984 [cit. 2020-05-23]. doi:10.1007/s11042-016-4282-5.
- [10] DONDI, P.; PORTA, M.; LOMBARDI, L. *Development of gesture-based human-computer interaction applications by fusion of depth and colour video streams*. IET Computer Vision, 2014 Dec, Vol.8(6), pp.568-578 [online]. [cit. 2020-04-13]. DOI:10.1049/iet-cvi.2013.0323/ISSN: 1751-9632
- [11] CARO-GUTIERREZ, J., M.E. BRAVO-ZANOQUERA a F.F. GONZÁLEZ-NAVARRO. *Methodology for automatic collection of vehicle traffic data by object tracking*. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [online]. Springer Verlag, 2017, 10061, s. 482-493 [cit. 2020-04-13]. DOI: 10.1007/978-3-319-62434-1_39. ISBN 9783319624334. ISSN 03029743.
- [12] YE, Y. ; FU, L. ; LI, B. *Object detection and tracking using multi-layer laser for autonomous urban driving* IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 22 December 2016, pp.259-264. DOI: 10.1109/ITSC.2016.7795564. ISBN: 9781509018895.
- [13] YILMAZ, Alper, Omar JAVED a Mubarak SHAH. *Object tracking: A survey*. ACM Computing Surveys (CSUR) [online]. ACM, 2006, 38(4), 13-es [cit. 2020-04-13]. DOI: 10.1145/1177352.1177355. ISSN 03600300.
- [14] CHOI, Jin, Taeg WHANGBO a Cheong KIM. *A contour tracking method of large motion object using optical flow and active contour model*. Multimedia Tools and Applications [online]. Boston: Springer US, 2015, 74(1), 199-210 [cit. 2019-12-10]. DOI: 10.1007/s11042-013-1756-6. ISSN 1380-7501.
- [15] BLUM, H. *A transformation for extracting new descriptors of shape*. Models for the Perception of Speech and Visual Form, MIT Press, 1967, pp.362-381. [cit. 2020-25-05].
- [16] XIN, Q. *Diesel engine system design*. Cambridge: Woodhead Publishing, 2011. Woodhead Publishing in mechanical engineering. ISBN 978-1-84569-715-0.

- [17] BLACK, M. a JEPSON, A.. *EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation*. International Journal of Computer Vision, 1998, Vol.26(1), pp.63-84. [cit. 2020-25-05]. DOI: 10.1023/A:1007939232436. ISSN: 0920-5691
- [18] GUENTHER, N. a SCHONLAU. M. *Support vector machines*. Stata Journal [online]. 2016, 16(4), 917-937 [cit. 2020-05-25]. DOI: 10.1177/1536867x1601600407. ISSN 1536867X.
- [19] MIHALJEVIĆ, B.; BENAVIDES-PICCIONE, R.; BIELZA, C. *Bayesian Network Classifiers for Categorizing Cortical GABAergic Interneurons*. Neuroinform 13, 193–208 (2015) [cit. 2020-25-05]. <https://doi.org/10.1007/s12021-014-9254-1>
- [20] PARK, J.; AN, Y.; JEONG, I.; KANG, G.; PANKOO, K. *Image Indexing using Spatial Multi-Resolution Color Correlogram*. 2007 IEEE International Workshop on Imaging Systems and Techniques, Krakow, 2007, pp. 1-4, [cit. 2020-25-05] doi: 10.1109/IST.2007.379604.
- [21] NOURANI-VATANI, N.; BORGES V.K., P.; ROBERTS M., J. *A Study of Feature Extraction Algorithms for Optical Flow Tracking* Proceedings of the 2012 Australasian Conference on Robotics and Automation. The Australian Robotics and Automation Association, Australia, pp. 1-7. [cite 2020-25-05]
- [22] HARRIS, C. a STEPHENS, M. *A Combined Corner and Edge Detector*. Alvey Vision Conference (1988). [cit. 2020-25-05]
- [23] LOWE, D. *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image*. The University of British Columbia (Corporate Author) ; 23 March 2004, [cit. 2020-25-05]
- [24] COMANICIU, D. a MEER, P. *Mean shift: A Robust Approach Toward Feature Space Analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603-619. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 24. 603 - 619. 10.1109/34.1000236, [cit. 2020-25-05]
- [25] Caselles, Vicent *A geometric model for active contours in image processing*. Numerische Mathematik 66 (1993): 1-31., [cit. 2020-25-05]
- [26] STAUFFER, C. ; GRIMSON, W.E.L. *Adaptive background mixture models for real-time tracking* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999, Vol.2, pp.246-252 [cit. 2020-25-05]

- [27] MONNET, A.; MITTAL, A.; PARAGIOS, N.; RAMESH, V. *Background Modeling and Subtraction of Dynamic Scenes*. In Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2 (ICCV '03). IEEE Computer Society, USA, 1305. [cit. 2020-25-05]
- [28] BISHOP, C. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, 1995. ISBN 978-0-19-853864-6.
- [29] VIOLA, P. a Michael J. *Real-Time Face Detection*. International Journal of Computer Vision [online]. Boston: Kluwer Academic Publishers, 2004, 57(2), 137-154 [cit. 2020-01-13]. DOI: 10.1023/B:VISI.0000013087.49260.fb. ISSN 0920-5691.
- [30] LINDBERG, Tony. *Scale Selection Properties of Generalized Scale-Space Interest Point Detectors*. Journal of Mathematical Imaging and Vision [online]. Boston: Springer US, 2013, 46(2), 177-210 [cit. 2020-04-13]. DOI: 10.1007/s10851-012-0378-3. ISSN 0924-9907.
- [31] SHAPIRO LINDA, G. STOCKMAN GEORGE C. *Computer vision*. New Jersey: Prentice-Hall, 2001, 580 s. ISBN 0-13-030796-3.
- [32] SRINIVASAN, K, K PORKUMARAN a G SAINARAYANAN. *Improved background subtraction techniques for security in video applications*. In: 2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication [online]. IEEE, 2009, s. 114-117 [cit. 2019-12-03]. DOI: 10.1109/ICASID.2009.5276945. ISBN 9781424438839.
- [33] TALABIS, M.; McPHERSON, R.; MIYAMOTO, I.; MARTIN, J. a KAYE, D. *Information Security Analytics : Finding Security Insights, Patterns and Anomalies in Big Data*. Waltham 2015, MA: Syngress, 182 pages. ISBN:0128005068, 9780128005064
- [34] WU, Y.; LIM, J. a YANG, M. *Object Tracking Benchmark*. IEEE Transactions on Pattern Analysis and Machine Intelligence [online]. IEEE, 2015, 37(9), 1834-1848 [cit. 2020-05-28]. DOI: 10.1109/TPAMI.2014.2388226. ISSN 0162-8828.
- [35] CHALLA, S.; R MORELANDE, M.; MUŠICKI D. a J EVANS, R. *Fundamentals of Object Tracking*. 2011, 9780521876285. DOI: 10.1017/CBO9780511975837. ISBN 9780521876285.
- [36] KUHN, H. W. *The Hungarian method for the assignment problem*. Naval Research Logistics (NRL) [online]. Hoboken: Wiley Subscription Services, Inc., A

- Wiley Company, 2005, 52(1), 7-21 [cit. 2020-05-28]. DOI: 10.1002/nav.20053. ISSN 0894-069X.
- [37] SHAFIQUE, K a M SHAH. *A noniterative greedy algorithm for multiframe point correspondence*. IEEE Transactions on Pattern Analysis and Machine Intelligence [online]. IEEE, 2005, 27(1), 51-65 [cit. 2020-05-28]. DOI: 10.1109/T-PAMI.2005.1. ISSN 0162-8828.
- [38] Athanesious J. J. a Suresh P. *Systematic Survey on Object Tracking Methods in Video*. In: International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 1, Issue 8, October 2012 [cit 2019-12-4]. ISSN: 2278 – 1323
- [39] BAR-SHALOM, Y, F DAUM a J HUANG. *The probabilistic data association filter*. IEEE Control Systems [online]. IEEE, 2009, 29(6), 82-100 [cit. 2020-05-29]. DOI: 10.1109/MCS.2009.934469. ISSN 1066-033X.
- [40] KIM, C.; LI, F.;CIPTADI, A. a M REHG, J.. *Multiple Hypothesis Tracking Revisited*. In: 2015 IEEE International Conference on Computer Vision (ICCV) [online]. IEEE, 2015, 2015, s. 4696-4704 [cit. 2020-05-29]. DOI: 10.1109/ICCV.2015.533. ISSN 15505499.
- [41] ÇELEBİ, O. C. *Bayesian Decision Theory Celebi Tutorial* [online]. Turkey, [cit. 2020-04-14]. Dostupné z: <https://www.byclb.com/TR/Tutorials/neural_networks/ch4_1.htm>
- [42] COMANICIU, D.; RAMESH, V. a MEER, P. *Kernel-based object tracking*. IEEE Transactions on Pattern Analysis and Machine Intelligence [online]. IEEE, 2003, 25(5), 564-577 [cit. 2020-05-30]. DOI: 10.1109/T-PAMI.2003.1195991. ISSN 0162-8828.
- [43] Perveen N., Kumar D., Bhardwaj Ishan *An Overview on Template Matching Methodologies and its Applications* International Journal of Research in Computer and Communication Technology, Vol 2, Issue 10, October- 2013 [cit. 2019-12-04]. ISSN: 2278-5841
- [44] FIEGUTH, Paul a Demetri TERZOPOULOS. *Color-based tracking of heads and other mobile objects at video frame rates*. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition [online]. IEEE, 1997, s. 21-27 [cit. 2020-05-29]. ISSN 10636919.
- [45] PORIKLI, F a O TUZEL. *Multi-Kernel Object Tracking*. In: 2005 IEEE International Conference on Multimedia and Expo [online]. IEEE, 2005,

- 2005, s. 1234-1237 [cit. 2020-05-29]. DOI: 10.1109/ICME.2005.1521651. ISBN 0780393317.
- [46] LI, X.; LIU, Q.; Zhenyu HE, Hongpeng WANG, Chunkai ZHANG a Wen-sheng CHEN. *A multi-view model for visual tracking via correlation filters*. Knowledge-Based Systems [online]. Elsevier B.V, 2016, 113(C), 88-99 [cit. 2020-04-14]. DOI: 10.1016/j.knosys.2016.09.014. ISSN 0950-7051.
- [47] RADENOVIĆ, F., G. TOLIAS a O. CHUM. *Deep Shape Matching*. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [online]. Springer Verlag, 2018, 11209, s. 774-791 [cit. 2020-06-01]. DOI: 10.1007/978-3-030-01228-1_46. ISBN 9783030012274. ISSN 03029743.
- [48] LIU, Zhao, Hui SHEN, Guiyu FENG a Dewen HU. *Tracking objects using shape context matching*. Neurocomputing [online]. Elsevier B.V, 2012, 83(C), 47-55 [cit. 2019-12-12]. DOI: 10.1016/j.neucom.2011.11.012. ISSN 0925-2312.
- [49] Patel, Ripal. *Contour Based Object Tracking*. (2012), International Journal of Computer and Electrical Engineering. 4. 525-528 [cit. 2020-06-01].
- [50] LEE, B.Y., L.H. LIEW, W.S. CHEAH a Y.C. WANG. *Occlusion handling in videos object tracking: A survey*. IOP Conference Series: Earth and Environmental Science [online]. Institute of Physics Publishing, 2014, 18(1) [cit. 2020-04-14]. DOI: 10.1088/1755-1315/18/1/012097. ISSN 17551307.
- [51] LIU, Wenqian, Octavia CAMPS a Mario SZNAIER. *Multi-camera Multi-Object Tracking*. ArXiv.org [online]. Ithaca: Cornell University Library, arXiv.org, 2017 [cit. 2020-06-01]. Dostupné z: <http://search.proquest.com/docview/2076640202/>
- [52] *A Complete Guide to Histograms*. Chartio [online]. USA, Kalifornie: Chartio, 2019 [cit. 2020-01-13]. Dostupné z: <https://chartio.com/learn/charts/histogram-complete-guide/>
- [53] SHAPIRO LINDA, G. STOCKMAN GEORGE C. *Computer vision*. New Jersey: Prentice-Hall, 2001, 580 s. ISBN 0-13-030796-3.
- [54] NOVAK, C.L a S.A SHAFER. *Anatomy of a color histogram*. In: Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition [online]. IEEE Comput. Soc. Press, 1992, 1992-, s. 599-605 [cit. 2020-06-01]. DOI: 10.1109/CVPR.1992.223129. ISBN 0818628553. ISSN 10636919.

- [55] DUKE, C.; PARK, K.; EWING, R.; EWING, R. a PARK, K. *Chi-Square*. Basic Quantitative Research Methods for Urban Planners. 1. Routledge, 2020, s. 133-149. DOI: 10.4324/9780429325021-8. ISBN 9780367343255.
- [56] Derpanis G., K. *The Bhattacharyya Measure* Mendelej Computer, 2008, 1.4: 1990-1992 [cit. 2020-06-01].
- [57] YANG, Shengli, Jun GAO a Rui SUN. *Geometrically Invariant Color Image Watermarking Using Mean-Based 2D Color Histogram*. In: 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing [online]. IEEE, 2010, 2, s. 3-6 [cit. 2020-06-01]. DOI: 10.1109/N-SWCTC.2010.138. ISBN 9781424465972.
- [58] SAVELONAS, Michalis A, Ioannis PRATIKAKIS a Konstantinos SFIKAS. *Fisher encoding of differential fast point feature histograms for partial 3D object retrieval*. Pattern Recognition [online]. Elsevier, 2016, 55(C), 114-124 [cit. 2020-01-13]. DOI: 10.1016/j.patcog.2016.02.003. ISSN 0031-3203.
- [59] KIM, D. a J. PAIK. *Automatic moving object segmentation using histogram-based graph cut and label maps*. Electronics Letters [online]. Institution of Engineering and Technology, 2012, 48(19), 1198-1199 [cit. 2020-01-13]. DOI: 10.1049/el.2012.0740. ISSN 00135194.
- [60] MOBALEGH, H., L.I. HELGADÓTTIR a R. ROJAS. *Shape based round object detection using edge orientation histogram* In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [online]. Springer Verlag, 2014, 8371, s. 326-335 [cit. 2020-01-13]. DOI: 10.1007/978-3-662-44468-9_29. ISBN 9783662444672. ISSN 03029743.
- [61] LI, W., X. ZHANG, N. XIE, W. HU, W. LUO a H. LING. *Probabilistic index histogram for robust object tracking*. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [online]. 2011, 6468(1), s. 184-194 [cit. 2020-01-13]. DOI: 10.1007/978-3-642-22822-3_19. ISBN 9783642228216. ISSN 03029743.
- [62] DORAZIL, J. *Detekce změn v digitálních obrazech*. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2017, 57 listů : ilustrace + 1 CD-ROM.
- [63] PŘIBYL, J. *Scaling object tracking using Integral Histogram* In: Youtube[online]. 2017 [cit. 2017-06-01]. Dostupné z <<https://youtu.be/OLZXnB5MB68>>

Seznam symbolů, veličin a zkratek

EDM	Earth mover's distance
HSV	Odstín, Sytost, Hodnota – Hue, Saturation, Value
ICA	Nezávislá analýza složek – independent component analysis
KL	Kullback–Leibler – Kullback–Leibler
LTP	Místní ternární vzor – Local Ternary Pattern
OF	Optický tok – Optical flow
MHT	Sledování více hypotéz – Multiple Hypotheses Tracking
PDF	Funkce hustoty pravděpodobnosti – Probability density function
PCA	Principal Component Analysis — Analýza hlavních komponent
RGB	Červená, Zelená, Zodrá – Red, Green, Blue
SIFT	Velikost-Invariance transformace prvků – Scale-invariant feature transform
SVM	Metoda podpůrných vektorů – Support vector machine