VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SLEDOVÁNÍ OBJEKTŮ V PANORAMATICKÉM VIDEU

OBJECT TRACKING IN PANORAMIC VIDEO

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. VÍT AMBROŽ**
AUTHOR

VEDOUCÍ PRÁCE — **Doc. Ing. MARTIN ČADÍK, Ph.D.**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2020/2021

Zadání diplomové práce

Student: **Ambrož Vít, Bc.**
Program: Informační technologie
Obor: Informační systémy
Název: **Sledování objektů v panoramatickém videu**
Object Tracking in Panoramic Video
Kategorie: Počítačová grafika

Zadání:

1. Seznamte se s metodami pro sledování objektů ve videu.
2. Vytipujte metody vhodné pro sledování objektů v panoramatickém videu. Popište vlastnosti vybraných metod.
3. Navrhněte a implementujte systém pro sledování objektů v panoramatickém videu. Do systému implementujte vybrané metody.
4. S daným systémem experimentujte, vyhodnoťte dosažené výsledky, diskutujte možnosti budoucího vývoje a případně navrhněte vlastní modifikace implementovaných metod.
5. Dosažené výsledky prezentujte formou videa a plakátu, příp. článku.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Čadík Martin, doc. Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 30. října 2020

Abstrakt

Tato diplomová práce mapuje dosavadní vývoj v oblasti sledování objektů v panoramatickém 360° videu. Práce si klade za cíl odhalit hlavní problémy, které souvisejí se sledováním objektů a dále se zaměřuje na jejich řešení v rámci panoramatického videa. Během studia této problematiky bylo zjištěno, že dosud bylo provedeno jen velmi málo řešení pro sledování objektů v ekvirektangulární projekci panoramatického videa. V této práci jsou proto představena dvě vylepšení pro metody sledování jediného objektu založené na adaptaci ekvirektangulárních snímků. Tato diplomová práce navíc přináší vlastní manuálně vytvořený dataset panoramatických videí s více než 9900 anotacemi. Pro tento nový dataset je provedeno detailní vyhodnocení celkově 12 významných i moderních algoritmů pro sledování objektů.

Abstract

The master thesis maps the state of the art of visual object tracking in panoramic 360° video. The thesis aims to reveal the main problems related to visual object tracking and moreover focuses on their solution in panoramic videos. In the study of the existing approaches was found that very few solutions of visual object tracking in equirectangular projection of panoramic video have been implemented so far. This thesis therefore presents two improvements of object tracking methods that are based on the adaptation of equirectangular frames. In addition, this thesis brings the manually created dataset of panoramic videos with more than 9900 annotations. Finally the detailed evaluation of 12 well known and state of the art trackers has been performed for this new dataset.

Klíčová slova

360° video, panoramatické video, sférické video, sledování objektů, panoramatické projekce, OpenCV

Keywords

360° video, panoramic video, spherical video, object tracking, panoramic projections, OpenCV

Citace

AMBROŽ, Vít. *Sledování objektů v panoramatickém videu*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Ing. Martin Čadík, Ph.D.

Sledování objektů v panoramatickém videu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Doc. Ing. Martina Čadíka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Vít Ambrož
13. května 2021

Poděkování

Tímto bych velmi rád poděkoval panu Doc. Ing. Martinovi Čadíkovi, Ph.D. za přátelský přístup, cenné rady a odborné vedení při řešení této práce.

Obsah

Kapitola 1

Úvod

Možnost sledovat určitý objekt ve videu patří již velmi dlouho mezi důležité úlohy ve zpracování obrazu. Tato možnost je dosud využívána napříč různými spektry oborů, mezi které patří například robotické, dohledové či bezpečnostní systémy. Vývoj v oblasti sledování objektů je především v posledních letech velmi rozšířený a neustále vznikají nové a vylepšené metody, které se touto úlohou zabývají. Tyto metody se ovšem často vyvíjí a testují pouze pro použití v běžném videu, se kterým se v současnosti potkáváme téměř každý den. Existuje mnoho situací, kdy může být výhodné zachytit široké okolí kolem samotné kamery. Nicméně možnosti klasických kamer jsou pro takový účel velmi omezené, jelikož dokáží zachytit pouze malý úhel záběru.

Řešením pro takové scénáře může být panoramatické video, mezi jehož hlavní vlastnosti patří právě široký úhel záběru. Panoramatické video se v dnešní době stává více populárním díky vývoji kamer, které umožňují taková videa pořizovat a vytvořit. Oblíbenost panoramatických snímků se zvyšuje i u běžných uživatelů, jelikož i sociální sítě postupně přidávají podporu pro zobrazení panoramatických fotografií a videí. Vlastnost širokého úhlu záběru s sebou ovšem přináší i různé problémy. Zpracování obrazu v panoramatickém videu je oproti běžnému videu výpočetně náročnější a velká část moderních metod zpracování obrazu dosud nebyla adaptována pro použití v panoramatickém videu. Právě spojení zmíněné úlohy sledování objektů a oblasti panoramatického videa představuje hlavní podstatu této práce, ve které je tato problematika detailně popsána a diskutována. Motivací této práce je tedy skutečnost, že oblast sledování objektů ve videu je v oblasti počítačového vidění stále velmi progresivní a přímo se nabízí ji dále rozvíjet.

Tato diplomová práce je rozdělena do několika kapitol. V kapitole 2 jsou představeny obecné vlastnosti panoramatického videa a také možnosti jeho pořízení i zobrazení. Kapitola 3 přináší shrnutí vývoje metod pro detekci a sledování objektů pro běžná videa. Na tento souhrn přímo navazuje kapitola 4, kde jsou popsány možnosti řešení sledování objektů v panoramatickém videu a zmíněny i konkrétní metody, které se touto problematikou již zabývaly. Na základě nastudovaných poznatků je v následující kapitole 5 uveden vlastní návrh pro vylepšení metod sledování objektů v ekvirektangulární projekci panoramatického videa. Je zde popsána vlastní implementace a veškeré použité technologie. V kapitole 6 je představen nový dataset, který byl vytvořen pro účely evaluace výsledků této práce. Získané výsledky jsou následně porovnány pomocí metrik, které se v současné době používají pro přesné porovnání moderních metod sledování i detekce objektů. V závěrečné kapitole 7 jsou shrnuty dosažené výsledky a nastíněn možný budoucí vývoj této práce i oblasti sledování objektů v panoramatickém videu.

Kapitola 2

Panoramatické video

V této kapitole budou uvedeny základní vlastnosti panoramatického videa, které by měly být dostatečné pro pochopení problematiky, jež bude řešena v dalších kapitolách této práce. Nejprve zde budou představeny panoramatické snímky, jejichž sekvence tvoří panoramatické video. Dále budou popsány možné způsoby a zařízení, které umožňují pořídit tento typ videa. Budou zde vysvětleny také možnosti jeho zobrazení a s tím související problémy. Na konci této kapitoly bude uveden i kontext souvislostí panoramatického videa s virtuální realitou.

2.1 Panoramatické snímky

Video je prostou sekvencí po sobě následujících snímků a tato skutečnost platí i pro panoramatické video jakožto sekvenci panoramatických snímků. V literatuře i na webu se lze často setkat i s označením sférické snímky, 360° snímky nebo všesměrové (z anglického *omnidirectional*) snímky. V této práci budou tyto snímky či videa označovány jako panoramatické. Je vhodné také upřesnit, že se tato práce zaměřuje na panoramatická videa, která se skládají ze snímků obsahujících celý 360° prostor v horizontální rovině. Jako panoramatické se totiž mohou označovat i širokoúhlé snímky, jejichž úhel záběrů je menší než zmíněných 360°. V následujících částech této práce bude vždy uvedeno, zda se konkrétní popis vztahuje na jednotlivé snímky či na celé sekvence snímků, respektive videa.

Pořízení panoramatického snímku

Na úvod zde budou zmíněny hlavní způsoby pořízení panoramatických snímků, respektive panoramatických videí. Nejprve uvažujme způsob, pomocí kterého je možné statický panoramatický snímek získat při použití jediného zařízení s klasickou čočkou (mobilní telefon, běžný fotoaparát). Pomocí takového zařízení je nejprve nutné samotné snímky pořídit a celkové panoráma poté vytvořit složením těchto běžných snímků. Zachycené snímky pak odpovídají jednotlivým částem výsledného panorámatu. Tímto způsobem je ovšem vytvoření kvalitního panoramatického snímku poměrně složité a vyžaduje manuální preciznost při zachycení snímků. Pro tuto metodu je nutné použít alespoň jedno zařízení, pomocí kterého se zachytí dostatečný počet snímků pokrývajících široký úhel záběru z jednoho místa. Pro složení panorámatu lze využít například nástroje Hugin[1]. Tento způsob mohou využívat profesionální fotografové pro vytvoření velmi kvalitní a detailní panoramatické fotografie.

[1] http://hugin.sourceforge.net/

V současné době, kdy je hlavním trendem používání mobilních technologií, lze panoramatický snímek pořídit jednoduše a rychle. Pro tento účel je možné využít zařízení, které má člověk neustále při sobě, například chytrý telefon. Existuje celá řada aplikací, přičemž je hned několik z nich volně dostupných například pro uživatele androidu[2]. Uživatel může mít během pár minut k dispozici panoramatickou fotografii, kterou může následně sdílet s ostatními. Pro tyto účely totiž vzniklo velké množství webových aplikací i knihoven, které možnost zobrazení panoramatického snímku podporují[3]. Také některé sociální sítě již dříve přidali podporu pro prezentaci a prohlížení panoramatických fotografií[4].

Pomocí jediného zařízení s klasickou čočkou ovšem není možné zachytit panoramatické video. Pokud bychom chtěli vytvořit panoramatické video složením více videí, museli bychom mít dostatečný počet kamer, které by byly teoreticky ve stejném bodě a současně zabíraly prostor pro všechny směry[5]. Tento přístup může sice vést k velmi kvalitnímu výsledku, nicméně jeho nevýhodou může být počet a cena všech potřebných zařízení. Další nevýhodou je také náročnost následného zpracování při spojování videí. Využití běžných kamer zkrátka naráží na zmíněný problém úhlu záběru a snímky z jediné kamery nemohou zachytit všechny části prostoru. Tato problematika lze zřejmě řešit pořízením většího množství běžných kamer, nicméně zde dochází k nárůstu ceny za pořízení, provoz či údržbu více kamer.

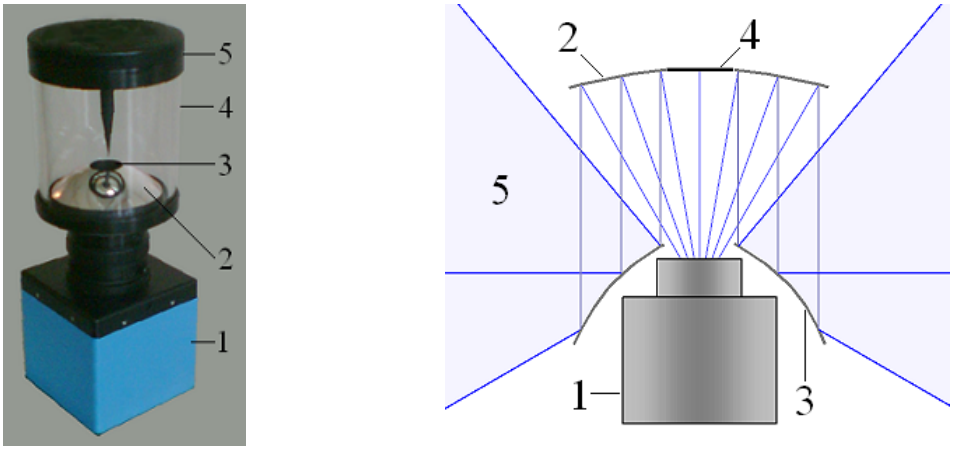
Sférické kamery

Pro účely pořízení panoramatických snímků i videa vznikla zařízení, která dokáží pokrýt celý 360° úhel záběru v horizontální rovině. Zmíněná zařízení lze označovat například jako 360° či všesměrové kamery [67], přičemž jsou tyto kamery schopny pořídit nejen statické 360° snímky, ale také 360° videa v reálném čase. Takové kamery lze v zásadě rozdělit na dva typy podle principu, který využívají pro snímání 360° úhlu záběru.

V této podsekci budou velmi stručně popsány typy těchto 360° kamer a bude uveden i kontext jejich vývoje. Bylo by jistě možné zde detailně popsat i přesné principy či parametry snímání kamer z hlediska optiky, nicméně pořízení panoramatických videí není cílem této práce. Jádrem této práce je zpracování obrazu ve videu, které je možné pomocí 360° kamer pořizovat. Z tohoto důvodu je zde uveden i tento prostý úvod vývoje 360° kamer, který by měl mimo jiné nastínit i motivaci řešení popsaného v následujících kapitolách této práce.

Na konci minulého století byly poprvé realizovány přístupy [89, 58, 61, 62], které umožnily pořízení 360° snímků pomocí tzv. katadioptrických kamer. Katadioptrické kamery kombinují princip běžné kamery společně se speciálně tvarovaným zrcadlem či zrcadly [67], které se nachází v potřebné vzdálenosti od senzoru kamery. Jednoduše řečeno, princip katadioptrického systému vychází z vlastností lomu a odrazu světla, přičemž je díky odrazu např. od parabolického zrcadla možné zachytit velmi široký úhel záběru. Samotné označení katadioptrického systému vychází ze dvou použitých oblastí – dioptriky (*dioptrics*) ve smyslu používání čočky pro snímání a optiky zabývající se odrazy a jevy zrcadlení (*catoptrics*) [75]. Katadioptrický systém nachází využití i mimo oblast kamerových zařízení, využívá se například při konstrukci teleskopů nebo mikroskopů. Příklad katadioptrické kamery je zobrazen na obrázku 2.1.

[2] https://play.google.com/store/apps/details?id=com.vtcreator.android360
[3] https://www.marzipano.net/
[4] https://facebook360.fb.com/360-photos/
[5] https://www.youtube.com/watch?v=1iRic5RZtDs



(a) (b)

Obrázek 2.1: Katadioptrická (všesměrová) kamera se dvěma zrcadly a její schéma [83] – (a) 1) kamerový systém, 2) spodní zrcadlo, 3) clona, 4) skleněné pouzdro (*glass housing*), 5) krytka, na jejíž spodní části se nachází horní zrcadlo. (b) 1) kamerový systém, 2) horní zrcadlo, 3) spodní zrcadlo, 4) rovinná část horního zrcadla (*black spot*), 5) úhel záběru zvýrazněný modrou barvou.

(a)

(b)

Obrázek 2.2: Snímek pořízený katadioptrickou kamerou[6] v (a) polární projekci a (b) v její příslušné rektifikované podobě.

[6] https://www.cs.columbia.edu/CAVE/projects/cat_cam_360/

Katadioptrická všesměrová kamera je schopna pokrýt úhel záběru celých 360° v horizontální rovině, ale kvůli její konstrukci není možné zachytit kompletní 180° prostor ve vertikální rovině. Tento typ 360° kamery tak není ryze všesměrový, jelikož oblasti nacházející se pod kamerou, respektive nad kamerou, jsou mimo úhel záběru. Výsledné snímky katadioptrické kamery jsou ve výchozím stavu obvykle zobrazeny v polární projekci (obr. 2.2a), přičemž je lze rektifikovat pomocí převodu mezi polární a kartézskou soustavou souřadnic (obr. 2.2b).

Na konci minulého století byl rovněž zmíněn i princip pořízení 360° snímků pomocí více čoček typu rybího oka [58, 61]. V této době se ovšem považovala možnost spojení více snímků typu rybího oka za velmi komplikovanou a to především kvůli tehdy dostupnému hardwaru potřebnému pro konstrukci jediné kamery. Takové zařízení by muselo umožnit, aby byl zajištěn teoretický průnik všech příchozích světelných paprsků v jediném optimálním bodě a následně by tak bylo možné přesně spojit zachycené snímky [61].

Uvedené rybí oko představuje druh širokoúhlého objektivu, který používá speciální čočku pro zachycení velmi širokého úhlu záběru (obr. 2.3a). Tyto čočky dokáží zachytit úhel typicky o velikosti alespoň 180° a to současně v horizontální i vertikální rovině. Výhodou rybího oka je velká hloubka ostrosti a možnost zaostření na velmi blízké předměty. Název tohoto typu objektivu je odvozen od perspektivy, která svým způsobem připomíná oko ryby. Na první pohled zřejmou vlastností těchto objektivů je soudkovité zkreslení snímků (obr. 2.3b), díky kterému se stalo rybí oko populární také jako specifický styl umělecké fotografie[7]. Jako rybí oko se tedy často označují přímo i samotné snímky pořízené tímto typem objektivu. Tyto objektivy byly původně vyvinuty pro uplatnění v astronomii či meteorologii [76], jelikož jsou schopny zachytit celou oblohu. V současné době ale nacházejí využití v řadě dalších oblastí, mimo jiné také právě ve sférických kamerách.

(a) (b)

Obrázek 2.3: (a) Čočka typu rybí oko [67] s ilustrací možnosti pořízení přibližně 180° úhlu záběru (*FOV - field of view*), (b) Snímek pořízený objektivem typu rybí oko[8]

Fakticky se již před desítkami let uvažovalo o principu jediné kamery, která by umožnila spojení více snímků typu rybího oka do jediného 360° snímku [61], nicméně se jej až do nedávné doby nepodařilo realizovat. Vývoj čoček typu rybího oka i vývoj v oblasti hardwaru znamenal, že se zmíněný přístup nakonec podařilo realizovat [67]. Trend vývoje nových

[7] https://pinterest.com/shari100/fabulous-fisheye-lens-creativity/
[8] https://en.wikipedia.org/wiki/File:Car_Fisheye.jpg

sférických kamer prakticky započal v minulém desetiletí[9] a v současné chvíli je zřejmé, že se tyto sférické kamery budou i nadále zdokonalovat.

Tyto sférické kamery jsou tedy založeny na dvou či více čočkách typu rybího oka. Příklady takové kamery jsou zobrazeny na obrázku 2.4. Velkou výhodou této kamery je jejich dostupnost a jednoduchost, díky které zvládne pořizovat panoramatické fotografie a videa i běžný uživatel. Sférickou kameru lze obvykle pohodlně synchronizovat například s chytrým telefonem, pomocí kterého je možné provádět nastavení snímání a případně pořídit fotografii či video dálkově pomocí bezdrátového připojení. V současné době vývoj v oblasti tohoto typu sférických kamer rychle postupuje a zaměřuje se na různé aspekty zlepšení panoramatického videa. Důraz ve vývoji se klade kupříkladu na maximální rozlišení, snímkovou frekvenci a také na stabilizaci videa.

(a) (b)

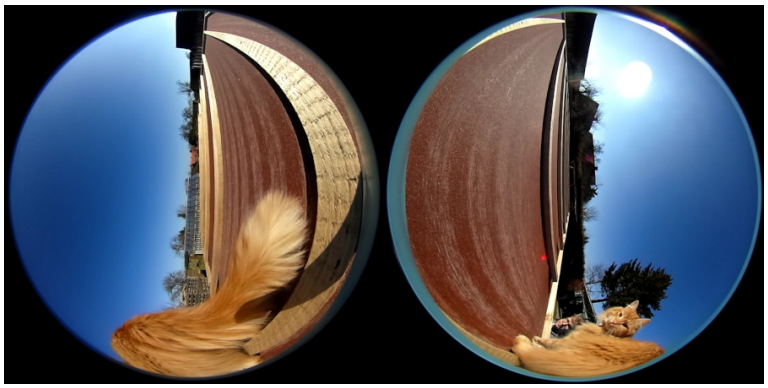
Obrázek 2.4: (a) Kamera Ricoh Theta Z1[10], která obsahuje 2 kamerové senzory a umožňuje pořídit panoramatickou fotografii v rozlišení 6720x3360 a panoramatické video v rozlišení až 3840x1920 při 29.97fps, (b) Kamera Insta360 Pro[11] umožňující pořízení 360° videa v rozlišení až 7680x3840(8K) při 30fps pomocí 6 kamerových senzorů typu rybího oka.

Hlavním principem pořizování 360° snímků je současné snímání všech směrů z jednoho teoretického bodu, což zajišťuje konstrukce kamery. Skládání snímků (*panorama stitching*) probíhá obvykle typicky až po samotném pořízení videa v rámci dalšího automatického zpracování například pomocí příslušného softwaru ke konkrétní kameře. Proces skládání může ovšem probíhat i prostřednictvím hardwaru samotné sférické kamery a lze tak například živě sledovat pořizované záběry (*live streaming*). Živé vysílání v reálném čase je ovšem většinou omezeno maximálním rozlišením nebo snímkovou frekvencí. Tyto kamery lze na rozdíl od katadioptrických 360° kamer označit jako ryze všesměrové, jelikož umožňují zachytit úhel záběru $360° \times 180°$.

Skládání snímků z těchto sférických kamer se nejčastěji provádí do ekvirektangulární nebo do kubické projekce. Vytvořená videa mají obvykle nastavena příslušná metadata a lze je spustit i v běžných videopřehrávačích (např. VLC Media Player[12]), kde je možné si zobrazit konkrétní podčást celého 360° prostoru. Obrázek 2.5 ilustruje popisovaný princip pořízení snímků typu rybího oka a jejich následného složení do ekvirektangulární projekce. Snímky pocházejí z videa, které bylo zaznamenáno pomocí vlastní sférické kamery Ricoh Theta SC[13]. Je možné si také všimnout jevu, který je zobrazen na obrázku 2.5c. V situaci

[9] https://en.wikipedia.org/wiki/List_of_omnidirectional_(360-degree)_cameras
[11] https://theta360.com/en/about/theta/z1.html
[11] https://www.insta360.com/product/insta360-pro/
[12] https://www.videolan.org/
[13] https://theta360.com/en/about/theta/sc.html

zachycené na videu je hlavním objektem kocour, jehož ocas se přiblížil velmi blízko ke sférické kameře, respektive se této kamery přímo dotkl. Právě tato skutečnost zapříčinila nedokonalé složení dvou snímků a ve výsledné ekvirektangulární projekci je jasně patrný nežádoucí šev, který jasně definuje oblasti, kde byly spojeny původní dva snímky typu rybího oka. Tento viditelný a rušivý jev vzniká nejčastěji u sférických kamer obsahujících právě dvě čočky typu rybího oka a to v situacích, kdy se konkrétní objekt příliš přiblíží ke sférické kameře.



(a)

(b)

(c)

Obrázek 2.5: (a) Dva snímky typu rybího oka, které byly pořízeny pomocí čoček sférické kamery Ricoh Theta SC. (b) Výsledný snímek videa, který byl složen z původních snímků do ekvirektangulární projekce. (c) Ilustrace problému, který vznikl v důsledku nedokonalého spojení snímků rybího oka.

2.2 Projekce a zobrazení

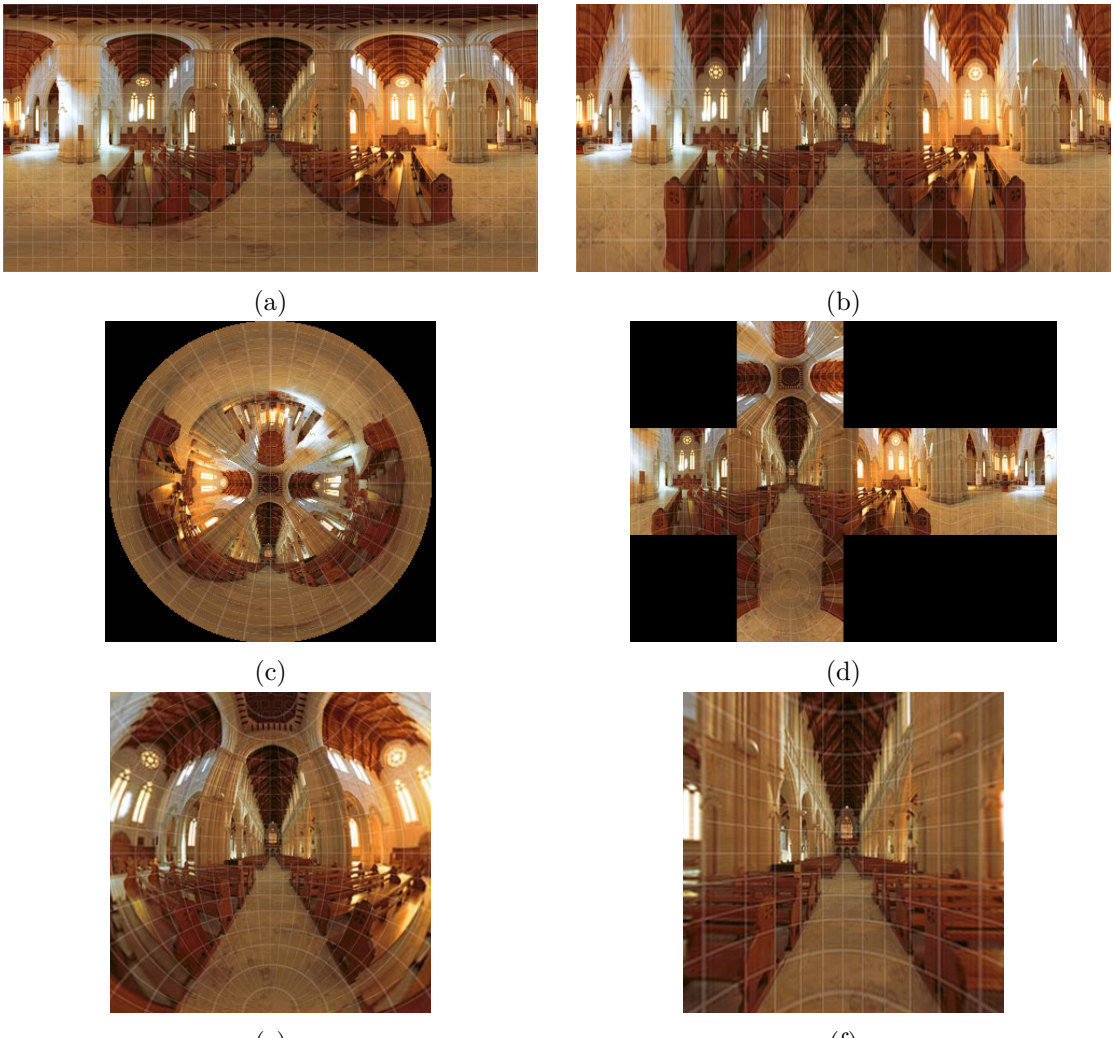
S panoramatickými snímky přichází otázka jejich zobrazení, respektive projekce. Asi nejlepší možností zobrazení 360° fotografie či videa z hlediska prostorové orientace člověka je virtuální zobrazení ve speciálním prohlížeči. V současné chvíli již existuje celá řada prohlížečů určených pro panoramatické fotografie. Tyto prohlížeče mohou být dostupné v podobě mobilní, desktopové či webové aplikace a umožňují si snímek libovolně otáčet nebo přibližovat. Existují i sítě určené speciálně pro sdílení 360° fotografií[14] a také zobrazovací knihovny pro vývojáře[15]. Virtuální způsob zobrazení snímků je tedy jistě velmi vhodný pro detailní a pohodlné prohlížení. Nicméně i 360° snímek, který je zobrazován pomocí zmíněného virtuálního prohlížeče, je v podstatě ve formátu určité projekce. Pomocí různých typů projekcí lze celý nebo téměř celý 360° snímek zobrazit do dvourozměrného prostoru, který máme k dispozici na dnešních zobrazovacích zařízeních. S podobným problémem zobrazení se lze setkat například v kartografii, kde je nutné část či celek zeměkoule reprezentovat jako dvourozměrnou mapu, přičemž zeměkoule představuje 360° snímek a mapa reprezentuje zobrazovací zařízení. Pro takový účel potřebujeme 2D projekci, která by byla schopna zobrazit rozsah až 360° v horizontální rovině a ve vertikální rovině rozsah až 180°.

V současné době je jednou z nejpoužívanějších projekcí tzv. ekvirektangulární projekce (*equirectangular projection*). Princip tohoto zobrazení si lze představit jako kompletní 360° × 180° prostor zobrazený na kouli, který je namapován do dvourozměrného obdélníkového prostoru. Formát respektive poměr stran tohoto rámečku by měl odpovídat 2:1, přičemž šířka rámečku je dvojnásobně větší než jeho výška. Na obrázku 2.6a si lze všimnout, že zkreslení je nejvyšší v oblasti stropu budovy, která je prakticky úplně rozprostřena po celé šířce rámečku. Tento fakt odpovídá tomu, že zkreslení je nejvíce výrazné poblíž pólů teoretické koule. Většina ostatních částí snímku je zakřivena sice méně, ale i tak je zde v této projekci jasně patrné radiální zkreslení. Ve dvourozměrném rámečku ekvirektangulární projekce je možné se orientovat také v běžné kartézské soustavě souřadnic [81]. Je vhodné si také uvědomit, že levý okraj rámečku v podstatě navazuje na jeho pravý okraj, respektive pravý okraj rámečku na jeho levý okraj. Jelikož ale ekvirektangulární projekce vychází ze sféry, musí být pro přesné transformace využita sférická soustava souřadnic [85]. Ekvirektangulární projekce by mohla být zařazena do skupiny cylindrických projekcí (*cylindrical projection*). Cylindrické projekce obecně zobrazují celých 360° v horizontální rovině a úhel se může lišit ve vertikální rovině. Menším úhlem záběru ve vertikální rovině lze například odstranit výrazné zkreslení na pólech (horní a dolní části), ke kterému dochází v ekvirektangulární projekci.

Mezi projekce, které dokáží zachytit celý 360° × 180° prostor, patří i kubické zobrazení (*cubic projection*). Toto zobrazení připomíná rozloženou krychli, která na svých stěnách zobrazuje odpovídající část panoramatického snímku. Každá stěna prakticky pokrývá prostor 90° v horizontálním a zároveň ve vertikálním směru. Dvourozměrná verze kubického zobrazení ovšem nemusí být zdaleka tak vhodná ke zpracování obrazu, jako zmíněná ekvirektangulární projekce. Třetí projekcí, která je schopná zachytit plné panoráma, je tzv. polární projekce (*polar projection*). Tato projekce je ve dvourozměrné podobě reprezentována kruhem, jehož poloměr odpovídá 180° ve vertikální rovině a jehož obvod v libovolné vzdálenosti od středu představuje 360° prostor v horizontální rovině. Pro transformace bodů je v této projekci využívána polární soustava souřadnic [84].

[14] https://roundme.com/explore
[15] https://photo-sphere-viewer.js.org/

Pro různé účely může být přínosné využít pouze menší část panoramatického snímku, čehož je možné docílit použitím některé další projekce. Příkladem je stereografická projekce, která může zobrazovat například záběr o velikosti 180° v horizontální rovině a až 180° i ve vertikální rovině. Pro budoucí popis řešení této práce je nutné zmínit také rektilineární projekci (*rectilinear projection*), která v ideálním případě umožňuje dosáhnout minimálního zkreslení. Rektilineární projekce tak velmi přibližně odpovídá snímku, který by byl pro zobrazenou část panoramatického snímku pořízen běžnou kamerou. Tento typ projekce se obecně může označovat také jako gnomopická projekce (*gnomopic projection*). Výčet všech zmíněných projekcí je zobrazen na následujícím obrázku 2.6.



Obrázek 2.6: Příklad 360° panoramatického snímku budovy[16] v (a) ekvirektangulárním zobrazení 360° × 180° (*equirectangular projection*), (b) cylindrickém zobrazení 360° × 120° (*cylindrical projection*), (c) polárním zobrazení 360° × 180° (*cylindrical projection*), (d) kubickém zobrazení (*cubic projection*) – 90° × 90° pro jednu stranu kostky, (e) stereografickém zobrazení 180° × 180° (*stereographic projection*), (f) rektilineárním zobrazení 110° × 110° (*rectilinear projection*).

[16] https://wiki.panotools.org/Projections

Zmíněné projekce jsou v kontextu možností zobrazování panoramatických snímků velmi důležité, a pro správné pochopení problematiky by bylo vhodné popsat projekce podrobněji, například z hlediska matematiky a souřadných systémů. Avšak pro jejich základní ilustraci a pro pochopení následující části této diplomové práce by měla být tato sekce dostačující. Existuje řada dalších projekcí panoramatických snímků, které zde zatím zmíněny nebudou. Na závěr je vhodné poznamenat, že lze provádět vzájemné převody mezi různými projekcemi pomocí matematických rovnic, které vycházejí z deskriptivní geometrie.

2.3 Panoramatické video a virtuální realita

Před uzavřením této kapitoly věnované panoramatickému videu se nabízí velmi stručně zmínit i odvětví virtuální reality. Pojem panoramatického videa, které je předmětem zájmu této práce, nelze zaměňovat s pojmem virtuální reality [86]. Tyto dvě oblasti spolu velmi úzce souvisejí, nicméně je nutné vymezit mezi nimi hned několik důležitých rozdílů. Panoramatické video a virtuální realita se především v posledních letech rozvíjejí díky pokroku technologií, mezi které patří již zmíněné sférické kamery nebo brýle pro virtuální realitu. Právě brýle pro virtuální realitu (*VR headset*) umožňují prohlížení jak panoramatického videa, tak i virtuální reality, pro kterou jsou v podstatě žádoucí výbavou. Příklad *VR headsetu* si lze prohlédnout na obrázku 2.7, přičemž toto zařízení umožňuje přenášet kromě obrazu také zvuk.

Obrázek 2.7: Headset pro virtuální realitu[17]

Hlavní odlišností je skutečnost, že panoramatické video lze pomocí *VR headsetu* pouze prohlížet z jednoho statického místa. Pokud se člověk bude pohybovat, tak se v rámci videa bude nacházet stále na stejné pozici, a to na pozici pořízení videa. Oproti tomu ve virtuální realitě má uživatel možnost se v 360° pohybovat a případně i interagovat s dostupnými elementy. S tím souvisí i výpočetní náročnost, která je pro virtuální realitu podstatně vyšší a to se projevuje například ve vyšší ceně potřebného vybavení[18]. Kromě zmíněných brýlí je pro virtuální realitu nutné mít k dispozici velmi výkonný hardware, který poskytují například moderní grafické karty. Virtuální realita je v dnešní době velmi populární pro hraní videoher, ale má mnoho dalších uplatnění v oblasti simulace reálných situací jako jsou třeba chirurgické operace nebo různé aspekty ve vzdělávání. Oproti tomu lze panoramatické video přehrát prakticky na libovolném zařízení. Jedná se pouze o specifický typ videa, jehož možnosti zobrazení byly uvedeny v předešlé sekci.

[17] https://en.wikipedia.org/wiki/File:Oculus-Rift-CV1-Headset-Back.jpg
[18] https://www.tech-tv.co.uk/360-video-vs-virtual-reality/

V souvislosti s brýlemi pro virtuální realitu se zde hodí ještě doplnit možnost úlohy a následné analýzy sledování pohybu očí v panoramatickém videu či virtuální realitě. V posledních letech se tomuto tématu věnovala řada prací a vznikly také datasety s pohyby očí a hlavy uživatelů [1, 16, 20]. Hlavním cílem výzkumu této oblasti je zjistit, jak člověk vnímá obsah panoramatického videa a na co konkrétně se v něm zaměřuje. Tento velmi krátký souhrn o virtuální realitě je zde uveden mimo jiné i z důvodu upřesnění, aby nedošlo k záměně oblasti sledování objektů (*object tracking*) a sledování pohybů očí (*eye tracking*) v panoramatickém videu. Kromě toho by ve virtuální realitě měl být každý objekt přesně definován a měla by být známa i jeho přesná pozice. Tudíž řešená úloha sledování objektů a některé další úlohy zpracování obrazu mohou na rozdíl od panoramatického videa v pravé virtuální realitě postrádat smysl a význam.

Existuje ovšem ještě oblast tzv. rozšířené reality (*augmented reality*), která umožňuje interaktivní vnímání reálného světa [80]. Rozšířená realita v podstatě kombinuje reálný svět s virtuálním světem a v tom spočívá hlavní rozdíl oproti virtuální realitě, kde se uživatel nachází pouze ve vytvořeném virtuálním světe. Svým způsobem lze rozšířenou realitu vnímat jako panoramatické video v reálném čase. Rozšířená realita by měla umožňovat přesnou registraci obrazu trojrozměrných objektů z reálného světa a právě proto je zde využití metod zpracování obrazu naprosto klíčové a žádoucí.

Se současným vývojem panoramatického videa a sférických kamer přichází potřeba řešení či vylepšování řady úloh z oblasti počítačového vidění a zpracování obrazu i pro 360° panoramatická videa. Patří mezi ně například automatická kinematografie nebo právě sledování a detekce objektů. Kromě toho se neustále vyvíjí i samotné metody zpracování obrazu, které byly dosud používané pouze pro běžná videa či snímky. Tím se zde v podstatě neustále otevírá i velký prostor pro testování a využití nejrůznějších metod v panoramatických snímcích.

Kapitola 3

Sledování a detekce objektů

Tato kapitola je věnována oblasti sledování objektů ve videu a představuje tak v podstatě základ této diplomové práce. Úloha sledování objektu ve videu patří již velmi dlouho mezi významné oblasti zpracování obrazu a může nacházet uplatnění v celé řadě oborů, například v bezpečnostních či dohledových systémech, při kompresi videa, v dopravě nebo v robotice. V této kapitole bude stručně popsáno široké spektrum existujících metod pro sledování objektů v klasickém videu. Budou zde také vysvětleny souvislosti s úzce související úlohou detekce objektů.

3.1 Sledování objektů ve videu

Sledování objektu ve videu je proces lokalizace pohybujícího se objektu ve videosekvenci během časového úseku [77]. Cílem úlohy sledování objektu ve videu je tedy zaměřit se na konkrétní objekt a dále jej na každém snímku videosekvence lokalizovat. Objekt se obvykle při sledování zároveň vizualizuje pomocí rámečku (*bounding box*), který má v nejjednodušším případě tvar obdélníku. Obvyklý scénář je takový, že se objekt označí na prvním snímku videa a na všech ostatních snímcích probíhá proces jeho sledování. Metody či algoritmy pro sledování objektů se často označují jako *trackery*, tudíž bude ve zbývající části práci rozuměno označení tracker jakožto metoda sledování objektů.

Na úvod této problematiky je vhodné si uvědomit i několik obecných principů, které při sledování objektů platí. Záměrem této úlohy je, aby byl pro aktuální snímek lokalizován sledovaný objekt, který již byl označen či lokalizován na některém z předchozích snímků videosekvence. Pro tuto úlohu je tedy klíčové mít přehled o pozici tohoto objektu na předchozích snímcích. Na základě těchto pozic se již dá odvodit řada znalostí jako rychlost a směr pohybu objektu, podle čehož se dá vytvářet i aktualizovat určitý model pohybu objektu (*motion model*) [77]. I pouhé informace o rychlosti a směru pohybu by tak v některých případech mohly vést k poměrně přesným výsledkům predikované pozice objektu.

Klíčovou informací je ovšem samotný vzhled sledovaného objektu, který je k dispozici už od prvního snímku, kde je objekt označen. Vzhled sledovaného objektu se ale ve videosekvenci často mění, a to například kvůli pohybu či rotaci objektu nebo i pohybu samotné kamery. Právě z tohoto důvodu se informace o vzhledu objektu vyplatí ukládat pro každý snímek, díky čemuž je možné si postupně vytvářet určitý model vzhledu objektu (*appearance model*). Takový model je vytvářený za samotného běhu trackeru (*online*) nebo je reprezentován samotným algoritmem, na kterém je tracker založen. Základní myšlenkou úlohy sledování objektů tak může být využití *motion modelu* pro predikci pozice sledova-

ného objektu a *appearance modelu* pro dosažení přesného výsledku. Kromě toho moderní metody sledování objektu využívají i další model, který je trénován předem (*offline*) a trackeru je tedy následně dostupný. Pro trénování takového modelu se v dnešní době využívají metody strojového učení (*machine learning*), respektive hlubokého učení (*deep learning*).

Výsledky sledování objektů

Algoritmus sledování objektů může dosahovat různých výsledků, které je možné dělit na několik základních kategorií. V nejjednodušším případě lze výsledky kategorizovat na správnou a chybnou lokalizaci sledovaného objektu. Při sledování objektu se obvykle uplatňuje také možnost vizualizace a vykreslení predikovaného rámečku (*bounding box*) kolem sledovaného objektu přímo do každého snímku videa. Díky tomu lze i pomocí lidského oka přibližně odhadnout výsledky procesu sledování, tedy zda sledování objektu probíhá správně či tracker selhává. Takový přístup hodnocení je ale velmi subjektivní, a může být užitečný nanejvýš pro ladění implementace trackeru.

Aby však bylo možné výsledky vyhodnotit objektivně, je nutné mít k dispozici referenční data (*ground-truth data*) pro každý snímek videa. To znamená, že sledovaný objekt je již předem správně označen a tudíž lze porovnávat výsledky metody sledování objektu s referenčním označením. Pro referenční označení objektu je možné použít některý automatizovaný anotační nástroj [7, 73], případně lze provést anotace manuálně. Pro účely porovnání a vyhodnocení algoritmů sledování objektu vznikla již řada datasetů s *ground-truth* daty, ze kterých vychází množství výzev a *benchmarků* [88, 45, 57, 25, 38]. V této podsekci se přímo nabízí zmínit nejznámější výzvy či datasety, jelikož se v současné době používají v podstatě jako standard pro srovnání různých trackerů.

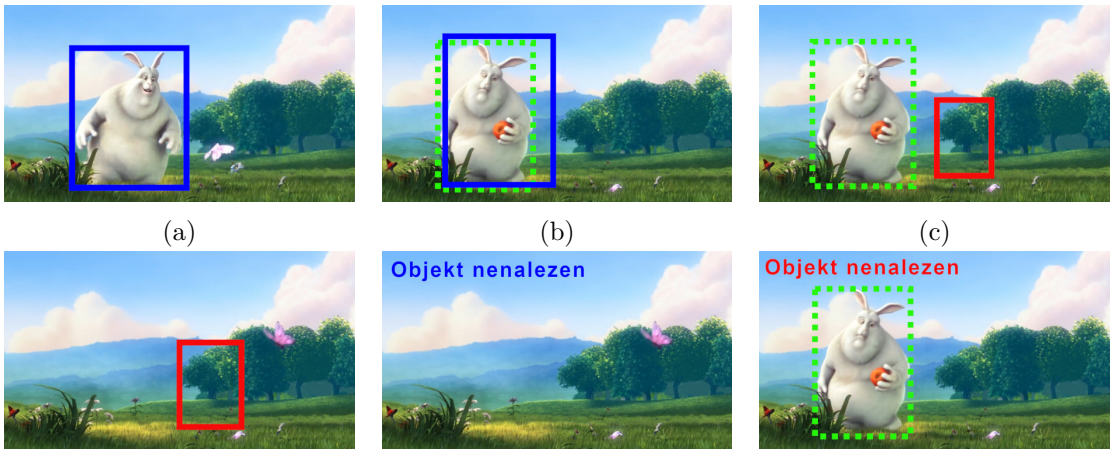
Pravděpodobně nejznámější výzvou je *VOT challenge*[1], která probíhá každým rokem již od roku 2013. Tato výzva se během let průběžně vyvíjí, přičemž jsou trackery hodnoceny v různých kategoriích a podle několika metrik, které byly mimo jiné v rámci *VOT* představeny [45]. *VOT* je organizována skupinou významných autorů, kteří se vývojem trackerů zabývali [54]. Druhou tradiční výzvou je *OTB (Object Tracking Benchmark)*[2], která prezentuje celkově dva datasety [88]. Lze se setkat s vyhodnocením trackerů na menším datasetu *OTB-50*, kterých obsahuje 50 anotovaných videí. Častěji se pravděpodobně využívá dataset *OTB-100*, který přidává dalších 50 videí a obsahuje tedy celkem 100 videosekvencí s více než 59 tisíci anotacemi.

Postupně však vznikají další významné datasety a výzvy, mezi než patří například *TrackingNet*[3] [57], *LaSOT*[4] [25] nebo *GOT-10k*[5] [38]. Tyto tři datasety obsahují převážně videa zveřejněná na YouTube[6], přičemž *TrackingNet* i *LaSOT* obsahují řádově tisíce videí a miliony anotací. Všechny zmíněné datasety obsahují různé typy objektů, neorientují se tak pouze na jediný konkrétní typ. Formát anotací se může pro každý dataset lišit, přičemž jsou do anotací často přidány i různé značky, které identifikují některé problematické aspekty sledování objektů v konkrétním videu v datasetu[7]. Díky různým značkám lze potom tracker vyhodnotit i pro jednotlivé skupiny problémů, jejichž hlavní příklady budou zmíněny v následující podsekci.

[1] https://www.votchallenge.net/
[2] http://cvlab.hanyang.ac.kr/tracker_benchmark/
[3] https://tracking-net.org/
[4] https://cis.temple.edu/lasot/
[5] http://got-10k.aitestunion.com/
[6] https://www.youtube.com/
[7] https://github.com/YuzheSHI/Benchmarks-for-Single-Object-Visual-Tracking

V těchto výzvách se využívá několik různých metrik, které obvykle vycházejí z míry překrytí výsledků trackeru s referenčními daty. Nejčastěji se lze setkat například s metrikami *Precision*, *AUC (Area Under Curve)*, *Robustness* či *EAO (Expected average overlap)*. Téma metrik přesnosti výsledků bude podrobněji představeno v kapitole 6, kde je detailně popsáno vyhodnocení provedené v této diplomové práci. Kromě přesnosti výsledků se často zkoumá či měří i rychlost trackeru, která se nejčastěji definuje na základě počtu snímků zpracovaných za sekundu – *FPS (frames per second)*.

Je vhodné si uvědomit veškeré typy výsledků, na které lze v úloze sledování objektů narazit. Sledování objektů je ve své podstatě i klasifikační úlohou, jelikož predikce trackeru prakticky určuje, zda se sledovaný objekt na konkrétním snímku nachází či nenachází. Současně s tím tracker samozřejmě určuje i pozici samotného objektu. Tracker může například predikovat pozici sledovaného objektu, přičemž se objekt fakticky na snímku vůbec nenachází (*false positive*). Naopak pokud se sledovaný objekt na snímku nachází, tracker může z nějakého důvodu selhat a žádný *bounding box* neoznačit (*false negative*). Pro neformální znázornění těchto situací byl ručně vytvořen obrázek 3.1, kde je referenční sledovaný objekt označen zelenou barvou a predikovaný *bounding box* barvou modrou, respektive červenou. V následující části práce budou videa často ilustrována formou několika snímků. Vždy bude platit, že mezi zobrazenými snímky je interval několika desítek či stovek dalších snímků. Pokud videosekvence obsahuje například ze 30 snímků za sekundu, bylo by nemožné vhodně ilustrovat průběh videa v rámci několika málo snímků, které se ve videu nacházejí bezprostředně za sebou.



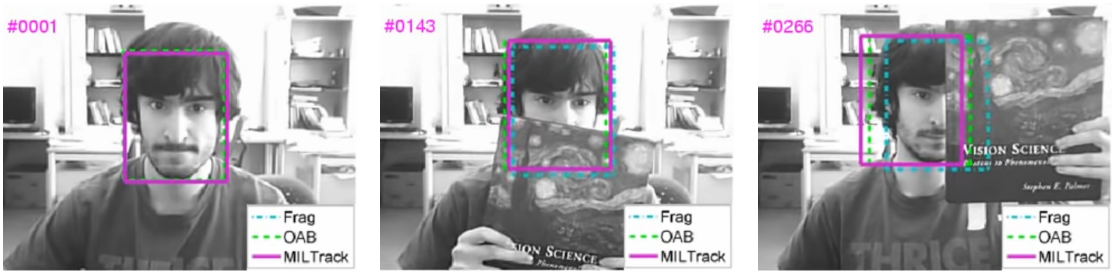
Obrázek 3.1: Demonstrace výsledků procesu sledování objektu ve videu[8]. (a) Na prvním snímku je označen sledovaný objekt, (b) Sledovaný objekt je po několika dalších snímcích označen přibližně správně podle ground-truth (*true positive*), (c) Sledovaný objekt se na snímku nachází, ale tracker jej chybně označuje mimo ground-truth oblast bez jakéhokoliv průniku (*false positive*), (d) Sledovaný objekt se na snímku nenachází, ale tracker jej chybně označuje (*false positive*), (e) Bylo správně určeno, že se sledovaný objekt na snímku právě nenachází (*true negative*), (f) Sledovaný objekt se na snímku nachází, ale tracker selhává a objekt neoznačuje (*false negative*).

[8] https://sample-videos.com/

Problémy sledování objektů

Při sledování objektů se může objevit řada situací, které mohou být pro tuto úlohu velmi problematické a mohou zapříčinit selhání trackeru. V počítačovém vidění existuje jev zvaný *occlusion* (překrytí), kdy dochází k částečnému nebo i úplnému zmizení konkrétního objektu. V případě sledování objektu ve videu může být objekt zakryt jiným objektem či překážkou. Sledovaný objekt je ovšem v případě *occlusion* stále v úhlu záběru videa, pouze v daný okamžik není část objektu vidět. Takový objekt se však může dříve či později na některém snímku ve videosekvenci znovu kompletně objevit. Většina moderních trackerů se na řešení *occlusion* zaměřuje a mají schopnost objekt lokalizovat i ve chvíli, kdy je jeho velká část zastíněna či zakryta. Úplné zakrytí (*full occlusion*) je ovšem mnohem více problematické a jen malá část trackerů se s takovým scénářem zvládne v delších videosekvencích úspěšně vypořádat.

Na obrázku 3.2 je zachycen scénář, kdy dochází k částečnému zakrytí sledovaného objektu, respektive lidského obličeje. Použité trackery si zde dokáží s *partial occlusion* poměrně úspěšně poradit, přestože velká část klíčových rysů obličeje je zakryta knihou. Pokud by byl obličej zastíněn zcela, ideální chování trackeru by mělo odpovídat selhání sledování, respektive výstupu, že se objekt na snímku nenachází. Pokud by se následně sledovaný objekt na snímku opět objevil, měl by být tracker chopen znovu začít lokalizovat sledovaný objekt.



Obrázek 3.2: Příklad problému *occlusion*, kde dochází k částečnému zakrytí lidského obličeje. Tato ilustrace byla uvedena v článku [2] i videu[9], které prezentovali tracker MIL.

Kromě problému *occlusion* se může objevit i několik dalších situací, které mohou předznamenat selhání trackeru. Podobnost s jevem *occlusion* lze najít i v situaci, kdy sledovaný objekt úplně zmizí z úhlu záběru (*Out of View*). Dalším problémem může být pro algoritmus výrazná změna osvětlení (*Illumination Variation*), jež může nastat například ve chvíli, kdy se sledovaný objekt pohybuje z tmavého prostoru na osvětlenou pozici. Problémy může způsobit také rozmazání rychle se pohybujícího sledovaného objektu (*Fast Motion*), případně i velmi rychlý pohyb kamery (*Motion Blur*). Kromě toho může sledovaný objekt rychle měnit tvar (*Aspect Ratio Change*) či svou velikost (*Scale Variation*) mezi jednotlivými snímky videosekvence. Objekt se může rovněž otáčet směrem ke kameře (*In-Plane Rotation*) nebo naopak směrem od kamery (*Out-of-Plane Rotation*). Problém může také snadno způsobit situace, kdy je na daném videu hned několik podobných objektů. Tracker může snadno začít predikovat chybné rámečky (*false positives*) odpovídajících jinému objektu, který je velmi podobný referenčnímu objektu.

Záměrně zde byly uvedeny i anglické termíny uvedených problémů, jelikož se s těmito pojmy lze setkat v rámci datasetů a výzev, které byly představeny v předchozí podsekci. Jistě by zde bylo možné výčet problémů dále rozepsat a případně ilustrovat podobně jako

[9] https://www.youtube.com/watch?v=n4QA3shA8Yw

occlusion. Jak ovšem tato práce později odhalí, tyto problémy jsou pro sledování objektů v panoramatickém videu vedlejší. Zmíněné problémy se pochopitelně mohou objevit i v panoramatickém videu, nicméně pro tento typ videa je klíčové řešit další nesnáze, aby se vůbec bylo možné přiblížit přesnosti výsledků sledování objektů v běžném videu.

3.2 Typy metod pro sledování objektů

Existuje několik možností a kritérií, podle kterých lze metody sledování objektů dělit. V této sekci budou uvedeny základní typy rozdělení, jež jasně umožňují definovat charakter trackeru. Vývoj trackerů probíhá již po několik desetiletí a vytvořit tak jeho detailní souhrn by bylo velmi složité. V této části bude proto uveden pouze stručný souhrn vývoje těchto metod, přičemž bude nastíněn i princip několika významných trackerů, pro které byly v této práci testována navržená vylepšení.

Metody podle počtu sledovaných objektů

Pravděpodobně nejvýše postaveným kritériem sledovacího algoritmu je počet objektů, které mají být ve videu sledovány. Je možné sledovat pouze jediný objekt (*SOT – Single Object Tracking*), případně více objektů současně (*MOT – Multiple Object Tracking*). Od počtu sledovaných objektů se v podstatě odvíjí i samotný princip trackeru. Tato práce je úzce zaměřena pouze na sledování jediného objektu (*SOT*), nicméně zde bude velmi krátce popsána i oblast sledování více objektů (*MOT*).

Při sledování jediného objektu je proces typicky stejný a odpovídá popisu, který byl již na začátku kapitoly uveden. Na prvním snímku videa je označen objekt a cílem sledování je tentýž objekt na všech ostatních snímcích lokalizovat a případně i vizualizovat ohraničující rámeček (*bounding box*). Typickým rysem velké části algoritmů *SOT* je možnost sledovat libovolný objekt, jehož typ není předem znám. Existující ovšem i metody, které se zaměřují pouze na sledování konkrétního typu objektu (např. lidských obličejů). Kromě označení *SOT* (*Single Object Tracking*) se pro tuto skupinu používá někdy i název *VOT*[10] (*Visual Object Tracking*).

Vývoj v oblasti sledování objektu má za sebou již dlouhou historii. Již na konci minulého století vznikaly metody a implementace, které se touto úlohou zabývaly. V této zprávě není příliš velký prostor na to, aby mohly být detailně popsány všechny důležité metody sledování objektů. Bude zde uvedena pouze malá část metod, které jsou významné z pohledu této práce a bude velmi stručně nastíněn jejich princip. Tradičním zástupcem je tracker KLT [53, 72, 68], pro který vznikly i další verze a implementace [10]. Jedná se v podstatě o techniku extrakce příznaků a registraci obrazu, kde je cílem minimalizovat rozdíly vzdáleností nalezených příznaků mezi dvěma snímky [53]. Tracker KLT představoval základ i pro některé novější trackery (např. pro tracker TLD [42]), ale i pro další oblasti ve zpracování obrazu.

Doposud byla realizována implementace několika významných trackerů i do populární knihovny *OpenCV*[11]. V kapitole 6 budou uvedeny veškeré trackery, které byly použity pro vyhodnocení provedené v této práci. Patří mezi ně mimo jiné i 5 trackerů, které jsou součástí rozšiřujících modulů knihovny *OpenCV* a proto zde budou tyto trackery stručně popsány.

[10] https://paperswithcode.com/task/visual-object-tracking
[11] https://github.com/opencv/opencv_contrib

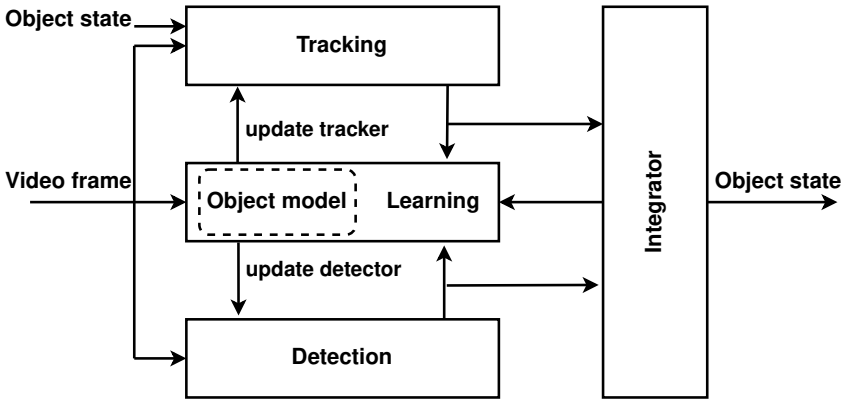
Nejstarším trackerem z knihovny *OpenCV* je tracker BOOSTING [31], který využívá *online* způsobu trénování klasifikátoru založeného na metodě strojového učení *AdaBoost*. Zvolený *bounding box* sledovaného objektu představuje pozitivní vzorek a kromě něj je vygenerováno i množství rámečků reprezentujících pozadí, které představují negativní vzorky pro trénování klasifikačního modelu. Klasifikace je prováděna postupně pro každý snímek videa a to pro pixely v blízkém okolí *bounding boxu* z předchozího snímku. Jako výsledný rámeček je pro každý snímek vybrán jednoduše ten, který dosáhl nejvyšší skóre na základě zmíněné klasifikace. Během vývoje však použití trackeru BOOSTING postupně ztratilo potenciál vzhledem k tomu, že vznikly další pokročilé trackery, které pracují na podobných principech (například MIL a KCF [2, 37]).

Tracker MIL [2] dále přidává i blízké okolí označeného objektu pro vytvoření většího množství pozitivních vzorků. Pro trénování klasifikačního modelu se zde nepřidávají jednotlivé vzorky, ale celé množiny pozitivních i negativních vzorků (označované jako *bag*). Oproti trackeru BOOSTING [31] se MIL umí lépe vypořádat s částečným zakrytím objektu (*occlusion*). Tato dvojice trackerů se ovšem velmi těžko umí vypořádat s úplným zakrytím objektu a jeho následným návratem.

Další známý přístup představil sledovací algoritmus označovaný jako MedianFlow [41]. Tento tracker staví na lokalizaci objektu v čase jednak samozřejmě dopředně (*forward*) a jednak také zpětně (*backward*). Na základě tohoto přístupu se zde poté využívá míry zvané *ForwardBackward*. Cílem je dosáhnout minimální hodnoty chyby míry *ForwardBackward*, která eliminuje nesprávné možnosti lokalizace objektu. Pro samotnou lokalizaci objektu se zde využívá principu KLT trackeru. MedianFlow velmi dobře detekuje zmizení objektu ze snímku (*true negative*) a jen zřídka predikuje *false positive* výsledky. Jeho přesnost je ovšem dobrá především pro pomalu pohybující se objekt, nebo objekt se snadno predikovatelnou trajektorií pohybu. Ani tento tracker se ovšem nedokáže úspěšně vypořádat s úplným zakrytím sledovaného objektu (*full occlusion*).

Autoři MedianFlow trackeru [41] následně navázali komplexnější metodou TLD (*Tracking-Learning-Detection*) [42]. Samotný název trackeru napovídá, že se tato metoda skládá ze tří modulů. Prvním je modul pro sledování (*tracker*), který jednoduše odhaduje pozici sledovaného objektu na základě jeho pohybu a prakticky se jedná pouze o vylepšení přístupu představeného u trackeru MedianFlow. Druhý modul je zde označovaný jako detektor, a na rozdíl od trackeru predikuje výsledky nezávisle pro každý jednotlivý snímek. Dalším rozdílem je, že detektor prohledává celý snímek, zatímco tracker predikuje pouze v malé části snímku. Je zde zároveň i modul učení (*learning*), který odhaduje chyby detektoru a na základě *online* tréninku postupně vylepšuje oba předchozí moduly. Princip této metody je přesně ilustrován na blokovém diagramu 3.3. Tracker TLD tedy kombinuje sledování s detekcí objektu, což může řešit problémy úplného zakrytí objektu. Tato skutečnost je ovšem kompenzována na úkor rychlosti, která je oproti dosud zmíněným trackerům výrazně nižší. Za zmínku stojí také skutečnost, že se další verze TLD trackeru (lze jej nalézt pod označením *Predator*) prosadily i v komerční sféře[12].

[12] http://www.tldvision.com/tld2.html



Obrázek 3.3: Blokový diagram metody TLD [42]

Dalším významným algoritmem je tracker MOSSE (*Minimum Output Sum of Squared Error*) [9]. Tento tracker je v podstatě průkopníkem metod sledování objektu, které využívají princip korelačních filtrů (*correlation filter*). Je zde využito techniky vzájemné korelace *cross-correlation* pro účely lokalizace sledovaného objektu. Taková korelace odpovídá provedení konvoluce mezi aktuálním snímkem a filtrem (někdy bývá označován jako *kernel*). Tracker MOSSE [9] využívá i Rychlé Fourierovy transformace (FFT), která výrazně urychluje dobu výpočtu a právě rychlost je velkou výhodou tohoto trackeru. Myšlenkou použití korelačních filtrů s řadou vylepšení pro sledování objektu se následně inspirovaly i další trackery. Za jeho vylepšené nástupce se dají považovat například trackery KCF [37] či CSR-DCF [54].

Právě tracker KCF (*Kernelized Correlation Filters*) [37] kombinuje využití korelačních filtrů s přístupem klasifikátorů ze zmíněných metod BOOSTING [31] a MIL [2]. KCF je značně pomalejší než tracker MOSSE, ovšem i tak umožňuje velmi rychlé sledování objektu dostatečné pro využití v reálném čase. Právě využití korelačních filtrů se ukázalo být poměrně klíčové pro další vývoj v oblasti sledování objektů. Příkladem budiž tracker CSR-DCF [54], který využívá pokročilého přístupu tzv. *Discriminative Correlation Filters*. Implementace tohoto trackeru je v *OpenCV* nazvána jako CSRT a je tak dosud nejnovějším trackerem v této nejznámější knihovně počítačového vidění.

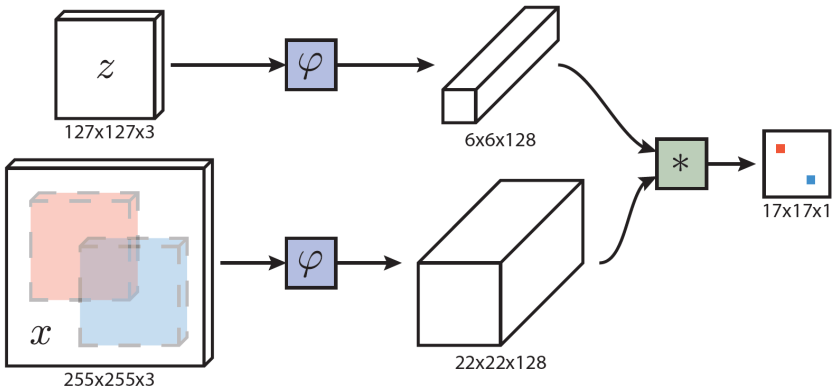
Pro *OpenCV* trackery bylo dosud provedeno několik porovnání, které bylo popsáno v odborných článcích [40], ve webových tutoriálech[13] nebo ilustrováno formou videa[14]. Je vhodné podotknout, že některé implementace uvedených trackerů nepodporují adaptivní velikost *bounding boxu*, což může výrazně zhoršit přesnost výsledků. Tato diplomová práce později uvede i vlastní porovnání 5 *OpenCV* implementací trackerů pro jejich použití v panoramatickém videu. Již v tuto chvíli je ale možné naznačit, že *OpenCV* implementace nejsou zdaleka dokonalé a neposkytují potřebnou přesnost. Výrazně lepší přesnosti je možné dosáhnout díky trackerům, jež vznikly během posledních let.

[13] https://learnopencv.com/object-tracking-using-opencv-cpp-python/
[14] https://www.youtube.com/watch?v=61QjSz-oLr8

Aktuální stav poznání (*state-of-the-art*) v oblasti sledování objektů je ovlivněn jednak přístupy založenými na korelačních filtrech a jednak také metodami hlubokého učení (*deep learning*). Nejnovější algoritmy sledování objektů lze kategorizovat například podle toho, zda algoritmus využívá korelační filtry či nikoliv [27]. Kromě toho je možné trackery kategorizovat i na základě příznaků (*features*), které využívají pro reprezentaci vzhledu objektu. Dříve se nejvíce využívali například příznaky zvané *SIFT (Scale-Invariant Feature Transform)*, *HOC (Histogram Oriented Gradients)* či *LBP (Local Binary Pattern)*. V současné době se ale trend pro výběr příznaků vzhledu objektů orientuje právě na hluboké učení a s tím souvisí i příznaky, které lze označit jako hluboké (*deep features*). Pro extrakci takových příznaků existuje řada metod, přičemž mezi nejznámější patří konvoluční neuronové sítě (*CNN*) [69], rekurentní neuronové sítě (*RNN*) [32] či reziduální sítě (*ResNet*) [35].

Přesná kategorizace trackerů je velmi obšírná, jelikož na výzkumném poli v oblasti sledování objektů dochází v posledních letech k velkému pokroku, který přináší řadu nových metod. Pro mapování tohoto rychlého vývoje byly již sepsány odborné články [27] a provedeny různá porovnání [57]. Velmi dobré rozdělení trackerů (*SOT*) lze dále nalézt například na těchto zdrojích zdroji[15][16]. Přestože takto rozsáhlý stav nelze v této práci podrobně shrnout, zaslouží si zde zmínit alespoň hlavní skupiny algoritmů pro sledování objektů.

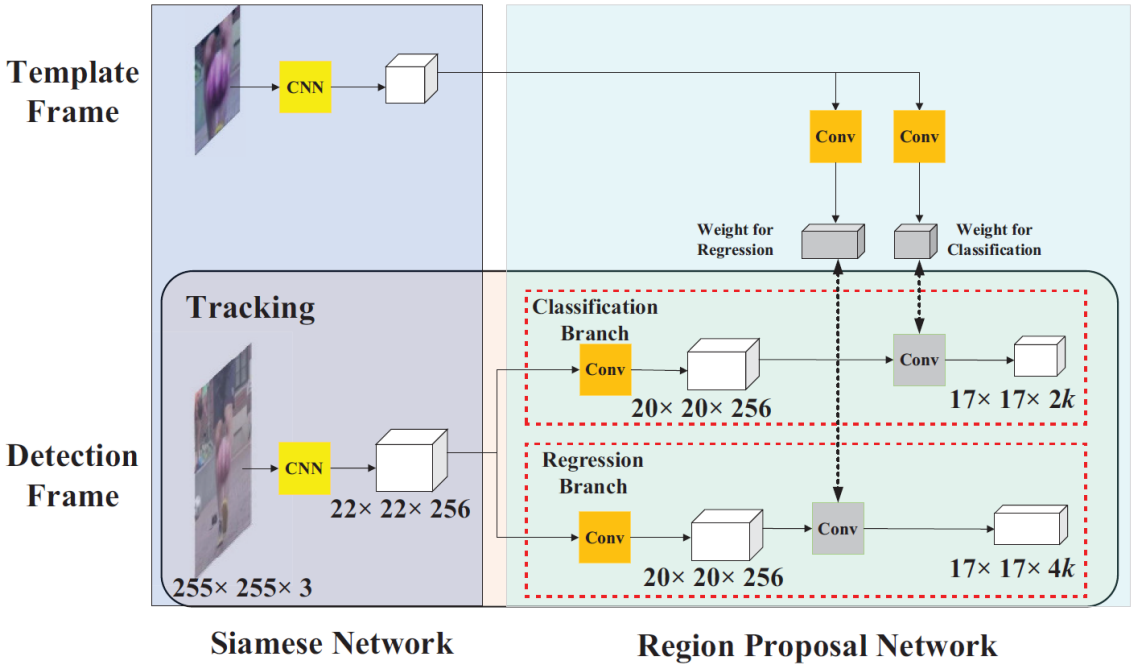
Jednou z nejvýznamnějších skupin jsou metody založené na tzv. siamských sítích (*Siamese Networks*). Myšlenka využití siamských sítí pro účely sledování objektů se poprvé objevila v roce 2016 [3] a od té doby ji převzala či dále rozvinula řada výzkumných skupin autorů [47, 46, 74, 95, 93]. Hlavní princip těchto algoritmů spočívá ve funkci *f(z,x)*, která dokáže určit míru podobnosti (*similarity score*) mezi referenční oblastí z a oblastí x na aktuálním snímku. Použitím této funkce pro porovnání sledovaného objektu a každého regionu na snímku lze získat výsledek, kterým je region dosahující nejvyšší míry podobnosti podle zmíněné funkce. Míra podobnosti je vypočítána na základě výsledku získaného z konvoluční neuronové sítě (*CNN*) pro oblast x a výsledku z druhé identické *CNN* pro referenční oblast z. Označení siamská sít je tedy odvozeno právě z použití dvou identických neuronových sítí. Obrázek 3.4 ilustruje schéma siamské sítě navržené v metodě SiamFC [3].



Obrázek 3.4: Schéma siamské sítě představené v metodě SiamFC [3]. Proměnná z je extrahovaná oblast z prvního snímku videa obsahující sledovaný objekt a proměnná x představuje aktuální snímek videa. Vzor z i aktuální snímek x je nejdříve zpracován stejnou CNN a poté je provedena operace konvoluce, podle které je vypočtena výsledná podobnostní mapa definující pozici sledovaného objektu.

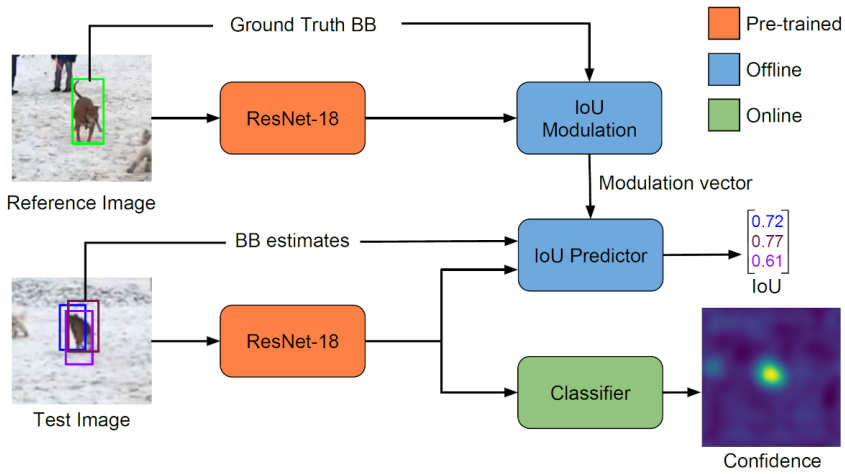
[15] https://github.com/YuzheSHI/Single-Object-Visual-Tracking_A-Paper-List
[16] https://github.com/foolwood/benchmark_results

Například tracker SiamRPN [47] přidává dále vylepšení v podobě tzv. *RPN (Region Proposal Network)*. Siamská síť je zde velmi podobná síti uvedené v SiamFC [3]. Kromě toho je zde využita síť pro návrh možných regionů (*RPN*), jež by měly odpovídat cílovému *bounding boxu*. Klasifikační větev je určena pro odlišení významných objektů a vedlejších částí (pozadí). Regresní větev je využita pro lokalizaci přesné pozice sledovaného objektu. V případně SiamRPN [47] je zmíněná *RPN* inspirována metodou detekce objektu FasterR-CNN [66], přičemž zde nedochází k *online* fázi trénování za běhu trackeru. Tento proces sledování označují autoři jako *One-Shot detection*. Na obrázku 3.5 si lze všimnout popisované architektury trackeru SiamRPN [47]. Kromě SiamFC [3] a SiamRPN [47] vznikly i další verze trackerů, které se tento přístup sledování objektů snažili dále zdokonalit – například SiamRPN++ [46], SiamMask [74], DaSiamRPN [93] nebo SiamDW [95]. Společným rysem těchto metod je jejich rychlost, jelikož jejich implementace umožňují na výkonných *GPU (Graphics Proccesing Unit)* dosahovat rychlosti potřebné pro použití v reálném čase. Například tracker SiamRPN [47] dosáhl v rámci *VOT2017 realtime challenge* rychlosti 160 FPS.



Obrázek 3.5: Architektura trackeru SiamRPN [47]. V levé části se nachází siamská síť pro extrakci příznaků. Vpravo se nachází *RPN (Region Proposal Network)*, která má dvě výstupní větve – jednu pro klasifikaci a druhou pro regresi. Predikovaný rámeček je založen na párové korelaci výsledků získaných z obou větví.

Poslední přístup sledování objektů, který zde bude uveden, reprezentuje tracker ATOM (*Accurate Tracking by Overlap Maximatization*) [18]. Tento tracker se skládá ze dvou modulů. Prvním z nich je tzv. *Target estimation* modul, který je trénován předem (*offline*) a jeho cílem je predikovat možné rámečky na základě metriky IoU (*Intersection over Union*). Tato metrika bude podrobněji popsána v kapitole 6 a v tuto chvíli stačí pouze nastínit, že *IoU* představuje míru překrytí mezi navrženým rámečkem a referenčním objektem definovaným na prvním snímku. Druhou částí je klasifikační modul, jehož učení probíhá online a využívá technik pro rozlišování mezi objektem oproti jeho pozadí. Úloha sledování objektů je zde tedy rozdělena na dva základní podproblémy – proces odhadování pozice a klasifikace. ATOM [18] si klade za cíl dosažení vysoké přesnosti, což dokazuje i jeho porovnání oproti metodám založených na siamských sítích [47, 93]. Je to dáno především využitím *online* klasifikátoru, což ovšem přináší i výrazné snížení rychlosti procesu sledování. I tak autoři uvádějí, že tento jejich implementace dosahuje rychlosti vyšší než 30 FPS na grafické kartě GeForce GTX-1080 [18].



Obrázek 3.6: Schéma architektury trackeru ATOM [18]. Pro oba moduly tohoto trackeru je přidáno rozšíření v podobě reziduální sítě (*ResNet-18*) potřebné pro extrakci příznaků. *Target estimation* modul je reprezentován neuronovou sítí, která je předem natrénována (*offline*). Následně je tento modul využit pro odhad rámečku podle míry překrytí s referenčním rámečkem podle metriky *IoU*. Klasifikační modul je zde reprezentován rovněž neuronovou sítí, jejíž učení probíhá za běhu trackeru (*online*). Cílem klasifikátoru je přesné oddělení sledovaného objektu od jiných objektů či pozadí, čehož je možné dosáhnout díky extrakci příznaků pro aktuální snímek.

Tracker ATOM byl vytvořen skupinou autorů, kteří již dříve představili například tracker ECO [17]. Tato výzkumná skupina v současné době postupně navazuje vývojem dalších metod sledování objektů – za zmínku stojí trackery DiMP [5] a KYS [6]. Kromě toho autoři vytvořili framework[17], díky kterému lze vyvíjet nové trackery, trénovat neuronové sítě a případně provádět i jejich vyhodnocení. Právě ve formě představené tímto frameworkem byly zveřejněny i oficiální implementace zmíněných trackerů [17, 18, 5, 6]. Kromě toho mohou i další autoři na tomto frameworku postavit své vlastní implementace[18].

[17] https://github.com/visionml/pytracking
[18] https://github.com/594422814/TransformerTrack

Sledování více objektů současně (*MOT*) je pokročilejší a komplexnější úloha než sledování jediného objektu (*SOT*). Sledovací algoritmus by měl být schopen pro každý snímek nalézt všechny objekty, jež musí následně lokalizovat. Algoritmus by tedy měl být schopen určit počet sledovaných objektů, přičemž tento počet není obvykle předem nijak specifikován. Zároveň by měl na po sobě následujících snímcích vždy zachovávat identitu sledovaných objektů. V současné chvíli se pro *MOT* využívá paradigma *Tracking-by-Detection*, tedy je zde využíváno algoritmů pro detekci objektů. Kromě sledování se tak současně udržuje informace o třídě či typu nalezených objektů. Krátký souhrn důležitých rozdílu sledování detekce objektů bude uveden v následující sekci 3.3. Demonstraci průběhu sledování více objektů současně je možné vidět na obrázku 3.7.

Obrázek 3.7: Ukázka ideálního průběhu sledování více objektů současně[19]. Ve videu se nacházejí lidé a dopravní prostředky na rušné ulici.

Evaluace *MOT* se velmi odlišuje od evaluace *SOT*. Výsledky totiž závisejí z velké míry na výsledcích použitého algoritmu pro detekci objektů. Nastává proto také otázka, zda lze objektivně vyhodnocovat *MOT* trackery s jednotným algoritmem detekci, či vyhodnocovat celé systémy sledování objektů s metodou detekce, pro kterou byly vytvořeny. Od roku 2015 je asi nejvíce používaným evaluačním prostředkem pro vyhodnocení této kategorie trackerů *Multiple Object Tracking Benchmark*[20]. Používají se zde také specifické metriky, například *MT (Mostly Tracked)*, *PT (Partially Tracked)*, *ML (Mostly Lost)* nebo komplexní metrika přesnosti *MOTA (Multiple Object Tracking Accuracy)*.

Jako zástupce *MOT* je možno uvést například trackery SORT [4] a navazující DeepSORT [87], které se zaměřují na sledování více objektů v reálném čase. Tyto trackery využívají metodu detekce založenou na konvolučních neuronových sítích FasterR-CNN [66]. Tato oblast se stále rychle vyvíjí společně s oblastí samotné detekce objektů. Tato diplomová práce je však zaměřena pouze na metody sledování jediného objektu a proto zde již *MOT* nebude dále poprobněji popsáno.

[19] https://motchallenge.net/vis/MOT16-13/gt/
[20] https://motchallenge.net/

Metody podle účelu

Kromě dělení metod podle počtu sledovaných objektů, mohou být metody děleny například podle jejich zaměření či účelu. Algoritmus sledování objektů může být zaměřen na robustnost, kdy je cílem sledovat objekt po dlouhou dobu bez selhání procesu sledování. Takové algoritmy se označují jako dlouhodobé (*long-term*) a měly by mít schopnost lokalizovat objekt i po jeho několikanásobném úplném zakrytí (*occlusion*), případně jeho zmizení a zpětném návratu do úhlu záběru. Druhou skupinou jsou trackery, které mohou být označeny jako krátkodobé (*short-term*). Tyto trackery jsou orientovány spíše na použití v kratších videosekvencích, kde obvykle stačí řešit pouze problémy částečného zakrytí (*occlusion*). Rozdělení na dlouhodobé a krátkodobé trackery bylo zapříčiněno skutečností, že první datasety pro evaluaci trackerů obsahovaly většinou pouze velmi krátké videosekvence. Až později se přidalo i vyhodnocení pro dlouhé videosekvence, což vedlo k označení *long-term*.

Zmíněné kategorie byly navrženy a víceméně standardizovány především v rámci *VOT challenge* [45]. Každý rok jsou pro výzvy *VOT* aktualizovány různorodé datasety, na kterých jsou provedeny evaluace desítek nových algoritmů sledování objektů. Právě v těchto výzvách lze nalézt kategorii pro *long-term* i *short-term* trackery. Například v *VOT2020 challenge* bylo uvedeno 5 různých kategorií, v nichž byly trackery porovnány[21]. Jedním ze specifických účelů je poté použitelnost trackeru v reálném čase, kde se zde bere v potaz rychlost trackeru v kombinaci s přesností jeho výsledků. Pojem použitelnosti v reálném čase se pochopitelně odvíjí od snímkové frekvence pořízeného videa, jejíž hodnota může být kupříkladu 30 snímků za sekundu (FPS).

Příklady konkrétních metod pro sledování byly již uvedeny v předchozí podsekci. Kategorizovat zástupce metod na *short-term* a *long-term* není úplně jednoduché, jelikož většina zmíněných metod nebyla vytvořena striktně pro konkrétní účel. Obvykle se pro tyto účely porovnávají až následně prostřednictvím zmíněných výzev. *Long-term* trackery obvykle dosáhnou dobré úspěšnosti i v *short-term* úloze, ovšem obvykle na úkor rychlost oproti *short-term* metodám. Příkladem *long-term* trackeru budiž tradiční tracker TLD [42] nebo tracker SiamDW [95], které byly v této kategorii v rámci *VOT* vyhodnoceny. Dnešní trackery mohou [18, 5] dosahovat poměrně přesných výsledků v obou kategoriích. Závěrem této sekce je vhodné sdělit, že současný stav vývoje sledování objektů zdaleka není na takové úrovni, aby bylo možné dosáhnout absolutní úspěšnosti pro libovolný objekt v libovolné videosekvenci [40]. Prostor pro zlepšení přesnosti v každé z kategorií trackerů je stále velmi velký a dá se očekávat, že se sledování objektů bude v blízkých letech neustále rozvíjet.

Veškeré zmíněné metody sledování objektů byly v této sekci popsány opravdu velmi krátce a stručně. Pro jejich přesnou interpretaci je vhodné vždy nahlédnout do samotného článku, ve kterém byla metoda detailně popsána. V následující kapitole 4 budou uvedeny možnosti sledování objektů v panoramatickém videu. Kromě toho se dále v kapitole 5 ukáže, že tato práce přináší univerzální vylepšení, která na libovolný tracker nahlíží jako na černou skříňku (*black box*). Tato vylepšení tedy lze pro konkrétní tracker využít i ve chvíli, kdy nemáme detailní znalost o principu fungování daného trackeru.

[21] https://www.votchallenge.net/vot2020/

3.3 Detekce objektů

V rámci této kapitoly by bylo vhodné zmínit i rozdíly mezi sledováním objektů (*object tracking*) a detekcí objektů (*object detection*) ve videu. V oblasti počítačového vidění spolu obě tyto úlohy velmi úzce souvisejí. Předmětem zájmu této práce je úloha sledování objektů, nicméně detekce se v řadě sledovacích algoritmů objevuje také [42, 87]. Právě z tohoto důvodu je nutné správně interpretovat smysl a motivaci úlohy detekce objektů.

Rozdíly sledování a detekce objektů

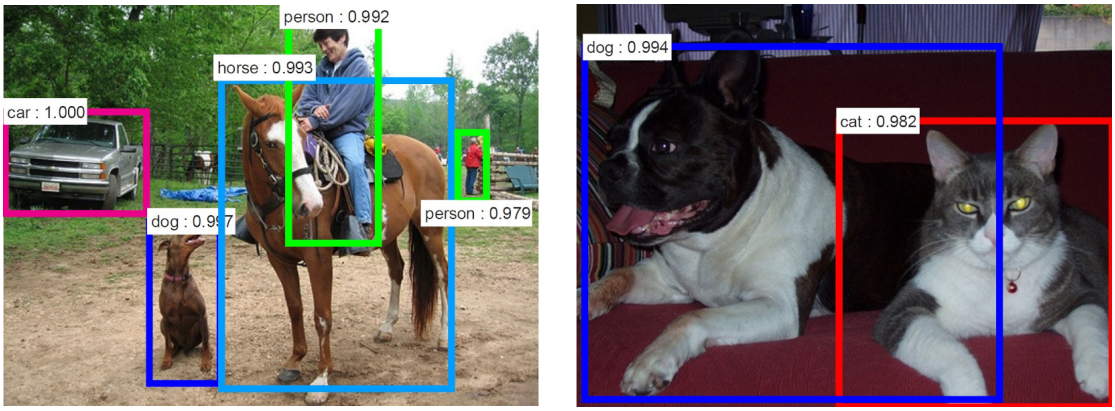
Jak již bylo zmíněno dříve, sledování objektu ve videu je proces, kdy je vybrán objekt, který je na všech po sobě jdoucích snímcích sledován a lokalizován. Tento proces tedy dává smysl pouze pro video, tedy sekvenci snímků. Sledování objektu má poznatky o tom, zda se objekt na videu pohybuje, případně kam se pohybuje, jak rychle se pohybuje nebo jaký má tvar či velikost. Naproti tomu detekce objektu je vyhledávání a lokalizace konkrétního objektu na jediném statickém snímku [30]. Algoritmy pro detekci objektů by měly být schopny nalézt konkrétní typ objektu v libovolném statickém obrázku bez předchozí znalosti o pozici tohoto objektu. To ovšem neznamená, že by detekce nebyla použitelná pro videa. V úvodu práce již bylo uvedeno, že video je sekvence snímků a tedy detekce může probíhat pro každý jednotlivý snímek videa. Nabízí se tedy možnost řešit problém sledování objektu tak, že pro každý snímek videa bude použit algoritmus pro detekci vybraného objektu.

Úloha sledování objektů si neklade za cíl rozpoznat, o jaký typ objektů se jedná. Detekce by naopak měla být schopna konkrétní objekt lokalizovat právě na základě rozpoznání konkrétního typu či třídy objektu (viz obrázek 3.8). Schopnost detekce konkrétního typu objektu s sebou přináší požadavek, aby měl algoritmus pro detekci objektů co nejlepší znalost o tomto typu či třídě objektu, například zda se jedná o člověka, zvíře či věc. Touto znalostí jsou myšleny obvykle atributy vzhledu objektu, respektive jak může objekt vypadat z různých úhlů. Pro potřeby detekce musí být tedy objekt vždy určitým způsobem klasifikován. V současné době se obvykle používají moderní metody strojového učení pro vytvoření modelu, který klasifikaci pro zvolené typy objektů umožňuje.

V počítačovém vidění existují i další konkrétní úlohy, které jsou s detekcí objektů úzce spojeny. Jedná se například o oblast segmentace obrazu (*image segmentation*), která umožňuje označit či rozdělit snímek přesně podle hranic přítomných objektů, zatímco u detekce objektů je obvykle dostačující objekt označit pomocí obdélníkového *bounding box*. Dalším rozšířením úlohy detekce může být i rozpoznávání obličejů (*face recognition*), které nachází uplatnění například v biometrických systémech. Taková úloha musí kromě detekce obličeje i přesně rozpoznat a identifikovat detekovanou tvář.

Použití detekce objektů v každém snímku videa pro úlohu sledování objektů je obvykle výpočetně náročnější než sledování objektu bez použití detekce. Metoda pro detekci totiž nemá k dispozici údaje o pozici objektu na předchozích snímcích či změnách jeho pohybu. I přesto se v některých typech metod tento přístup používá, což bude později uvedeno. Řada metod obě úlohy kombinuje a detekci používá například po určitém intervalu snímků pro verifikaci, že proces sledování objektu probíhá správně a zda se označený *bounding box* neodchyluje od sledovaného objektu. Naopak metoda sledování může metodě detekce objektu ve videu pomoci vypořádat se do určité míry se zakrytím objektu (*occlusion*). Kromě toho sledování umožňuje zachovávat identitu objektů na rozdíl od metod detekce[22].

[22] https://www.youtube.com/watch?v=SsiHH_wrwDg



(a) (b)

Obrázek 3.8: Znázornění výstupu algoritmu pro detekci, který dokáže detekovat různé typy či třídy objektů [66]. Kromě rámečků je zde zároveň uvedena i hodnota míry příslušnosti detekovaného objektu do konkrétní třídy objektů.

Metody detekce objektů

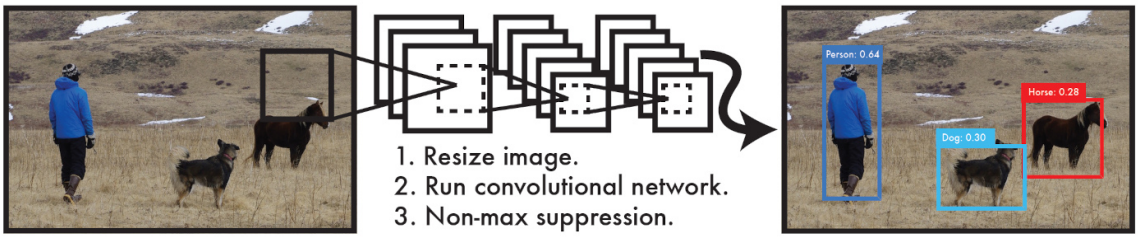
V posledních letech dosáhl vývoj v oblasti detekce objektů výrazného zlepšení především díky rozvoji neuronových sítí a hlubokého učení. Vznikla řada algoritmů detekce, které jsou dále postupným rozvojem dále zdokonalovány[23]. Lze tvrdit, že výzkum algoritmů detekce objektů je v současnosti ještě více rozšířený než výzkum v oblasti sledování jediného objektu (*SOT*). Existuje velké množství porovnání (*benchmarks*) právě pro vyhodnocení úspěšnosti detekce objektů (Microsoft COCO[24] [50]). Detektory se obvykle hodnotí především podle dvou kritérií – podle rychlosti a přesnosti. Rychlost detekce se obvykle udává v čase pro zpracování jednoho snímku. Přesnost se často měří podle hodnoty AP (*Average Precision*), respektive mAP (*mean Average Precision*). Přestože metody detekce objektů nejsou přímo hlavním zájmem této práce, tak zde nyní budou velmi stručně zmíněny významní zástupci těchto metod, které lze označit také jako detektory. Jak již bylo dříve naznačeno, právě metody pro detekci tvoří důležitý základ algoritmů pro sledování více objektů (*MOT*) a současně se některé jejich principy mohou objevit i při sledování jediného objektu [47].

První skupinou jsou detektory, jejichž algoritmus detekce se skládá ze dvou fází. V první fázi je generováno velké množství kandidátních regionů, které je následně pomocí RPN (*Region Proposal Network*) zúženo na regiony, jež mohou obsahovat potencionální objekty. Pro tyto regiony jsou následně extrahovány příznaky (*features*), podle nichž je ve druhé fázi provedena samotná klasifikace objektů. Zástupcem této skupiny jsou například metody, které využívají konvolučních neuronových sítí (*CNN – convolutional neural networks*). CNN jsou sítě, které pomocí konvoluce získávají příznaky ze vstupního obrazu a tyto příznaky jsou následně předány neuronové síti ve formě vektoru. Za revolučním vznikem této skupiny detektorů stojí algoritmus R-CNN [30], který ve své originální verzi dosahoval přesných výsledků, avšak výpočty byly velmi pomalé. Navazující práce [29, 66, 34] princip výpočtu postupně několikanásobně zrychlily, i tak se ale tyto metody neřadí mezi nejrychlejší a jejich hlavní předností je dobrá přesnost a schopnost detekovat i velmi malé objekty. Na stejných základech je postavena i další související metoda FPN (*Feature Pyramid Networks*) [48], která využívá víceúrovňové pyramidy pro extrakci příznaků.

[23] https://paperswithcode.com/task/object-detection
[24] https://cocodataset.org/

Druhou skupinu představují metody s jedinou fází detekce, ve které zapouzdřují predikci potencionálních regionů současně s klasifikací objektů. Tyto metody jsou obvykle rychlejší a dají se považovat za použitelné i v reálném čase, což je jejich hlavní výhodou. Patří sem především detektor YOLO (*You Look Only Once*) [63], k němuž doposud vznikla řada dalších verzí, které originální YOLO vylepšují především v přesnosti a také v možnosti aplikace pro vyšší rozlišení snímků [64, 65, 8].

YOLO se oproti předchozímu přístupu metod R-CNN liší především v tom, že se pro lokalizaci objektu na obrázku nevyužívá přístupu množství různých generovaných regionů (*RPN*). Využívá se zde jediné konvoluční neuronové sítě, která simultánně zpracuje oblasti snímku jakožto celku. Tento celek se vždy rozděluje na daný počet stejně velkých částí do mřížky a pro každou část je vytvořeno několik *bounding box* oblastí. Pro každý *bounding box* je vypočtena pravděpodobnost příslušnosti do konkrétní třídy objektů a pokud je pravděpodobnost dostatečně vysoká, je možné tento *bounding box* ještě dále upravit. Princip YOLO (viz obr. 3.9) je tedy výrazně jednodušší a přímočarý oproti detektorům zmíněným výše. Existují návody pro implementaci detektoru YOLO[25] a autoři tohoto přístupu detekce rovněž zveřejňují jejich postupný vývoj[26]. YOLO detektory jsou tedy velmi rychlé, nicméně jejich nevýhodou může být nepřesnost a to především pro skupiny malých objektů (např. hejna ptáků) [63].



Obrázek 3.9: Schéma systému YOLO [63]. Tento systém nejdříve změní velikost vstupního snímku na 448×448 a následně je snímek zpracován pomocí konvoluční neuronové sítě, na jejímž základě predikuje množství detekovaných rámečků představujících možné objekty. Finálními rámečky detekce jsou poté oblasti, které přesáhly zvolenou míru (*threshold*).

Jako kompromis mezi rychlostí a přesností lze považovat detektor RetinaNet [49], který byl navazující prací autorů FPN [48]. RetinaNet [49] má podobně jako YOLO [63] pouze jednu fázi, nicméně dosahuje pomalejší rychlosti, avšak výrazně lepší přesnosti než právě detektory typu YOLO. Podobně i detektor SSD [52] má také pouze jednu fázi a představuje balancovanou variantu rychlosti a přesnosti. SSD využívá přístupů tzv. *multiboxů*. To znamená, že si pro každý navržený *bounding box* přidává i několik blízkých modifikací, například změnou jeho poměru stran nebo jeho velikosti.

Na závěr této sekce i kapitoly je třeba dodat, že kromě zmíněných metod detekce i sledování objektů vznikla spousta dalších úspěšných metod, které není možné z hlediska rozsahu či z důvodu odbočení od hlavního tématu dále uvádět. V této kapitole byly stručně představeny principy i metody, které budou v následující části práce odkazovány v kontextu jejich využití pro panoramatické snímky a videa.

[25] https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/
[26] https://pjreddie.com/yolo/

Kapitola 4

Sledování objektů v 360° videu

Tato kapitola vychází z informací uvedených v předchozích kapitolách a zaměřuje se na dosavadní vývoj sledování a detekce objektů v panoramatických snímcích a videu. Tato část tedy představuje klíčový úsek práce, jelikož je nutné mít dobrý přehled o již existujících principech, aby bylo možné dosáhnout vlastního přínosu prostřednictvím této diplomové práce. Nejprve zde budou uvedeny problémy, které jsou specifické pro panoramatické video a které například v běžném videu není nutné v rámci sledování objektů řešit. Poté budou popsány i konkrétní principy a metody, které doposud vznikly pro řešení úlohy sledování i detekce objektů v panoramatickém videu.

4.1 Problematika sledování objektů v panoramatickém videu

V kapitole 2 byly představeny základy potřebné pro pochopení specifických vlastností panoramatického videa. V této sekci budou uvedeny problémy, které jsou se sledováním objektů v panoramatickém videu spojeny. Budou zde rozebrány příčiny těchto problémů a později v kapitole 5 budou navrženy i možné přístupy k jejich řešení. Pro úlohu sledování objektů v panoramatickém videu mohou nastávat podobné problémy jako pro sledování objektů v běžném videu. Může zde často docházet k problému částečného či úplného zakrytí (*occlusion*). Sférická kamera může být pevně umístěna a pohybovat se tak mohou pouze samotné objekty. Dále se sférická kamera může při pořizování videa také otáčet, v jednodušším případě se kamera otáčí pouze v horizontálně rovině. Pokročilejší situací je možnost, že se kamera může pohybovat či otáčet v různých trajektoriích, jako k tomu dochází při pořizování videa z rukou člověka či jiném specifickém pohybu. Je vhodné poznamenat, že se tyto scénáře související s pohybem a otáčením objevují i u běžných kamer, kde je úhel záběru mnohem menší. Pro úlohu sledování objektů v panoramatickém videu ovšem nastává hned několik specifických problémů.

Například zde již tak často nedochází k situaci, kdy sledovaný objekt zmizí z úhlu záběru 360° videa. Je to způsobeno širokým úhlem záběru, ze kterého prakticky objekt nemůže zmizet, pokud se ovšem dostatečně od kamery nevzdálí. Pokud již ke zmizení sledovaného objektu z úhlu záběru skutečně dojde, takový objekt se obvykle již nevrátí (například chodec procházející na ulici kolem sférické kamery). Kromě toho v případě zobrazení celého panoramatického snímku ($360° \times 180°$) dochází k výraznému zkreslení, které může zapříčinit selhání algoritmů sledování objektů. Vyskytuje se zde tedy řada skutečností způsobená širokoúhlým záběrem, nicméně některé z nich by mohl samotný algoritmus pro sledování objektů využít ve svůj prospěch.

Projekce

Klíčovou otázkou sledování objektů v panoramatickém videu je, v jakém zobrazení či projekci se budou snímky videa zpracovávat, respektive v jaké projekci bude tracker predikovat pozici sledovaného objektu. Zkreslení či zakřivení je jasně viditelné i lidským okem a v projekci celých 360° × 180° snímků do dvourozměrného prostoru se mu zcela vyhnout nedá. Zkreslení je zde nutné brát jako fakt, pro který je možné zkoumat jeho dopady na sledování objektů. Sledovaný objekt se navíc může vyskytovat během časového úseku ve videosekvenci na odlišných částech snímků, kde se zmíněné zakřivení může lišit. Příkladem budiž snímky v ekvirektangulární projekci na obrázku 4.1. V ekvirektangulární projekci je zkreslení nejvyšší především v horní a spodní části, která odpovídá pólům sféry mapované právě do tohoto zobrazení. Ovšem i v dalších částech ekvirektangulární projekce je patrné určité radiální zkreslení.

Obrázek 4.1: Člověk na snímcích[1] vypadá velmi rozdílně, což je způsobeno zkreslením, které se na různých místech snímku v ekvirektangulární projekci může lišit.

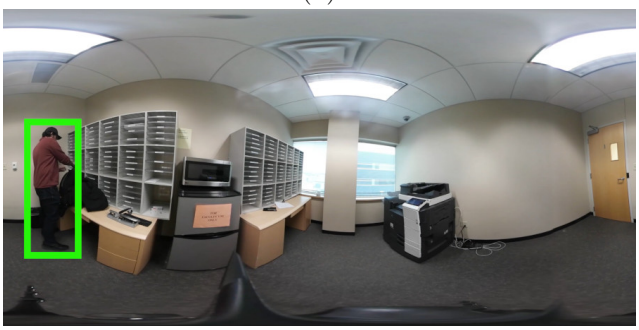
Další problém souvisí především se zmíněnou ekvirektangulární projekcí panoramatického snímku. Je zřejmé, že se objekty mohou objevovat současně na levém i pravém okraji současně. Souvisí to s faktem, že na sebe panoramatický snímek teoreticky navazuje v horizontální rovině. Příklad tohoto problému je ilustrován na obrázku 4.2, kde byl anotován ideální průběh sledování objektu. V ekvirektangulárním zobrazení může k takovým situacím docházet velmi často. Objekt se může sám pohybovat, jako je tomu například v případě chůze člověka. Kromě toho se může otáčet i samotná sférická kamera při pořizování videa a statický objekt se díky otáčení může rovněž přesunout přes okraj rámečku na okraj opačný. K tomuto problému v běžném videu reálně docházet nemůže a je zřejmé, že takové situace musí metoda pro sledování objektů v ekvirektangulární projekci 360° řešit. K tomuto problému ovšem nedochází například v projekci kubické a polární.

Právě problém přechodu objektu mezi okraji ekvirektangulárních snímků se stal jedním z hlavních cílů řešení této práce. Metody sledování objektu nejsou ve svém výchozím stavu schopny tento problém řešit, což ukázalo i vyhodnocení popsané v kapitole 6. S přechodem objektu mezi okraji snímku se totiž nedokáží úspěšně vypořádat ani současné moderní trackery, které v běžném videu dosahují jinak velmi přesných výsledků.

[1] https://github.com/uenian33/360_object_detection_dataset

(a)

(b)



(c)

Obrázek 4.2: Objekt (muž) se pohybuje po kanceláři a během pár okamžiků se jeho pozice v ekvirektangulární projekci panoramatického videa[2][60] rychle změní - (a) pravá část snímku, (b) přechod na okraji, (c) levá část snímku.

Výpočetní náročnost a velikost objektů

Dalším možným problémem metod sledování objektů je vysoké rozlišení panoramatických videí. Je zřejmé, že pro udržení dobré kvality vyžaduje panoramatické video výrazně vyšší rozlišení než mají běžná videa, která zabírají pouze omezený úhel záběru. Dnešní 360° videa jsou obvykle pořizována alespoň v rozlišení 4K. Z toho plyne, že výpočetní složitost pro zpracování obrazu v panoramatických videích bude obvykle vyšší, než by tomu bylo v případě běžných videí. Kromě toho jsou v širokém úhlu záběru videa objekty výrazně menší, než je tomu v běžném videu. Proto může některým trackerům činit problém právě malá velikost sledovaného objektu.

[2] https://github.com/acmmmsys/2019-360dataset

4.2 Metody sledování objektů v panoramatickém videu

Tato práce není zcela revoluční záležitostí, jelikož se problematice sledování objektů v panoramatickém videu věnovala řada různých výzkumů a dosud vzniklo již několik metod a odborných článků. Tyto odborné články obvykle představují vylepšení či modifikaci některého trackeru, který byl původně vytvořen a testován pro sledování objektů v běžném videu. V této sekci je proto uveden souhrn metod a článků, které se již sledováním v panoramatickém videu dříve zabývaly. Právě rešerše těchto existujících metod představuje důležitou část této práce a proto zde budou popsány jejich hlavní principy.

Přímé použití metody sledování objektu

Při úvodním zamyšlení nad řešením úlohy sledování objektu v panoramatickém videu se přímo nabízí naivní řešení. První možnost, která člověka může napadnout, je přímo využít existující trackery, které byly testovány pouze pro použití v běžném videu. Pro řešení této otázky vznikly také některé studie [43, 55], kde byly existující trackery testovány přímo pro sledování objektů v ekvirektangulární projekci.

Článek z roku 2019 [55] porovnal 8 implementací trackerů dostupných z rozšiřujících modulů knihovny OpenCV[3]. Konkrétně se jednalo o trackery BOOSTING [31], MIL [2], MedianFlow [41], TLD [42], MOSSE [9], KCF [37], GOTURN [36] a CSRT [54]. Vyhodnocení probíhalo na datasetu 9 ekvirektangulárních videosekvencí s označeným referenčním objektem. Nejlepších výsledků v této práci dosáhly nejnovější trackery GOTURN [36] a CSRT [54]. Byly zde diskutovány získané výsledky a možné příčiny úspěšnosti či neúspěšnosti jednotlivých trackerů pro všech 9 videosekvencí. Závěr této práce [40] je takový, že pro dosažení kvalitních výsledků v panoramatickém videu by bylo nutné trackery výrazně vylepšit. Autoři tohoto vyhodnocení rovněž zveřejnili zmíněný dataset, jenž byl v jejich práci použit pro vyhodnocení OpenCV trackerů[4].

Podobně i v rámci jiné práce [43], která vznikla v roce 2018, byla provedena evaluace metod sledování jediného objektu na panoramatických snímcích v ekvirektangulární projekci. Dle jejich autorů se jedná o vůbec první studii, která se zabývá vyhodnocením trackerů pro 360° videa v ekvirektangulární projekci [43]. Pro vyhodnocení bylo vybráno celkem 6 trackerů – TLD [42], GOTURN [36], STRUCK [33], C-COT [19] a MDNet [59]. V práci [43] rovněž nebyla prezentována žádná vylepšení zmíněných trackerů pro jejich použití v panoramatickém videu. Byla zde pouze vyhodnocena přesnost založená na metrice IoU a také robustnost všech trackerů na základě jejich selhání *failures*. Autoři [43] uvedli, že se ve své budoucí práci budou nadále zabývat tématem sledování objektů v panoramatickém videu a sami navrhnou jejich možná vylepšení. Doposud již však žádný navazující přístup v oblasti panoramatického videa nevytvořili.

Autoři také uvedli, že jejich evaluační dataset byl složen z videosekvencí pořízených pomocí sférické kamery Nokia OZO[5] a měl být také zveřejněn. Na rozdíl od předchozí práce [40], tento dataset v tuto chvíli není veřejně dostupný. Vzhledem k tomu, že se vyhodnocení v této diplomové práci částečně podobá právě obou zmíněným evaluacím [55, 43], byla zde snaha kontaktovat autory zmíněného článku [43] za účelem získání jejich podrobných poznatků, respektive vytvořeného datasetu. Bohužel byla tato snaha neúspěšná, a proto jinak jistě užitečné anotace z datasetu [43] nebyly v této diplomové práci nijak využity.

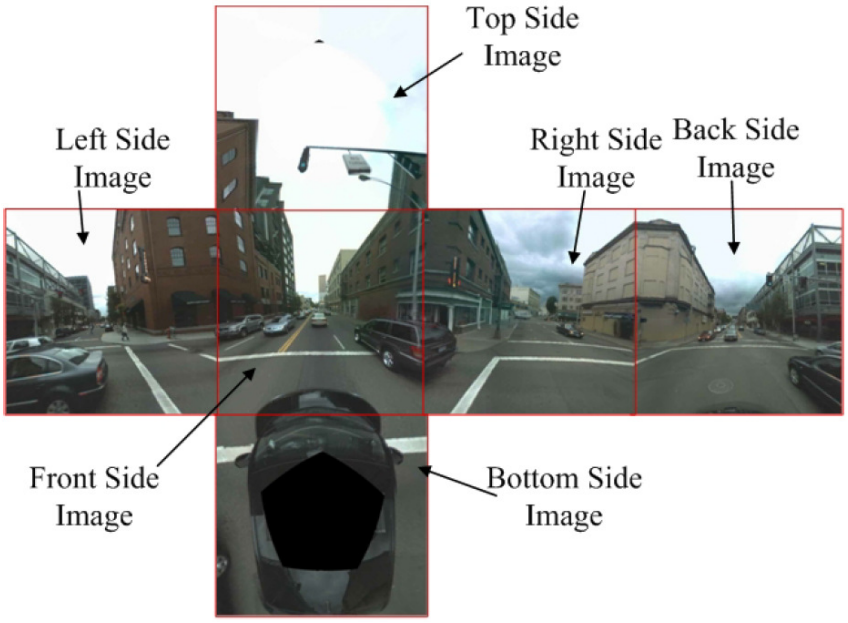
[3] https://github.com/opencv/opencv_contrib
[4] https://drive.google.com/drive/folders/1YbpOG6yWXYCsPO6nzEMRJR-exKODSos8
[5] https://en.wikipedia.org/wiki/Nokia_OZO

Adaptované metody

Následující rešerše představuje stručný souhrn metod, které byly dosud vytvořeny pro sledování objektů v panoramatickém videu. Velká část principů starších metod by již dnes pravděpodobně nenašla praktické uplatnění, ale je vhodné mít o takových metodách komplexní přehled. Označení adaptované metody vychází z předpokladu, že byl použit některý existující přístup sledování objektů adaptovaný pro jeho použití v 360° videu.

Metody sledování objektů v 360° se začaly objevovat již na počátku tohoto století [28, 56], což nepochybně souviselo s vývojem katadioptrických kamer, jež byly představeny v kapitole 2. Postupně vznikaly další metody, kupříkladu metoda z roku 2006 [44] byla zaměřena na sledování lidí v cylindrické projekci 360° videa, které pořizoval mobilní robot pomocí katadioptrické kamery. V následujícím roce vznikla metoda [12], která se orientovala na sledování silničních čar v panoramatických záběrech pořízených během jízdy vozidel. Jejím záměrem bylo adaptovat některé funkce asistence řízení, které by bylo možné realizovat pomocí jediné všesměrové kamery. Samotný algoritmus sledování zde byl založen extrakci příznaků v kombinaci s použitím Kalmanova filtru.

V roce 2010 byla uvedena metoda [97], která se zaměřila na sledování objektů v panoramatickém videu v kubické projekci pokrývající plných 360° × 180°. Hlavním cílem této metody je sledování objektů v reálném čase ve snímcích z pohybujícího se vozidla, přičemž se tracker musí vypořádat s přerušením a deformací způsobených kubickou projekcí. Článek [97] přesně popisuje úpravy a reprojekce, pomocí kterých lze zobrazovat konkrétní část kubické projekce tak, aby mohlo sledování probíhat i mezi hranami zobrazené krychle. Jedná se o tracker specificky zaměřený na sledování objektů z oblasti dopravy. Tato metoda je prakticky založena na *Mean-Shift* algoritmu [14], který umožňuje rozlišení sledovaného objektu oproti vedlejším objektům a pozadí. Metoda dále navrhuje řešení problémů, které jsou zapříčiněny záběry z jedoucího vozidla, například pokud je sledovaný objekt rozmazaný vlivem pohybu (*Motion Blur*).



Obrázek 4.3: Sledování objektů probíhá ve zmíněné metodě [97] v kubické projekci 360° videa.

Další metoda MTLD [24] již není zaměřena na konkrétní typ objektů. Název metody *Modified Tracking-Learning-Detection* [24] napovídá, že tato metoda využívá tracker TLD (*Tracking-Learning-Detection*) [42]. Článek [24] tak popisuje určitá vylepšení implementace trackeru *TLD* pro jeho využití v panoramatickém videu. Tracker *MTLD* provádí výpočty v rektifikované projekci polárních snímků, kterou je možné vidět na obrázku 4.4b. Proces rektifikace je velmi stručně popisován v samotném článku [24], přičemž se zřejmě jedná o specifickou verzi převodu mezi polární a kartézskou soustavou souřadnic.

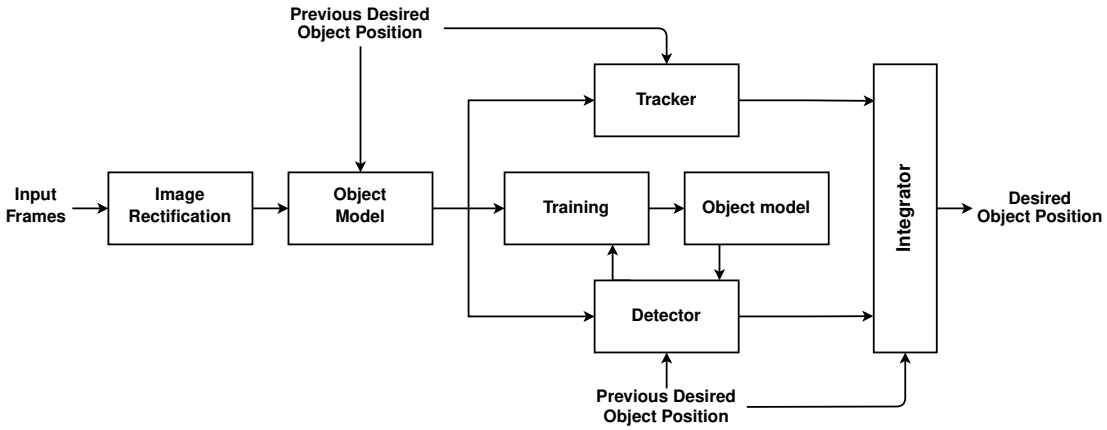
(a)

(b)

Obrázek 4.4: 360° snímek v (a) polární projekci a (b) rektifikované projekci podle MTLD [24]

Metoda *MTLD* [24], respektive TLD [42], využívá mechanismu online trénování, tedy průběžného trénování modelu během samotného procesu sledování. Jednotlivé moduly metody TLD byly již stručně popsány v přechozí kapitole v sekci 3.2. Právě v těchto modulech provádějí autoři MTLD některé modifikace – kupříkladu upravují a přidávají klasifikátory do modulu pro detekci. Oproti TLD [42] autoři rovněž omezují i prohledávanou oblast, jelikož objekt v použité rektifikované projekci panoramatického snímku nemůže na po sobě jdoucích snímcích výrazně či skokově změnit svoji pozici. Kompletní schéma metody [24] je znázorněno pomocí blokového diagramu na obrázku 4.5.

V článku [24] ovšem chybí informace o tom, jak se řeší například problém přechodu objektu z jednoho okraje na druhý okraj ve snímku v rektifikované projekci. Je pravděpodobné, že tento autoři problém zanedbali, nebo případně spoléhali na detekční modul, který by za určitých okolností mohl sledovaný objekt detekovat i v opačné části snímku. V rámci evaluace autoři [24] pak ukazují výrazné zlepšení úspěšnosti sledování objektu a také zrychlení implementace pro panoramatické snímky oproti původnímu algoritmu TLD [42]. Tracker MTLD se podle uvedených měření [24] nedá považovat za použitelný v reálném čase.



Obrázek 4.5: Blokový diagram metody *MTLD* [24]

Ve stejném roce autoři *MTLD* přicházejí také s řešením sledování objektů v polární projekci 360° videa [23] (obr. 4.4a). Jejich řešení se snaží vypořádat se zkreslením a rotací, která je v uvedené polární projekci panoramatického snímku jasně patrná. Při porovnání této polární projekce a ekvirektangulární projekce lze uvažovat, že horní část snímku v ekvirektangulární projekci odpovídala středu v projekci polární, a současně by spodní část snímku v ekvirektangulární projekci odpovídala části na obvodu snímku v polární projekci.

Autoři uvádějí, že rektifikace zmíněná výše [24], zvyšuje náročnost výpočtu. Právě z důvodu rychlosti a náročnosti trackeru se autoři [23] rozhodli v rámci své práce navázat a pokračovat v problematice sledování objektu v jiné projekci panoramatického videa. Samotný algoritmus sledování je založen na trackeru KLT [10], přičemž se zde využívá návrhu kandidátních regionů, jež mají podobu lichoběžníku a nikoliv tradičního obdélníku.

Autoři předchozích článků [23, 24] na svoji práci dále navázali formou metody [21]. V této metodě [21] se opět uvažuje proces sledování objektů v polární projekci (obr. 4.4) a je zde opět zdůrazněna nevýhoda použití procesu rektifikace, čímž autoři označují převod snímku v polární projekci do konkrétní rektifikované podoby. Časová náročnost pro předzpracování snímku do rektangulární podoby je podle autorů nezanedbatelná především kvůli zvyšování velikosti rozlišení panoramatického videa. Autoři zde rovněž uvádějí, že jejich algoritmus sledování objektů v polární projekci [23] měl sice lepší časovou náročnost než první metoda *MTLD* [24], ale přesnost výsledků byla bez provedené rektifikace horší. Rozdílem oproti předchozí práci předchozí přístup [23] je mimo jiné přístup extrakce příznaků, jelikož je v navazujícím přístupu [21] využita metoda *SURF (Speeded Up Robust Features)*. Tito autoři zveřejnili i detailnější souhrn jejich výzkumu formou článku [22]. Během řešení této diplomové práce byli autoři [24, 23, 21, 22] kontaktováni s cílem získat některé podrobnosti z jejich práce a případně i dataset, který využili pro evaluaci jejich metod. Bohužel se nepodařilo získat zpětnou vazbu, která by mohla této práci určitým způsobem pomoci (například právě datasetem obsahujícím videa v polární projekci).

V posledních letech vznikly i další metody [71, 11], jejichž proces sledování objektu probíhá v polární projekci 360° pořízeného pomocí katadioptrického systému [75]. Taková videa nelze považovat za ryze všesměrové, jelikož nepokrývají celých 180° ve vertikální rovině. Je tedy zřejmé, že vývoj sledování objektů v polární projekci probíhá už poměrně dlouhou dobu a vznikla řada metod, jež byly v této sekci velmi prostě popsány. Naproti tomu vývoj principů, které by umožnily efektivní sledování objektů v ekvirektangulární

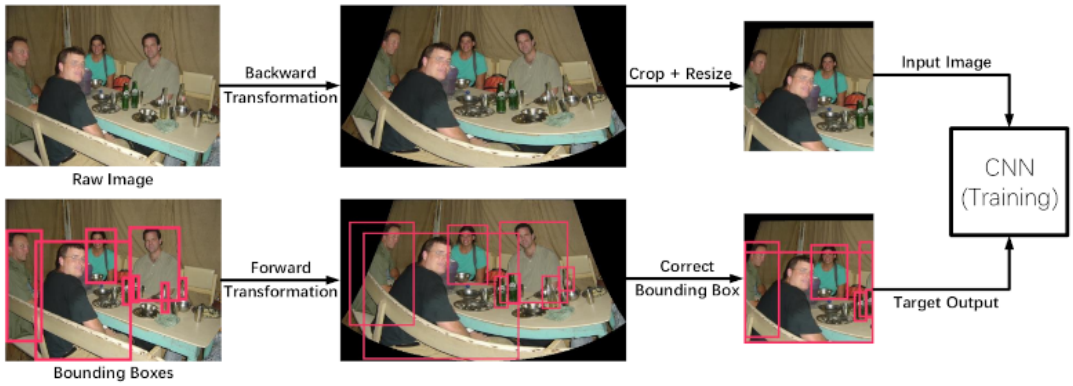
projekci je v tuto chvíli spíše na počátku. A přitom právě trend nových sférických kamer určuje, že v současnosti nejvíce používanou projekcí je právě ekvirektangulární zobrazení, nebo popřípadě kubické zobrazení. Právě video pořízené sférickou kamerou bývá výchozím způsobem mapováno nejčastěji do ekvirektangulární projekce. Pro vylepšení sledování jediného objektu pro panoramatické video v ekvirektangulární projekci toho doposud bylo provedeno velmi málo [43], a právě tato skutečnost se stala hlavní motivací této diplomové práce.

Na závěr této sekce bude krátce zmíněna i oblast sledování více objektů současně (*MOT*). Na rozdíl od sledování jediného objektu (*SOT*) již vznikly metody adaptované právě pro sledování více objektů současně v ekvirektangulární projekci [51, 26]. Metoda z roku 2018 [51] je založena na trackeru SORT (*Simple online and realtime tracking*) [4], respektive DeepSORT [87] a proces sledování zde probíhá právě v ekvirektangulární projekci. Autoři zde adresují problém přechodu objektu z jednoho okraje ekvirektangulárního snímku na opačný okraj, který je řešen pomocí rozšíření okrajů (*boundary connection*). Díky tomu je pak možné detekovat i objekt nacházející se přímo na okraji. O rok později vznikla další metoda OmniTrack [26], která umožňuje rovněž sledování více objektů současně a je postavena na detektoru YOLOv3 [65].

4.3 Detekce objektů v panoramatických snímcích

Lze tvrdit, že detekce objektů v 360° snímcích má již za sebou větší pokrok než sledování objektů, a to především v diskutované ekvirektangulární projekci. I pro panoramatické snímky má detekce objektů přímou souvislost s úlohou sledování objektů, a to především s výše zmíněným sledováním více objektů současně (*MOT*). Detekce objektů v 360° panoramatických snímcích může nacházet uplatnění například v oblasti asistence řízení vozidel nebo při navigaci dronu [94]. Bude zde uvedeno i několik zástupců metod pro detekci objektů v panoramatických snímcích, které vznikly v nedávné době.

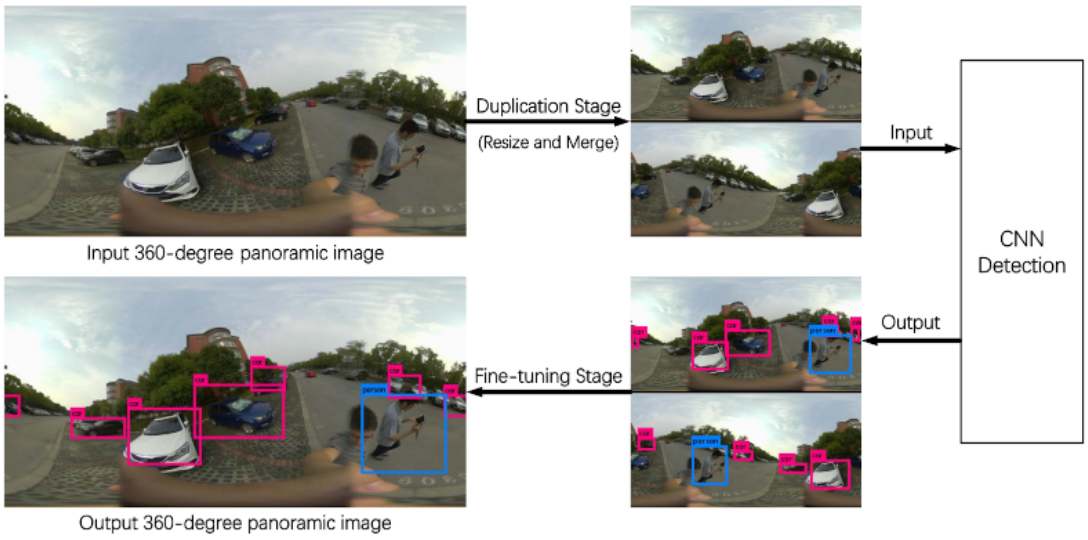
První metodu, která bude v této části stručně popsána, popisuje článek z roku 2017 [94]. Tato metoda představuje systém založený na konvoluční neuronové síti se zaměřením na aplikaci detekce objektů v reálném čase. Autoři zde představují myšlenku transformace, která umožní převod existujících datasetů s označenými objekty na běžných snímcích na dataset snímků se zkreslením odpovídajícím výřezu panoramatického snímku v ekvirektangulární projekci (viz obr. 4.6). Díky tomu je tak možné v podstatě simulovat mírné radiální zkreslení, které se v ekvirektangulární projekci objevuje. Autoři [94] uvádějí, že by pro trénování sítě bylo ideální využít dataset přímo s anotovanými 360° snímky, avšak takový dataset v době vzniku této metody neexistoval. Samotné trénování modelu na transformovaných snímcích probíhá pomocí hlubokého učení, konkrétně je založeno na CNN použité v detektoru YOLO [63].



Obrázek 4.6: Schéma transformace [94] běžných snímků na výřez z panoramatické projekce, díky čemuž je možné existující datasety detekce objektů transformovat na datasety použitelné pro trénování modelů za účelem detekce objektů v panoramatických snímcích.

Kromě transformace je v této metodě [94] řešen i problém objektu nacházejícího se na horizontálních okrajích v ekvirektangulární projekci. S tímto problémem se tato metoda vypořádává pomocí tzv. duplikační etapy (*duplication stage*), která kromě vstupního snímku vytvoří i druhý snímek, který odpovídá posunu vstupního snímku o 180° v horizontální rovině. Pro oba snímky je následně provedena CNN detekce a výsledky jsou poté v rámci finální etapy (*fine-tune stage*) sloučeny v jeden konečný výsledek. Jednoduché schéma detekce této metody je znázorněno na obrázku 4.7.

Jak již bylo zmíněno, detekce je založena na detektoru YOLO [63] za účelem využití v reálném čase. Tato metoda [94] původní detektor YOLO překonává v přesnosti pro panoramatické ekvirektangulární snímky, nicméně trpí podobnými nedostatky jako YOLO. Ve vyhodnocení bylo ukázáno, že tento detektor velmi špatně detekuje skupiny malých objektů [63], kterých se v ekvirektangulární projekci 360° vyskytuje mnohem více než v omezeném úhlu záběru běžných snímků. Autoři [94] ovšem dodávají, že by bylo možné využít i jinou metodu detekce objektů [30, 29, 66].



Obrázek 4.7: Etapy detekce objektů v metodě [94]

Další práce [91] je založena na detektoru YOLOv2 [64] a identifikuje tři hlavní problémy detekce v panoramatických snímcích – nedostatek panoramatických snímků s označenými objekty, vysoké rozlišení panoramatických snímků a zkreslení způsobené panoramatickou projekcí. Tato práce [91] byla mimo jiné vytvořena skupinou autorů, která již dříve představila zmíněnou evaluaci pro trackery v ekvirektangulární projekci [43]. Vytvořili vlastní dataset[6] panoramatických snímků v ekvirektangulární projekci určený pro evaluaci detekce objektů. Tento dataset obsahuje 903 snímků s vysokým rozlišením, kde je označeno více než 7000 objektů. Autoři implementovali vlastní anotační nástroj, pomocí kterého byly následně anotovány objekty na snímcích pro účely vyhodnocení jejich přístupu.

Autoři ve své práci [91] představili modifikaci detektoru YOLOv2 [64] – tzv. mp-YOLO (*Multi-projection YOLO*), jehož myšlenkou je rozdělení ekvirektangulární projekce panoramatického snímku na několik stereografických reprojekcí. Počet reprojekcí je dán na základě úhlu záběru dané reprojekce (*Field Of View*). Platí, že čím menší je úhel záběru, tím více je nutné provést stereografických reprojekcí a tedy i následného zpracování každé z těchto reprojekcí separátně pomocí detektoru YOLOv2 [64]. Zde jsou využity celkem 4 reprojekce s úhlem záběru o velikosti 180° horizontálně i vertikálně. Tyto 4 reprojekce se v horizontální rovině současně překrývají v 90° z důvodu zkreslení na okrajích těchto 180° stereografických reprojekcí. Po dokončení procesu detekce v každé z reprojekcí je následně zpětně vytvořena jejich ekvirektangulární projekce. Nakonec je nutné provést i výsledné zarovnání rámečků *bounding boxes* detekovaných objektů, aby odpovídali obdélníkům v ekvirektangulární projekci, nikoliv v původních reprojekcích a bylo tak možné porovnávat výsledky. V práci je následně provedena evaluace výsledků a porovnání mezi detektory FasterR–CNN [66], YOLOv2 [64] a právě představenou metodou mp–YOLO [91], přičemž jsou zde použity původní natrénované modely podle klasických snímků (dataset tedy není nijak adaptován).

Kromě samotných metod detekce byly již také detailně řešeny některé přístupy pro extrakci příznaků (*features*) pomocí konvolučních neuronových sítí [70, 15]. V roce 2020 byl mimo jiné vytvořen i rozsáhlý dataset *360-Indoor* [13] za účelem strojového učení modelů (respektive hlubokého učení) pro detekci a klasifikaci v panoramatických snímcích. Tento dataset je možné stáhnout po vyplnění dotazníku účelu použití[7]. V datasetu *360-Indoor* se objevuje celkem 37 tříd objektů, které jsou převážně tvořeny lidmi a také předměty z interiéru budov a místností. Nachází se zde více než 3000 snímků a přibližně 90000 *ground-truth* anotací. Je možné jej tedy využít pro evaluaci, ale především i hluboké učení. *360-Indoor* by tak mohl dát základ pro vznik dalších datasetů, jež by v budoucnu mohly výrazně zlepšit výsledky hlubokého učení na panoramatických snímcích. Ukázku anotovaného snímku z datasetu si lze prohlédnout na obrázku 4.8. Je vhodné poznamenat, že souřadnice označených objektů (*groundtruth data*) jsou definovány ve sférické soustavě souřadnic a jejich rámečky jsou zde nazvány jako BFoVs (*Bounding Field-of-Views*) [13].

[6] https://github.com/uenian33/360_object_detection_dataset
[7] http://aliensunmin.github.io/project/360-dataset/

Obrázek 4.8: Příklad anotovaného snímku v ekvirektagulární projekci, který je součástí datasetu *360-Indoor* [13] určeného pro strojové učení i vyhodnocení detekce a klasifikace objektů v panoramatických snímcích.

Kapitola 5

Návrh a implementace

V této kapitole bude představeno a navrženo vlastní řešení pro vylepšení metod sledování objektů v panoramatickém videu. Budou zde přesně popsány provedené modifikace, které byly inspirovány studiem vlastností panoramatických videí a také rešerší dosud existujících přístupů k této problematice. Dále budou uvedeny veškeré prostředky a existující implementace, které byly využity v programové části této diplomové práce.

5.1 Návrh řešení

Návrh vlastního řešení vychází ze studia metod sledování objektů a především také ze samotných vlastností 360° videa. V kapitole 4 byly uvedeny různé přístupy, které byly využity v dosavadních metodách zabývajících se sledováním objektů v panoramatickém videu. Právě rešerše a studium existujících praktik je velmi důležitá, a to z důvodu, aby se vlastní práce nezabývala realizací již existujících principů.

V této práci nebyla reimplementována žádná z metod specializovaných na sledování objektů v panoramatických videích, a to z důvodu neopodstatněného přínosu takových reimplementací nebo nedostatečných informací v některých článcích zmíněných v předchozí kapitole 4. Právě díky podrobnému zmapování dosavadního vývoje se směr této práce postupně orientoval na skutečnosti, které dosud nebyly detailně otestovány a nebyl zkoumán jejich vliv na úspěšnost sledování objektů. Motivací této práce se tak stala možnost vylepšení metod sledování jediného objektu (SOT) v ekvirektangulární projekci panoramatického videa, přičemž proces sledování zde probíhal přímo v kartézské soustavě souřadnic [81]. Možnosti vylepšení trackerů lze v zásadě rozdělit do tří zastřešujících kategorií.

Úprava implementace algoritmu

Smyslem této práce není přesně reimplementovat existující trackery podle originálních odborných článků. Taková implementace bývá často velmi složitá, jelikož v článku nemusí být všechny potřebné údaje a pro přesnou interpretaci údajů může být žádoucí navázat dobrou komunikaci s jejich autory. Předmětem zájmu se proto staly trackery, pro které je možné nalézt dostatek údajů a také jejich existující implementace. Takové implementace mohou být v ideálním případě volně dostupné pro akademické účely, případně i pro komerční využití. Jenom velmi těžko by bylo možné v rámci této diplomové práce provést vlastní reimplementaci některého trackeru, která by dosahovala stejných nebo lepších výsledků než oficiální implementace, na které skupiny autorů pracovaly měsíce až roky.

Samotné úpravy implementace trackerů pro použití v panoramatickém videu závisejí především na projekci panoramatických snímků, ve které sledování objektů probíhá. Například článek [24] přinesl cílenou úpravu konkrétních částí trackeru TLD [42], aby jej bylo možné využít v rektifikované podobě polárních snímků panoramatického videa. Řada dalších přístupů představila úpravy na nejnižší úrovni implementací trackerů, například specifickou extrakci příznaků [11] či úpravu tvaru rámečku (*bounding boxu*) [23, 21]. Díky tomu pak bylo možné sledování objektů provádět přímo v polární projekci 360° videa. Prakticky se tak výsledek celého procesu sledování objektů odvíjí od toho, zda probíhá ve sférických souřadnicích [85], polárních souřadnicích [84] či přímo v kartézské soustavě souřadnic [81].

Samotnou úpravou implementace je zde tedy myšlena modifikace určité části trackeru a to na jeho nejnižší úrovni (*jádro metody*). Úpravy tohoto typu nejsou triviální a je nutné k nim mít řádné opodstatnění. Pro takové modifikace by tedy musel být zvolen konkrétní tracker. Dále by musely být provedeny a detailně popsány veškeré úpravy jeho původní implementace. Nakonec by muselo být ukázáno, že tyto modifikace skutečně mají přínos pro vylepšení přesnosti trackeru. I kdyby však konkrétní modifikace vylepšila vybraný tracker, nebylo by zaručeno, že by pak identická úprava mohla zároveň vylepšit úplně jiný tracker. Jednalo by se tedy spíše o specifické a nikoliv univerzální modifikace. Například zmíněná metoda MTLD [24] se zaměřila na detailní úpravy trackeru TLD [42], ale je téměř jisté, že by podobných úprav nebylo možné dosáhnout například pro tracker KCF [37]. Jelikož zadání této diplomové práce nestanovilo žádnou konkrétní metodu, úprava interní implementace některého trackeru se tak stala spíše méně zajímavou možností pro dosažení určitého přínosu v řešené problematice.

Adaptace modelu pro panoramatické snímky

Moderní trackery často využívají hlubokého učení pro vytvoření modelu, jenž bývá trénován předem (*offline*) a trackeru je následně dostupný. Výsledek procesu sledování tak může být velmi ovlivněn i zmíněným modelem, respektive sítí. Uvažujeme, že sledování objektů v 360° videu probíhá v některé projekci z výčtu uvedeného v kapitole 2. Jak již bylo zmíněno, tyto projekce, pokrývající celých 360° × 180°, obsahují různé zkreslení či zakřivení. Výchozí modely, které jsou obvykle představeny v samotném článku o vybraném trackeru [18], jsou však trénovány na základě anotací v běžných snímcích, jakožto součásti některého rozsáhlého datasetu [25].

Pro trénink zmíněného modelu trackeru by mohlo být výhodné mít k dispozici dataset anotovaných panoramatických snímků. Problém ovšem nastává v samotné existenci a dostupnosti dostatečně širokého datasetu. V nedávné době vznikl v podstatě první rozsáhlý dataset se zaměřením na učení modelu pro detekci a klasifikaci objektů [13] v ekvirektangulární projekci. Je pravděpodobné, že se v blízké době mohou objevit i další podobné datasety, které budou znamenat větší pokrok pro učení modelů za účelem jejich využití pro zpracování obrazu 360° snímků. Bylo by také možné využít i přístupu uvedeného v metodě detekce [94], kde jsou běžné snímky z existujících datasetů [50] transformovány, respektive radiálně zkresleny. Teoreticky lze vytvořit i vlastní rozsáhlý dataset s manuálními anotacemi. Vytvoření takového datasetu by bylo zřejmě obrovským přínosem pro budoucí vývoj detekce objektů v 360° snímcích, ale výrazně by přesáhlo rozsah této diplomové práce.

Tato podsekce zde byla uvedena pouze jako další kategorie možných vylepšení, která by sice neřešila některé problémy související s konkrétní projekcí, ale mohla by přispět ke zlepšení přesnosti moderních trackerů. Navržené vylepšení, v podobě trénování speciálního modelu pro účely sledování objektů v 360° videu, ale nebylo v této práci nijak realizováno.

Adaptace panoramatických snímků

Již mnohokrát zde zaznělo, že proces sledování objektů závisí na konkrétní podobě panoramatických snímků, respektive na jejich projekci. Bylo také uvedeno, že se tato práce bude orientovat na sledování objektů, kde budou vstupní panoramatické videosekvence v ekvirektangulární projekci. Myšlenka úpravy výchozí ekvirektangulární projekce na několik reprojekcí se objevila například v článku [91] zabývajícím se detekcí objektů. Jiná metoda [51] pro sledování více objektů současně představila princip, kdy je každý snímek rozdělen na dvě projekce. První projekce je identická jako vstupní ekvirektangulární snímek a druhá projekce představuje rotaci originálního snímku o 180° v horizontální rovině. V každém snímku jsou detekovány objekty v obou projekcích a výsledek je následně spojen. Díky tomuto přístupu je možné detekovat i objekty, které se nacházejí na okrajích ekvirektangulární projekci, jelikož v jedné ze dvou projekcí se s velkou pravděpodobností na okraji nacházet nebudou.

Právě přístup úpravy projekce, ve které bude proces sledování objektů probíhat, se stal hlavní náplní této diplomové práce. V následující sekci budou představeny dvě možnosti, kterými lze řešit například zmíněný problém přechodu objektu mezi okraji ekvirektangulární projekce. Nevýhodou těchto přístupů je jisté zpomalení trackeru, jelikož i proces reprojekce ovlivňuje dobu zpracování každého snímku. Naopak velkou výhodou adaptace panoramatických snímků je univerzálnost a možnost jejich využití prakticky pro libovolný tracker. Tato adaptace probíhá nad implementací samotného trackeru, na který je nahlíženo jako na černou skříňku (*black box*). Tracker jednoduše predikuje výsledky na adaptovaném snímku, přičemž získané výsledky lze zpětně převádět do originálního snímku. Tento princip jednoduše demonstruje pseudokód algoritmu 1.

```
...
video ← videoLoad();
frame ← read(video);
boundingBox ← selectROI();
frameAdapted ← reprojection(frame, boundingBox);
boundingBoxAdapted ← reprojectionBBox(boundingBox);
tracker ← trackerInit(boundingBoxAdapted, frameAdapted);
while frame ← read(video) do
    frameAdapted ← reprojection(frame, boundingBox);
    boundingBoxAdapted ← trackerUpdate(tracker, frameAdapted);
    boundingBox ← reprojectionBBoxBackward(boundingBoxAdapted);
    ...
end
...
```

Algoritmus 1: Pseudokód přístupu adaptace vstupních snímků.

Všechny tři zmíněné kategorie vylepšení by mohly přinést cílenou přesnost trackerů pro jejich použití v 360° videu. Bylo by jistě možné upravovat projekci vstupních snímků a současně natrénovat specifický model pro využití ve zvolené projekci. V této práci jsou však realizovány pouze možnosti navržené v poslední kategorii vylepšení. První vylepšení je založeno na simulaci sférické rotace podle sledovaného objektu. Druhé vylepšení se pak omezuje pouze na část ekvirektangulární projekce a proces sledování probíhá v rektilineární či perspektivní projekci panoramatických snímků. Tato vylepšení budou podrobněji představena v následující sekci.

5.2 Implementace a vylepšení metod

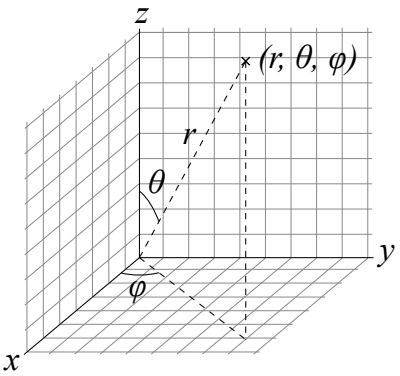
Popis v této sekci není vztahován na jedinou konkrétní metodu sledování objektů. Popis se se týká libovolného trackeru, pro který jsou navrženy úpravy vstupních snímků a proces sledování tak probíhá právě v upravených snímcích. První vylepšení je zaměřeno na řešení přechodu objektu mezi okraji snímku. Druhé vylepšení se orientuje na odstranění zkreslení z původních panoramatických snímků. Pro obě vylepšení jsou na vstupu uvažovány originální $360° \times 180°$ videa v ekvirektangulární projekci, která je v současnosti pravděpodobně nejvyužívanějším způsobem mapování 360° snímků.

Řešení přechodu objektu na okrajích snímku

Ekvirektangulární projekce, do které jsou mapovány vstupní panoramatické snímky, představuje zobrazení sféry do dvourozměrného prostoru. Přesněji se jedná o mapování ze sférické soustavy souřadnic [85] do kartézského souřadného systému [81]. V kartézské soustavě souřadnic se každý bod nachází ve dvourozměrném prostoru a jeho pozice je obvykle označována jako dvojice (x, y), přičemž x udává horizontální souřadnici a y vertikální souřadnici pozice bodu. Ve zpracování obrazu je kartézská soustava souřadnic využívána obvykle pro identifikaci konkrétního pixelu v načteném snímku a na tento pixel se nahlíží jako na bod v kartézském souřadném systému.

Každý bod ve sférické soustavě souřadnic se ale nachází v trojrozměrném prostoru a jeho pozice je charakterizována hned třemi hodnotami [85]. Souřadnice bodu ve sférické soustavě se nejčastěji definuje podle trojice (r, θ, φ), kde r udává vzdálenost bodu od počátku souřadnic, θ udává úhel odklonu průvodiče bodu od osy x a φ představuje úhel mezi průvodičem a osou z [85]. Bod ve sférické soustavě souřadnic je zobrazen na obrázku 5.1.

Pro ekvirektangulární projekci a video v ní mapované platí, že každý snímek je v podstatě rozprostřen na celém plášti sféry. Hodnota vzdálenosti od počátku souřadnic r je tudíž pro účely sférického videa zanedbatelná, jelikož je tato vzdálenost pro každý zobrazený bod stejná. Klíčové jsou tak pouze oba úhly θ a φ. Tyto úhly si lze velmi dobře představit v souvislosti s geografií a to konkrétně na příkladu zeměkoule. Hodnoty těchto úhlů se pro zeměkouli udávají pomocí dobře známých pojmů zeměpisné šířky (*latitude*) a zeměpisné délky (*longitude*). Pomocí dvou souřadnic lze tedy přesně označit každé místo na naší planetě. Úhel θ tedy může nabývat hodnot $(0°, +180°)$, respektive $(-90°, +90°)$ a úhel φ hodnot v rozmezí $(0°, +360°)$, respektive $(-180°, +180°)$.



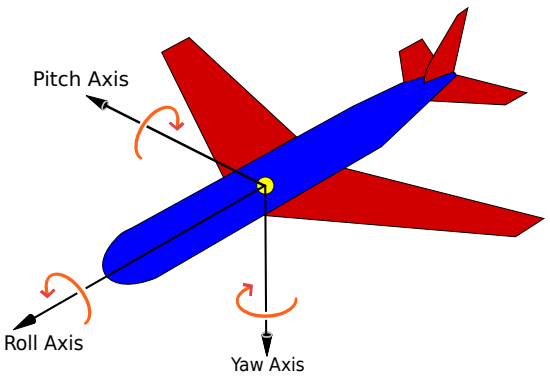
Obrázek 5.1: Bod ve sférické soustavě souřadnic je charakterizovaný trojicí hodnot[1] [85].

[1] https://en.wikipedia.org/wiki/File:3D_Spherical.svg

Originální 360° video může být pořízeno pomocí statické sférické kamery, která se nijak nepohybuje. Může se ovšem pohybovat libovolný objekt, který je předmětem zájmu úlohy sledování objektů. Nyní si představme 360° video zobrazené na kouli, respektive na sféře. Při pohybu sledovaného objektu může docházet pouze ke změnám souřadnic, které definují pozici tohoto objektu ve sférické soustavě souřadnic. V takovém se případě se budou pouze nepatrně měnit hodnoty úhlů souřadnic, na kterých se sledovaný objekt nachází. Pokud však budeme uvažovat stejnou situaci v ekvirektangulární projekci, kde je každý bod definován v kartézské soustavě souřadnic, může dojít k problému přechodu objektu na okrajích, který byl adresován již v kapitole 4. Bod odpovídající kupříkladu středu sledovaného objektu může vzhledem k pohybu objektu skokově změnit hodnotu své horizontální souřadnice. Sledovaný objekt se může na videu pohybovat směrem doprava, přičemž například na 100. snímku videa se střed objektu nachází na pozici $(frame\_width - 1, y)$. Na následujícím 101. snímku videa se již střed objektu může nacházet na pozici $(1, y)$. Celý objekt se tak může posunout z pravé části snímku do části levé během několika po sobě jdoucích snímků.

Již při prvních experimentech s přímým využitím trackerů v ekvirektangulární projekci (v kartézském souřadném systému [81]), byl výše uvedený fakt zhledán problematickým, jelikož trackery v takové chvíli selhali. V okamžiku přechodu objektu mezi okraji ekvirektangulární projekce trackery obvykle začaly predikovat falešně negativní (*false negative*) nebo falešně pozitivní (*false positive*) výsledky. Z tohoto důvodu se tato práce zaměřila na řešení problému přechodu objektu na základě sférické rotace.

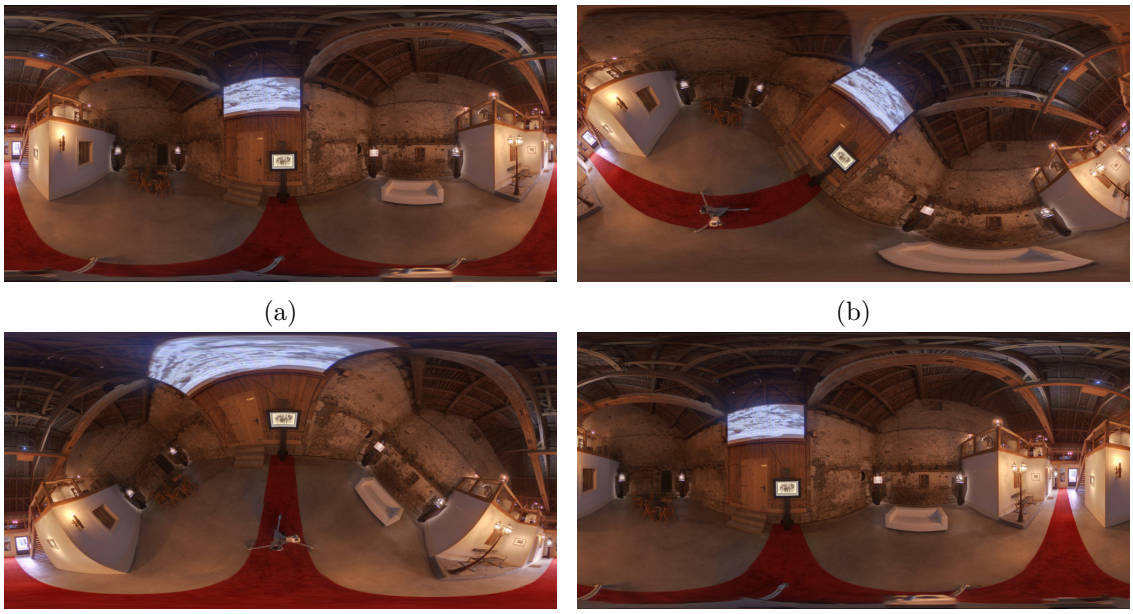
Bylo by možné zde popsat veškeré rovnice související se sférickou rotací. Tyto rovnice ale v této diplomové práci implementovány prakticky nebyly, a proto zde bude navržený princip vizualizován formou několika obrázků. Pro sférickou rotaci se využívají tzv. Eulerovy úhly [82], díky kterým lze sféru rotovat kolem jedné ze tří os x, y, z, případně i kolem více os současně. Pro rotaci kolem osy x se lze setkat s označením *roll*, pro rotaci kolem osy y s názvem *pitch* a pro rotaci kolem osy y s názvem *yaw*. Tyto názvy vycházejí z leteckých principů [78], kde definují konkrétní náklon a otočení letadla. Tato práce se pochopitelně zabývá rotací sféry[2] a nikoliv trojrozměrného předmětu jako je letadlo. Pro správné pochopení zmíněných rotací se však velmi často uvádí obrázek letadla 5.2.



Obrázek 5.2: Sférická rotace je založena na Eulerových úhlech [82]. Pro operaci sférické rotace kolem konkrétních os souřadného systému se lze často setkat s označeními *yaw, pitch, roll* [82]. Toto označení se používá v souvislosti s náklonem či rotací letadla[3][78].

[2] https://en.wikipedia.org/wiki/File:Euler2a.gif
[3] https://en.wikipedia.org/wiki/File:Yaw_Axis_Corrected.svg

Pro implementaci ekvirektangulární rotace by bylo nutné nejprve vypočítat pozici všech bodů sféry pomocí Eulerových úhlů [82], respektive pomocí rotačních matic[4]. Následně by bylo možné provést i jejich převod do kartézského souřadného systému [81] pro výslednou ekvirektangulární projekci. Některé implementace těchto rotací již existují[5], přičemž je zřejmé, že potřebné operace mohou probíhat i v reálném čase[6]. Ukázku rotace kolem jednotlivých os sférického souřadného systému [85] si lze prohlédnout na obrázku 5.3.



(a) (b) (c) (d)

Obrázek 5.3: (a) Originální 360° snímek v ekvirektangulární projekci[7], (b) rotace kolem osy x (*roll*) o 45°, (c) rotace kolem osy y (*pitch*) o 45°, (d) rotace kolem osy z (*yaw*) o 45°.

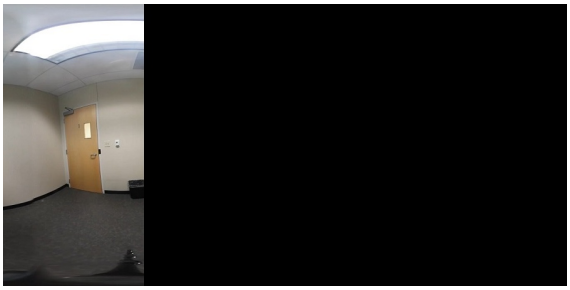
V programové části této práce však sférické rotace nebyly formálně implementovány. Namísto toho se podařilo realizovat simulaci rotace kolem osy z, která je klíčová pro problém přechodu sledovaného objektu mezi okraji v ekvirektangulární projekci. Tato rotace ovšem nebyla implementována podle zmíněného přístupu s využitím rotačních matic a Eulerových úhlů [82]. Namísto toho bylo využity vlastnosti ekvirektangulární projekce, a to sice faktu, že na sebe navazuje levý a pravý okraj ekvirektangulárního snímku.

Díky tomu lze jednoduše provést afinní operaci posunu v kartézské soustavě souřadnic, pro kterou byla využita funkce *cv::warpAffine*[8] z knihovny *OpenCV*. Takový posun se však v kartézské soustavě souřadnic neprovádí podle stupňů (úhlů), ale přímo podle počtu pixelů. Nicméně je zřejmé, že pokud celá šířka ekvirektangulárního snímku představuje celých 360°, lze snadno odvodit, že například posun o úhel 180° by odpovídal posunu o polovinu hodnoty šířky snímku.

[4] https://en.wikipedia.org/wiki/Rotation_matrix#Basic_rotations
[5] https://github.com/whdlgp/Equirectangular_rotate
[6] https://www.youtube.com/watch?v=l1N0lEKIeLA
[7] https://github.com/FoxelSA/libgnomonic/wiki/Equirectangular-rotation_v0.1
[8] https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html

Uvažujme příklad (viz obr 5.4), že má být ekvirektangulární snímek o šířce $1000px$ rotován o 90° doprava (což ve sférické soustavě souřadnic odpovídá rotaci kolem osy z). Úhel 90° stupňů by tak pro šířku ekvirektangulárního snímku o hodnotě $1000px$ odpovídal posunu o $1000px * (90°/360°) = 250px$. Ekvirektangulární snímek je tak posunut nejprve o $250px$ doprava, což odpovídá obrázku 5.4c. Následně je ale nutné provést ještě jeden posun a to naopak směrem doleva. Původní ekvirektangulární snímek je proto posunut o $1000px * ((360° - 90°)/360°) = 750px$ směrem doleva, jak je znázorněno na obrázku 5.4b. Na obrázcích 5.4b a 5.4c byly pro demonstraci doplněny černé části, které představují oblast, o kterou byl originální snímek posunut. Pokud jsou ale tyto černé části zanedbány, je možné obrázky 5.4b a 5.4c spojit, čímž je vytvořen snímek odpovídající cílené rotaci o 90°. Toto spojení bylo jednoduše realizováno pomocí funkce *numpy.concatenate*[9].

(a)



(b)

(c)

(d)

Obrázek 5.4: (a) Původní 360° snímek v ekvirektangulární projekci, (b) původní snímek posunutý o 270° doleva, (c) původní snímek posunutý o 90° doprava, (d) výsledný 360° v ekvirektangulární projekci, který byl pomocí simulace 90° sférické rotace v ose z transformován z původního ekvirektangulárního snímku.

[9] https://numpy.org/doc/stable/reference/generated/numpy.concatenate.html

Díky tomuto přístupu je tedy možné provádět simulaci sférické rotace ve chvíli, kdy se sledovaný objekt přibližuje k okraji ekvirektangulárního snímku. Jedná se tak prakticky o virtuální sférickou kameru, jejíž směr se odvíjí od předcházející pozice sledovaného objektu, která je definována rámečkem (*bounding box*). Pokud se horizontální souřadnice levého rohu predikovaného rámečku přiblíží k levému okraji snímku, je provedena rotace směrem doleva. Analogický postup se uplatní i v případě pravého rohu rámečku blížícímu se k pravému okraji snímku, přičemž bude snímek rotován doprava. Pokud by se sledovaný objekt během celé videosekvence ani jednou nepřiblížil k okraji snímku, k žádnému posunu či rotaci by tak vůbec nedošlo. Popisovaný princip je demonstrován na obrázku 5.5.

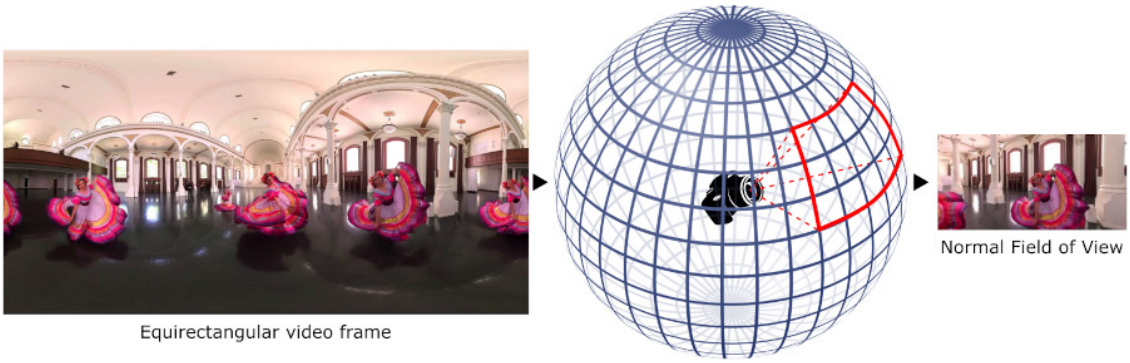
Obrázek 5.5: Demonstrace vylepšení označeného jako *BORDER*. Vlevo jsou ekvirektangulární snímky, ve kterých probíhá samotný proces sledování objektů a vpravo jsou ekvirektangulární snímky načtené přímo z videa. Je zřejmé, že na této videosekvenci došlo k situaci, kdy se sledovaný objekt dostal přes pravý okraj originálního snímku. Nicméně díky simulaci sférické rotace byl tracker schopen sledovaný objekt stále zaměřovat. Tento proces sledování odpovídá výsledkům upravené implementace trackeru Ocean [96].

Jemný posun začíná ve chvíli, kdy je vzdálenost rámečku od levého, respektive pravého okraje snímku menší než 1/5 celé šířky ekvirektangulárního snímku. Ve chvíli, kdy je vzdálenost rámečku od levého, respektive pravého okraje snímku menší než 1/8 šířky ekvirektangulárního snímku, je prováděn razantnější posun, aby nedošlo k přechodu rychle se pohybujícího objektu přes okraj. Konkrétní hodnoty posunu byly stanoveny experimentálně a jejich změna by mohla mít vliv na celkové výsledky. Hodnoty stanovených konstant lze nalézt přímo v samotných modulech v programové části této práce. Navržený způsob vylepšení bude v této práci dále označován jako *BORDER*, jelikož řeší prakticky pouze přechod objektu přes okraje ekvirektangulárních snímků.

Tato podsekce prezentovala jednoduché vylepšení, které lze označit například jako simulaci sférické rotace kolem osy z nebo jako reprojekci vstupních ekvirektangulárních snímků. Účelem tohoto vylepšení je řešit problém sledování objektu, jenž může přecházet mezi okraji ekvirektangulární projekce. Tato modifikace má velký potenciál výrazně zlepšit úspěšnost trackerů, které nemají výrazný problém s radiálním zkreslením nacházejícím se v ekvirektangulární projekci. Pro budoucí řešení se nabízí myšlenka přesné implementace sférických rotací. Díky tomu by mohla být prováděna i rotace kolem osy y ve sférické soustavě souřadnic [85], která by mohla sledovaný objekt držet na vertikálním středu ekvirektangulární projekce. Sledovaný objekt by se tak nemohl dostat na horní či spodní části ekvirektangulární projekce, které jsou velmi silně zkresleny.

Řešení na základě převodu do rektilineární projekce

Druhé vylepšení se zaměřilo na odstranění radiálního zkreslení, které může mít rovněž negativní dopad na proces sledování objektů v ekvirektangulární projekci 360° videa. Projekce, která takové odstranění umožňuje, se nazývá rektilineární nebo obecně gnomopická[10]. Tato projekce teoreticky odpovídá běžnému snímku s omezeným úhlem záběru, přičemž omezení úhlu záběru je pro mapování do této projekce klíčové. Na rozdíl od ekvirektangulární 360° × 180° projekce lze v rektilineárním zobrazení zobrazit nejvýše úhel záběru ∼(114° × 114°). Případný vyšší úhel již totiž obsahuje zkreslení, a jedná se tak už o stereografickou či perspektivní projekci [91]. Demonstraci rektilineárního mapování ilustruje obrázek 5.6. Následující obrázky a rovnice byly převzaty z blogu[11] popisujícího postup tohoto mapování.

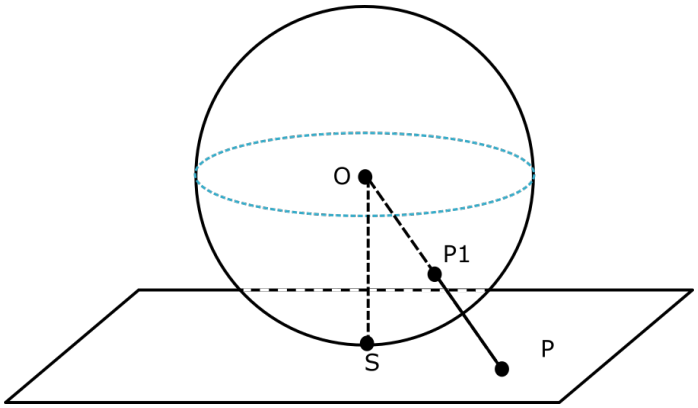


Obrázek 5.6: Ilustrace mapování snímku z ekvirektangulárního zobrazení do rektilineární (gnomopické) projekce.

[10] https://wiki.panotools.org/Rectilinear_Projection

[11] http://blog.nitishmutha.com/equirectangular/360degree/2017/06/12/How-to-project-Equirectangular-image-to-rectilinear-view.html

Pro realizaci rektilineárního mapování je využita existující Python implementace[12]. Tato převzatá implementace je upravena pro účely sledování objektu ve vytvořené rektilineární projekci. Mapování sférického snímku do rektilinární projekce je zde založeno na rovnicích 5.1, 5.2 a 5.3. Předpokládejme, že bod S na obrázku 5.7 odpovídá sférickým souřadnicím $(\theta_0, \varphi_0) = (0, 0)$ a současně středu zobrazené roviny.



Obrázek 5.7: Zobrazení části sféry na rovinu, přičemž bod O je počátek sférické soustavy souřadnic [85] a bod S odpovídá sférickým souřadnicím $(\theta_0, \varphi_0) = (0, 0)$ a současně středu zobrazené roviny. Bod P zobrazený na rovině pak odpovídá bodu $P1$ ve sférické soustavě.

$$x = \frac{\cos\theta \sin(\varphi - \varphi_0)}{\cos(c)} \tag{5.1}$$

$$y = \frac{\cos\theta_0 \sin\theta - \sin\theta_0 \cos\theta \cos(\varphi - \varphi_0)}{\cos(c)} \tag{5.2}$$

$$\cos(c) = \sin\theta_0 \sin\theta + \cos\theta_0 \cos\theta \cos(\varphi - \varphi_0) \tag{5.3}$$

V uvedených rovnicích odpovídá x horizontální souřadnici a y vertikální souřadnici bodu v kartézské soustavě souřadnic ve výsledném mapování rektilineární projekce. Symboly θ a φ představují úhly, respektive souřadnice konkrétního bodu ve sférické soustavě souřadnic podle anotace odpovídající popisu v úvodu této sekce. Převzatá implementace tedy umožňuje mapovat libovolnou část sféry do rektilineární projekce, přičemž je vždy definován střed cílové projekce. Rovnice 5.1 a 5.2 jsou provedeny současně pro každý potřebný pixel, jenž má být mapován a to prostřednictvím optimalizované implementace založené na NumPy[13]. Pro vyhlazení výstupního rektilineárního snímku je využíváno bilineární interpolace.

Princip samotného vylepšení trackeru tedy spočívá v udržování sledovaného objektu v úhlu záběru rektilineární projekce. Nejprve je označen sledovaný objekt pomocí rámečku v ekvirektangulární projekci (v kartézské soustavě souřadnic [81]). Poté je provedeno mapování části ekvirektangulárního snímku do rektilineární projekce, přičemž středem rektilineární projekce je střed označeného objektu. Současně s tím je přepočtena i pozice rámečku z ekvirektangulární projekce do rektilineární projekce. Samotný tracker je pak inicializován rektilineárním snímkem a rámečkem, který byl přepočten do souřadnic rektilineárního

[12] https://github.com/NitishMutha/equirectangular-toolbox
[13] https://numpy.org/

snímku. Proces sledování objektu pak probíhá po celou dobu v rektilineární projekci a pozice predikovaného rámečku jsou přepočítávány zpětně do originální ekvirektangulární projekce.

Je zřejmé, že z malého úhlu záběru videa v rektilineární projekci může sledovaný objekt zmizet (*Out of View*) podobně jako z úhlu záběru běžného videa. Představené vylepšení se proto snaží držet sledovaný objekt ve středu rektilineární projekce a tudíž při ideálním průběhu procesu sledování tento objekt z úhlu záběru nezmizí. Toto vylepšení by tedy šlo formulovat jako princip virtuální kamery, která drží sledovaný objekt co nejblíže středu rektilineární projekce. Pohyb virtuální kamery není na po sobě jdoucích snímcích skokový, ale jedná se pouze o posun o několik málo pixelů. Pro tento posun proto byly definovány konstanty na základě toho, jak rychle se objekt pohybuje. Případná analýza stanovení těchto konstant by jistě mohla mít velký přínos pro vylepšení přesnosti navrženého přístupu.

Jak již bylo zmíněno, pro vytvoření rektilineárního snímku pomocí převzaté implementace je zapotřebí definovat střed cílové rektilineární projekce, který odpovídá určitému bodu v ekvirektangulární projekci. Celý proces adaptace závisí především na tom, s jakou úspěšností tracker predikuje rámeček. V případě, kdy tracker selže, dospěje proces sledování objektů k téměř bezvýchodné situaci, jelikož sledovaný objekt může z úhlu záběru zmizet. V takovém případě navržený přístup pravděpodobně dosáhne velmi nepřesných výsledků. Pokud se naopak podaří objekt udržet v úhlu záběru po celou dobu videosekvence, může dojít k velmi přesnému procesu sledování objektu, které navíc probíhá ve snímcích bez výrazného radiálního zkreslení. Teoreticky se tak tracker může přiblížit výsledkům, kterých dosahuje v běžném videu s omezeným úhlem záběru.

Toto vylepšení má ovšem oproti sledování objektu v ekvirektangulární projekci jednu zřejmou nevýhodu, která plyne rovněž z omezeného úhlu záběru. Pokud se sledovaný objekt přiblíží velmi blízko virtuální rektilineární kameře, může dojít k selhání celého procesu sledování. Toto vylepšení s tímto problémem počítá a snaží se ho kompenzovat úpravou úhlu záběru (*field of view*) rektilineární projekce. Výchozí úhel záběru je definován 90° v horizontální směru ekvirektangulární projekce a příslušným úhlem ve vertikálním směru. Představené řešení umožňuje během procesu sledování adaptovat i tento úhel záběru, který se v případě blížícího objektu může zvyšovat. Blížící se objekt lze v takovém případě identifikovat podle velikosti predikovaného rámečku, která přesahuje určitou část výšky či šířky rektilineárního snímku. V takové situaci prezentované řešení provádí simulaci *zoomu* virtuální kamery, respektive mění se úhel záběru zobrazený v rektilineární projekci.

Bylo stanoveno, že horizontální úhel záběru lze v navrženém řešení měnit od 90° do 144°. Z toho vyplývá, že může dojít i k přesažení limitu úhlu záběru rektilineární projekce. V případě, že tento úhel přesáhne přibližně 114°, se začne objevovat mírné perspektivní zkreslení. Poté se již tak jedná o perspektivní projekci, která ovšem umožní alespoň částečné řešení zmíněného problému přiblížení. I v tomto případě byly konstanty pro operace oddálení, respektive zpětného přiblížení stanoveny experimentálně. Pro budoucí řešení se tak otevírá možnost analyzovat konkrétní konstanty, které je možné nalézt v programové části této diplomové práce. Navržené vylepšení bylo označeno jako *NFOV (Normal Field Of View)* a jeho ilustraci lze nalézt na obrázku 5.8.

Závěrem je vhodné doplnit skutečnost, že princip zmíněné virtuální kamery byl inspirován článkem [92], kde je uveden přístup zaměřený na generování běžných videosekvencí ze sférického videa[14]. Díky tomuto generování je možné vytvořit konkrétní videosekvenci,

[14] https://www.youtube.com/watch?v=XBQBePzMiMw

která může být následně využita například jako ideální scénář pro testování konkrétního problému úlohy sledování objektů. Tento přístup generování videosekvencí byl využit i pro vytvoření datasetu *VOT-LT2019*[15] za účelem vyhodnocení *long-term* trackerů. Přístup sledování objektů v rektilineární projekci se již objevil také v implementaci[16] metody sledování více objektů [90], kde je kromě samotného sledování objektů řešena také jejich 3D lokalizace.

Obrázek 5.8: Demonstrace vylepšení označeného jako *NFOV (Normal Field Of View)*, které umožňuje eliminaci radiálního zkreslení. Vlevo jsou rektilineární snímky, ve kterých probíhá samotný proces sledování objektů a vpravo jsou ekvirektangulární snímky načtené přímo z videa, do kterých je predikovaný rámeček zpětně převáděn. Je zřejmé, že na této videosekvenci došlo k situaci, kdy se sledovaný objekt dostal přes okraj originálního snímku. Díky rektilineární projekci se zde však úspěšně podařilo zaměřovat rychle jedoucí vozidlo. Tento proces sledování odpovídá výsledkům upravené implementace trackeru KYS [6].

[15] https://www.votchallenge.net/vot2019/dataset.html
[16] https://github.com/fandulu/MPLT

5.3 Implementace systému

Primárním cílem této práce byla implementace samotných přístupů pro sledování objektů v panoramatickém videu. Je tedy zřejmé, že je zde kladen důraz právě na tuto progresivní oblast počítačového vidění. Proto také nebylo záměrem vytvořit konkrétní aplikaci a nebyly uvedeny žádné návrhy související například s grafickým uživatelským rozhraním. V této práci vznikl jednoduchý systém ve formě Python balíku (*Python package*)[17] a výsledná podoba programové části této práce má tedy podobu přehledné adresářové struktury sestávající se z Python modulů. Jazyk Python byl zvolen mimo jiné i proto, že v současné době patří mezi nejvíce využívané jazyky pro implementaci metod z oblasti zpracování obrazu. Hlavním důvodem volby tohoto jazyka však byla skutečnost, že pomocí něj a jemu dostupných balíků bylo možné implementovat všechna představená vylepšení, která byla založena na úpravě projekcí. Pokud by se však práce zaměřila na konkrétní úpravy interní implementace některého trackeru, bylo by pravděpodobně výhodnější zvolit například jazyk C++, který by mohl umožnit lepší optimalizaci implementace.

Kromě samotného jazyka Python byly pro implementaci vybrány i další knihovny. Celý systém je postaven na knihovně *OpenCV*[18], respektive na její rozšířené verzi[19], která obsahuje implementace trackerů zmíněných v kapitole 3. Dále je zde využito balíku *NumPy*[20], který je přidán především pro implementaci úprav ekvirektangulárních snímků. Instalace těchto dvou knihoven či balíků by měla být dostačující pro spuštění *OpenCV* trackerů společně s vlastními vylepšeními *BORDER* a *NFOV*, která byla implementována za účelem sledování objektů v ekvirektangulární projekci 360° videa. Jak bude uvedeno v kapitole 6, bylo provedeno vyhodnocení celkem 5 trackerů [2, 41, 42, 37, 54] z knihovny *OpenCV*.

V této práci byly rovněž vytvořeny kódy pro úpravu oficiálních implementací 7 dalších trackerů z externích zdrojů. Implementace trackerů ECO [17], ATOM [18], DiMP [5] a KYS [6] byly převzaty z oficiálního repozitáře jejich autorů[21]. Dále byla použita implementace[22] trackeru DaSiamRPN [93] a třetím externím zdrojem byl repozitář[23] obsahující implementace trackeru SiamDW [95] a Ocean [96]. Instalace těchto 7 trackerů je ovšem popsána zvlášť formou souborů README, jelikož jsou tyto implementace postaveny na platformě *Anaconda*[24]. Jejich instalace proto vyžaduje zadání cesty souborového systému právě k nainstalované platformě *Anaconda*. Programová část této práce tedy poskytuje vylepšení pro všech 12 trackerům, přičemž pro spuštění *OpenCV* trackery stačí spustit hlavní instalační skript a pro případné zprovoznění implementace zbývajících 7 trackerů stačí postupovat podle přiložených návodů, které jsou ve formátu Markdown[25].

[17] https://docs.python.org/3/tutorial/modules.html#packages
[18] https://opencv.org/
[19] https://pypi.org/project/opencv-contrib-python/
[20] https://numpy.org/
[21] https://github.com/visionml/pytracking
[22] https://github.com/foolwood/DaSiamRPN
[23] https://github.com/researchmm/TracKit
[24] https://www.anaconda.com/
[25] https://en.wikipedia.org/wiki/Markdown

Obrázek 5.9: Ukázka anotace snímku pomocí jednoduchého nástroje, který byl vytvořen za účelem tvorby vlastního datasetu. Tento nástroj je postaven na *mouse events* a *keyboard events*, které lze zachytit pomocí knihovny *OpenCV*.

Všechny zmíněné trackery byly včetně jejich vylepšení *BORDER* a *NFOV* následně vyhodnoceny. Součástí programové části jsou tedy i kódy pro jejich evaluaci. Pro potřebné výpočty a operace s tenzory byl využit framework *PyTorch*[26]. Vizualizace výsledků byla poté zprostředkována pomocí balíku *Matplotlib*[27] [39] a balíku *seaborn*[28]. Byly zde rovněž využity i balíky *statsmodels*[29] a *pandas*[30] za účelem provedení statistické analýzy pro velké množství získaných výsledků.

Kromě možnosti spuštění trackerů byly implementovány také moduly, jež umožňují vykreslení výsledků procesu sledování objektů. Kromě nich byl vytvořen i vlastní anotační nástroj (obr. 5.9), který vznikl za účelem manuálních anotací pro tvorbu nového vlastního datasetu 360° videí v ekvirektangulární projekci. Je možné si všimnout, že tento nástroj umožňuje anotovat i objekty, jež jsou na okraji ekvirektangulárního snímku. Pro tuto možnost je vytvořena vlastní jednoduchá třída pro reprezentaci rámečku (*bounding boxu*). Všechny trackery i další moduly lze jednoduše spustit z příkazové řádky.

Veškeré moduly byly přehledně komentovány a přímo ve zdrojových kódech byly vyznačeny všechny části inspirované existujícími návody a také části obsahující převzaté zdrojové kódy. Vytvořené třídy a metody byly popsány na základě konvencí formátu Docstring[31]. Výsledné zdrojové kódy byly při finálním dokončování práce zveřejněny rovněž ve formě *GitHub* repozitáře[32].

[26] https://pytorch.org/
[27] https://matplotlib.org/
[28] https://seaborn.pydata.org/
[29] https://www.statsmodels.org/stable/index.html
[30] https://pandas.pydata.org/
[31] https://en.wikipedia.org/wiki/Docstring
[32] https://github.com/VitaAmbroz/360Tracking

Kapitola 6

Vyhodnocení

Tato kapitola nejdříve prezentuje dataset s vlastními anotacemi pro 360° videa v ekvirektangulární projekci a uvádí také důvody, které vedly k jeho vytvoření. Dále jsou zde popsány metriky, pomocí kterých bylo provedeno detailní vyhodnocení 12 trackerů a navržených vylepšení. Výsledky vyhodnocení jsou zde následně vizualizovány formou grafů a tabulek. Na konci této kapitoly jsou uvedeny i výstupy statistické metody, která byla provedena pro analýzu rozptylu velkého množství získaných výsledků.

6.1 Dataset

Aby bylo možné objektivně vyhodnotit výsledky úlohy sledování objektů, je žádoucí mít k dispozici dataset videosekvencí s označenými objekty. Anotované referenční objekty se označují jako *groundtruth* data a reprezentují ideální průběh procesu sledování daného objektu. Implementace trackerů je možné aplikovat na videa z referenčního datasetu a ukládat výsledky procesu sledování. Pro inicializaci trackeru je použita *groundtruth* anotace objektu na prvním snímku videa. Získané výsledky lze poté porovnat s *groundtruth* daty.

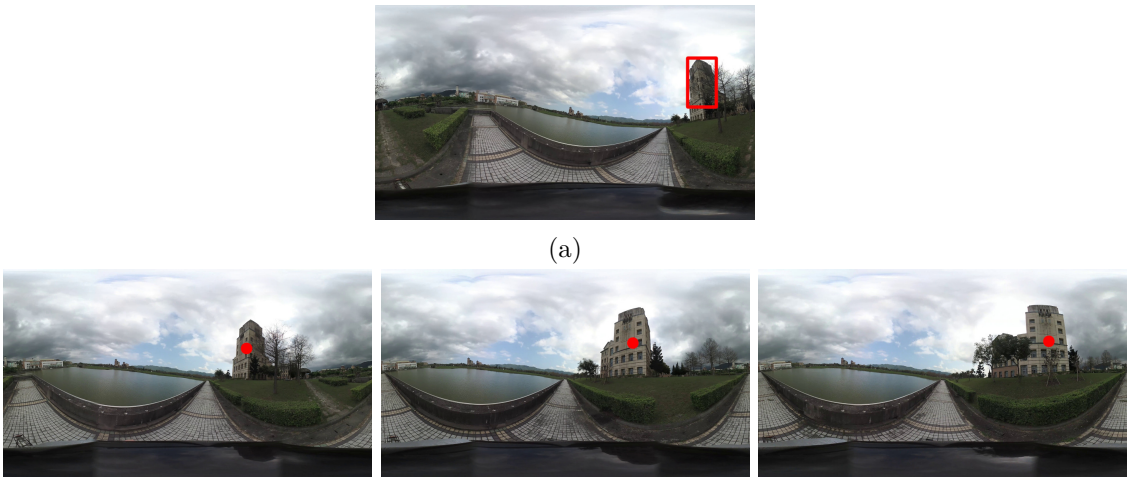
Během studia existujících metod sledování objektů bylo nalezeno několik dostupných datasetů [88, 45, 57, 88, 38], které byly zmíněny již v kapitole 3. Tyto datasety ovšem obsahují pouze běžné snímky, a proto je tedy nelze využít i pro evaluaci sledování objektů v panoramatickém videu. Pro účely sledování objektů v 360° videu dosud pravděpodobně nevznikl žádný standardizovaný *benchmark*, který by byl založen na anotacích objektů v tomto specifickém typu videa. Pro jiné úlohy počítačového vidění byly již vytvořeny pokročilé *groundtruth* datasety 360° videosekvencí[1], které ovšem neobsahují anotace potřebné pro vyhodnocení úlohy sledování objektů.

Ve velké části nalezených článků, které se zaměřily na sledování objektů v 360° videu, byla provedena také evaluace představených metod. Některé články [11] však v podstatě neobsahují klíčové informace o vyhodnocení a datasetu, na kterém bylo provedeno. Jiné články [24, 23, 21, 97, 43] sice vyhodnocení představují podrobněji, ale jejich datasety nebyly zveřejněny. Během řešení této diplomové práce byly kontaktováni autoři [24, 23, 21, 43] za účelem využití jejich datasetů, ale bohužel se žádný z těchto datasetů nepodařilo získat.

Byla ovšem nalezena i práce zabývající se porovnáním trackerů přímo v ekvirektangulární projekci 360° [55], kde autoři evaluační dataset zveřejnily. Tento dataset však neobsahuje anotace rámečků (*bounding boxes*), ale pouze anotace bodů odpovídajících středu sledovaného objektu (viz obr. 6.1). Dále se také podařilo kontaktovat autory metody sle-

[1] https://salient360.ls2n.fr/datasets/benchmark-dataset/

dování více objektů [51] v ekvirektangulární projekci, kteří následně poskytli svůj dataset *App360*[2] pro možnost jeho použití v této diplomové práci. Anotace tohoto datasetu si lze prohlédnout na obrázku 6.2. Autoři zde odstranili nejvíce zkreslené části a omezili vertikální úhel záběru pouze na střední oblast v původní ekvirektangulární projekci.



(a) (b) (c) (d)

Obrázek 6.1: Demonstrace anotací v datasetu [55], kde je první snímek označen obdélníkovým rámečkem a pro následující snímky je již anotován pouze střed sledovaného objektu.

Obrázek 6.2: Demonstrace anotací v datasetu [51], kde je na každém snímku 360° videa označeno více objektů současně. V tomto datasetu byl ovšem výrazně snížen vertikální úhel záběru oproti původní ekvirektangulární projekci.

[2] https://github.com/KengChiLiu/MOT360

Právě nedostatek dostupných anotací vedl k motivaci pro vytvoření vlastního datasetu, který by mohl být použit pro vyhodnocení prezentovaných vylepšení *BORDER* a *NFOV*. Pro vytvoření anotací pro úlohu sledování objektů lze využít některé existující automatizované nástroje. Tyto nástroje umožňují automatický návrh anotací na základě použitého trackeru či detektoru. Mohou být implementovány ve formě desktopových aplikací (ViTBAT[3][7]) nebo případně i webových aplikací (VATIC[4][5][73]). Lze nalézt také pokročilé komerční nástroje[6], které umožňují široký okruh formátů anotací a lze je využít například i pro vyhodnocení úlohy segmentace na základě anotací referenčních polygonů. Tyto nástroje byly ovšem vytvořeny pro anotaci běžných snímků a neumožňují tak anotaci objektu, který může přecházet mezi okraji snímku.

Právě tato skutečnost vedla k vytvoření jednoduchého anotačního nástroje, který byl uveden už v předchozí kapitole 5. Tento nástroj umožňuje realizovat i anotace pro scénáře, kdy se objekt nachází na okraji ekvirektangulárního snímku a byl tedy použit pro konstrukci nového vlastního datasetu. Před zahájením samotného označování byly zkoumány anotace v existujících datasetech běžných snímků (např. VOT[7], OTB[8]), aby mohly být vlastní anotace provedeny bez výrazných nepřesností a omylů, které by pramenily z neznalosti konvencí správných anotací rámečků. Při označování objektů tak byl vždy vybrán nejmenší možný rámeček (*bounding box*), který ohraničoval sledovaný objekt. Jestliže na některém snímku videosekvence docházelo k částečnému zakrytí *partial occlusion* či úplnému zakrytí (*full occlusion*) anotovaného objektu, bylo stanoveno, že pokud bude zakryta více než polovina objektu, nebude na takovém snímku snímku objekt označen.

Anotace pro každý snímek má stanovený formát (*frame_number, top_left_corner_x, top_left_corner_y, width, height*), kde *frame_number* je číslo snímku ve videosekvenci, (*top_left_corner_x, top_left_corner_y*) je pozice levého horního rohu rámečku v kartézském souřadném systému a hodnoty *width,height* odpovídají výšce, respektive šířce rámečku. Pro konstrukci datasetu byly využity oba datasety [55, 51] s ekvirektangulárními videosekvencemi, jejichž anotace byly ilustrovány na obrázcích 6.1 a 6.2. Použité videosekvence z těchto datasetů [55, 51] byly ovšem vlastním způsobem reanotovány a původní anotace těchto datasetů tak byly prakticky využity pouze pro inspiraci. Kromě nich byla anotována i 360° videa[9] z datasetu zaměřeného na orientaci uživatele v 360° videu [60]. Nakonec byla anotována i vlastní videa pořízená sférickou kamerou *Ricoh Theta SC*[10].

Vytvořený dataset se skládá z celkově 21 videosekvencí a 9909 anotovaných snímků. Nachází se zde 8 vlastních 360° videí a 13 videí ze tří externích zdrojů [55, 51, 60], přičemž všechna videa byla záměrně ponechána ve svém původním rozlišení. Jedná se celkem o 4 různá rozlišení – $1280\times720px$, $1920\times960px$, $3840\times2160px$ a $3840\times1920px$. Rozlišení $1920\times960px$ a $3840x1920px$ tedy přesně odpovídají poměru stran ekvirektangulární projekce $2:1$, ovšem rozlišení $1280\times720px$ a $3840\times2160px$ mají poměr stran $16:9$. Nejedná se tak o ryzí formát ekvirektangulární projekce, nicméně odlišnost poměrů $2:1$ a $16:9$ nebyla zhledána jako výrazně ovlivňující faktor výsledků procesu sledování. Klíčovou vlastností použitých videí je skutečnost, že pokrývají celý $360°\times180°$ úhel záběru. Všechna videa by

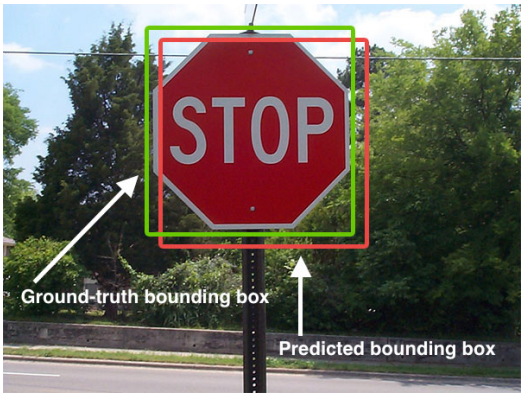
[3] https://vitbat.weebly.com/
[4] http://www.cs.columbia.edu/~vondrick/vatic/
[5] https://dbolkensteyn.github.io/vatic.js/
[6] https://sixgill.com/ai-powered-labeling/
[7] https://www.votchallenge.net/challenges.html
[8] http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html
[9] https://github.com/acmmmsys/2019-360dataset/tree/master/SampleVideos/Source
[10] https://theta360.com/en/about/theta/sc.html

v případě budoucího využití tohoto datasetu bylo možné renderovat do jednotného rozlišení, respektive by bylo možné i přepočíst všechny anotované pozice objektů.

Výsledný dataset byl zveřejněn[11] a může být využit i dalšími autory, kteří se v budoucnu budou zabývat problémem sledování objektů v ekvirektangulární projekci 360° videa. Byly vytvořeny i vizualizované ukázky[12][13] *groundtruth* anotací pro jednotlivé videosekvence tohoto datasetu. Pro možné budoucí vylepšení tohoto datesetu se nabízí například možnost převodu pozic z kartézského souřadného systému [81] do sférické soustavy souřadnic [85]. Mohlo by se tak docílit podoby datasetu, který byl vytvořen v rámci metody generování běžných videosekvencí ze sférického videa [92]. Právě dataset a anotační nástroj, jež byly prezentovány v článku [92], mohly být velkou inspirací pro vytvoření vlastního datasetu, ale autoři je bohužel nezveřejnili.

6.2 Metriky pro vyhodnocení

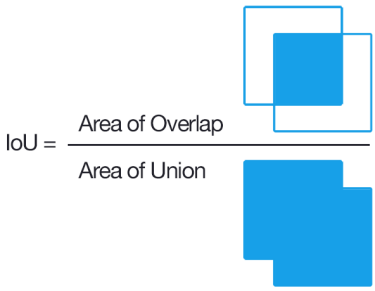
Ve chvíli, kdy jsou k dispozici výsledky procesu sledování vybraného trackeru a současně také *groundtruth* anotace (viz obr 6.3), je možné provést vlastní vyhodnocení. Během studia existujících datasetů či *benchmarků* [88, 45, 57, 25, 38] bylo pozorováno vyhodnocení trackerů na základě několika různých metrik. Proto tuto práci byly zvoleny dvě základní metriky přesnosti, které se objevily i v představených datasetech [88, 45, 57, 25, 38] a také v samotných článcích prezentujících trackery [47, 18, 5, 6]. Obrázky uvedené v této sekci byly převzaty z webového článku[14], jenž popisuje metriku *IoU (Intersection over Union)*.



Obrázek 6.3: Objekt (dopravní značka) je ohraničen zeleným referenčním rámečkem (*Ground-truth bounding box*). Je zde zobrazen také červený rámeček jakožto výsledek predikce metody sledování či detekce objektů (*Predicted bounding box*).

První použitá metrika se nazývá *IoU (Intersection over Union)* a definuje poměr překrytí referenčního a predikovaného rámečku oproti celkovému obsahu obou rámečků. Metrika *IoU* hodnotí jednak přesnost pozice predikovaného rámečku a jednak také přesnou velikost predikovaného rámečku. Tato metrika se může uplatnit jako základ vyhodnocení nejen pro metody sledování objektů, ale také například pro související úlohu detekce objektů. Výpočet této metriky je vizuálně ilustrován pomocí obrázku 6.4.

[11] https://drive.google.com/drive/folders/13tkE4vY3FGGD42kDIjyS9K423vrvpKoU
[12] https://www.youtube.com/watch?v=kgXd6uoXa8M
[13] https://www.youtube.com/watch?v=7hHXhmMCmQ8
[14] https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/



Obrázek 6.4: Ilustrace výpočtu metriky *IoU (Intersection over Union).*

Formálně lze výpočet metriky *IoU* zapsat pomocí rovnice 6.1, kde B_G odpovídá referenčnímu rámečku (*groundtruth bounding box*) a B_P představuje rámeček, který predikoval tracker. Pokud je výpočet hodnoty *IoU* proveden pro každý snímek videosekvence, lze vytvořit funkce demonstrující přesnost výsledků, které tracker predikoval. Příkladem může být funkce přesnosti (*accuracy*) představená pro *VOT challenge* [45].

$$IoU = \frac{B_G \cap B_P}{B_G \cup B_P} \tag{6.1}$$

Druhá metrika bude v této práci označena jako *Precision* [57]. Lze se ovšem setkat i s dalšími označeními, například *Location Error* či *Center Error*. Prakticky se jedná o euklidovskou vzdálenost[15] mezi středem S_G referenčního rámečku a středem S_P predikovaného rámečku. V této práci probíhají výpočty v kartézském souřadném systému, a proto lze popisovanou metriku zapsat pomocí L2 normy uvedené v rovnici 6.2. Tato metrika určuje pouze přesnost pozice predikovaného rámečku a nikoliv už přesnost jeho velikosti.

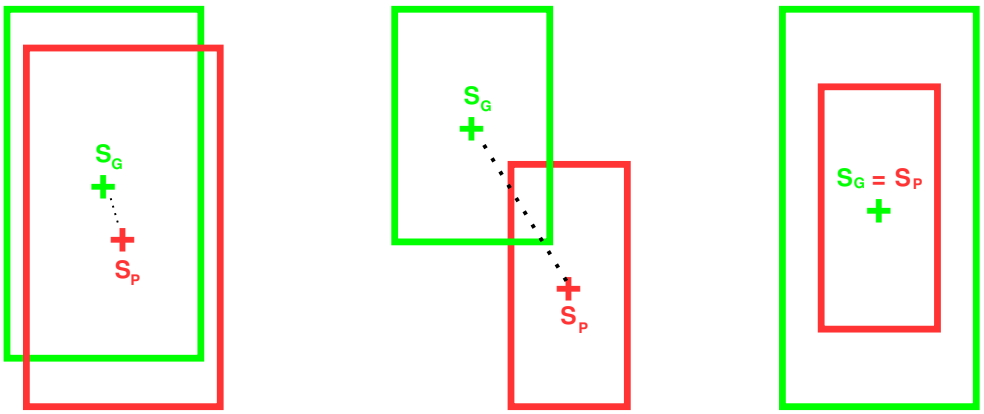
$$Precision = \|S_G - S_P\|_2 \tag{6.2}$$

Lze se setkat i s normalizovanou variantou metriky *Precision*, která se obvykle označuje jako *Normalized Precision* [57]. Její použití je vhodné ve chvíli, když se liší poměr stran a velikost videosekvencí, pro které vyhodnocení úlohy sledování objektů probíhá. V předchozí sekci 6 bylo řečeno, že vytvořený dataset obsahuje různá rozlišení, která odpovídají formátům HD, FullHD a 4K. Tím vzniká velký rozsah od nejvyššího rozlišení 4K až po rozlišení HD. Proto byla provedena vlastní normalizace metriky *Precision* a to na základě výšky nejmenšího videa v datasetu, která odpovídá hodnotě $720px$. Hodnota *Precision* je zde tedy normalizována pro výsledky videosekvencí s rozlišením FullHD a 4K. Prakticky se jedná o jednoduchý výpočet definovaný rovnicí 6.3, který je založen na výšce ($Video\_Height$) konkrétního videa, přičemž hodnota výšky videí v datasetu má pouze hodnoty $720px$, $960px$, $1920px$ a $2160px$. Vlastní normalizace prakticky neovlivnila výsledky, které by jinak byly formálně získány pomocí metriky *Normalized Precision*. Vlastní normalizace proběhla za předpokladu, že všechny videosekvence odpovídají ekvirektangulárnímu poměru stran $2:1$, respektive velmi blízkému poměru $16:9$. Kdyby se zde objevovalo více různých poměrů stran, pak by bylo využito originální metriky *Normalized Precision* [57].

$$Normalized\_Precision = Precision * \frac{720}{Video\_Height} \tag{6.3}$$

[15] https://en.wikipedia.org/wiki/Euclidean_distance

Využití obou těchto metrik umožňuje porovnat trackery pro přesnost predikované pozice rámečku a současně také pro přesnost velikosti navrženého rámečku. Rozdíl výsledků metrik *IoU* a *Precision* je ilustrován na obrázku 6.5. Na tomto obrázku je jasně patrné, v jakých situacích budou výsledky obou metrik stejně úspěšné, respektive stejně neúspěšné a kdy se naopak výsledky metrik budou výrazně lišit.



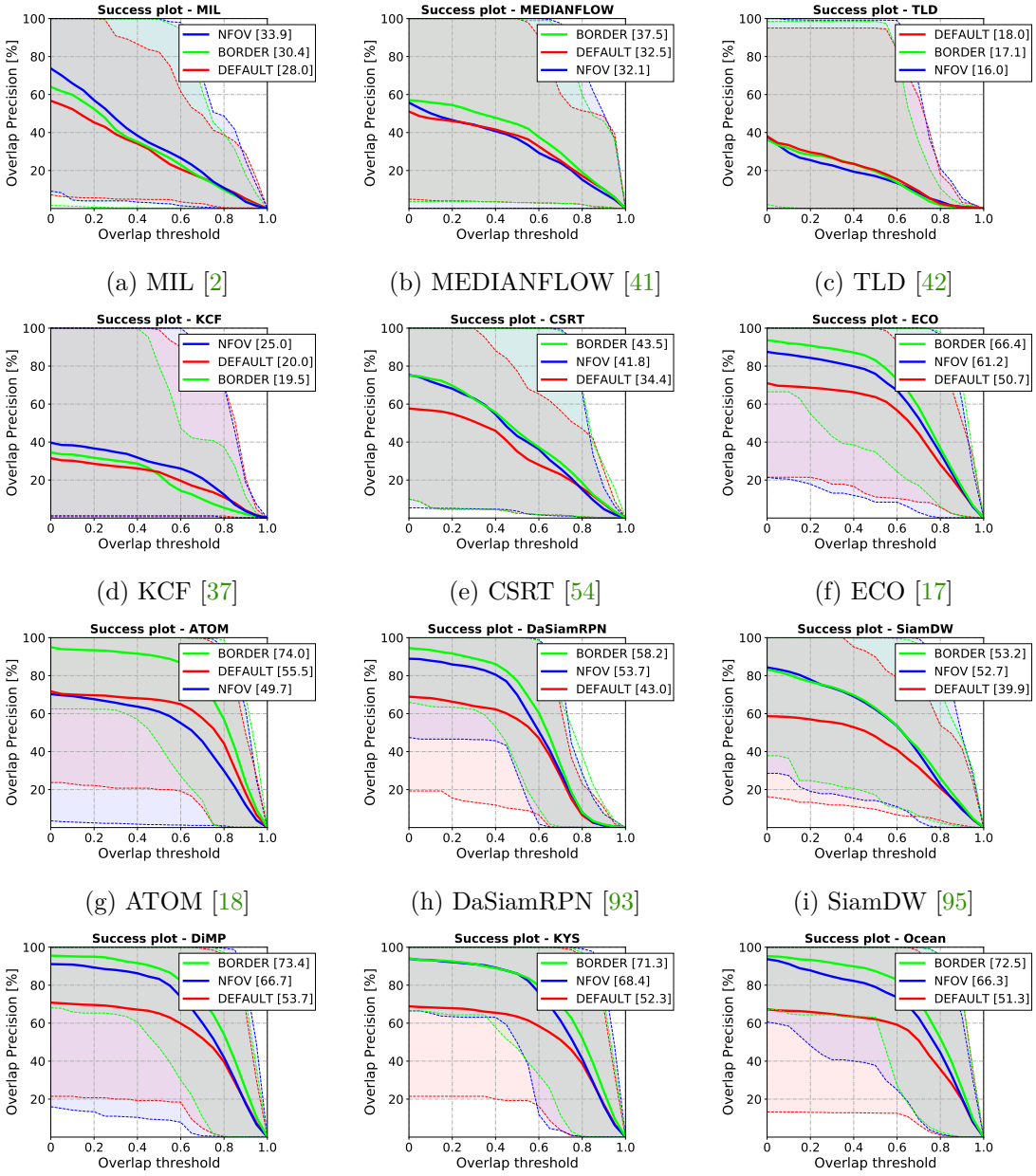
Obrázek 6.5: Na obrázku jsou znázorněny tři různé scénáře, kde zelený rámeček se středem S_G představuje referenční anotaci objektu a červený rámeček se středem S_P odpovídá predikci trackeru. Scénář vlevo ilustruje přesný výsledek a scénář uprostřed naopak nepřesný výsledek pro obě metriky *IoU* i *Precision*. Situace vpravo signalizuje poměrně nepřesný výsledek pro metriku *IoU* a naopak nejlepší možný výsledek pro metriku *Precision*.

6.3 Výsledky

Samotné vyhodnocení bylo realizováno celkem pro 12 trackerů, které byly uvedeny v kapitole 5. Pro každý z těchto trackerů byly vytvořeny tři verze. První verze, označená jako *DEFAULT*, představuje přímé využití originální implementace trackeru pro účely sledování v ekvirektangulární projekci 360° videa. Označení *BORDER* odpovídá vylepšení pomocí sférické rotace a *NFOV* je název vylepšení, kdy proces sledování probíhá v rektilinární projekci pomocí virtuální kamery. Každá verze byla spuštěna pro každý tracker a to pro všech 21 videosekvencí vytvořeného datasetu.

Na základě získaných výsledků byly vypočteny metriky *IoU (Intersection over Union)* a *Precision*. Díky hodnotám metriky *IoU* bylo možné sestrojit tzv. *Success* grafy. Tyto grafy zobrazují závislost procentuálního počtu přesných výsledků (*Overlap precision*) na tolerované míře překrytí (*Overlap threshold*) podle hodnoty *IoU*. Podle *Success* křivky lze vypočíst i hodnotu *AUC (Area Under Curve)*, která odpovídá ploše pod *Success* křivkou. Na obrázku 6.6 jsou zobrazeny grafy pro všech 12 trackerů ve třech verzích *DEFAULT*, *BORDER* a *NFOV*. V legendě grafů jsou pak uvedeny odpovídající hodnoty *AUC*.

Podle metriky *Precision* byly sestrojeny také tzv. *Precision* grafy, které zobrazují závislost procentuálního počtu přesných výsledků (*Distance precision*) na tolerované odchylce (*Location error threshold*) podle hodnoty *Precision*. Na obrázku 6.7 jsou zobrazeny tyto grafy pro rovněž pro všech 12 trackerů ve třech různých verzích. V legendě grafů jsou pak uvedeny hodnoty, které udávají, kolik procent středů predikovaných rámečků se odchýlilo maximálně o vzdálenost $20px$ od středů *groundtruth* rámečků. Ve všech grafech je znázorněn také rozptyl, jehož vznik byl zapříčiněn nevyrovnanými výsledky pro všech 21 videosekvencí vlastního datasetu.



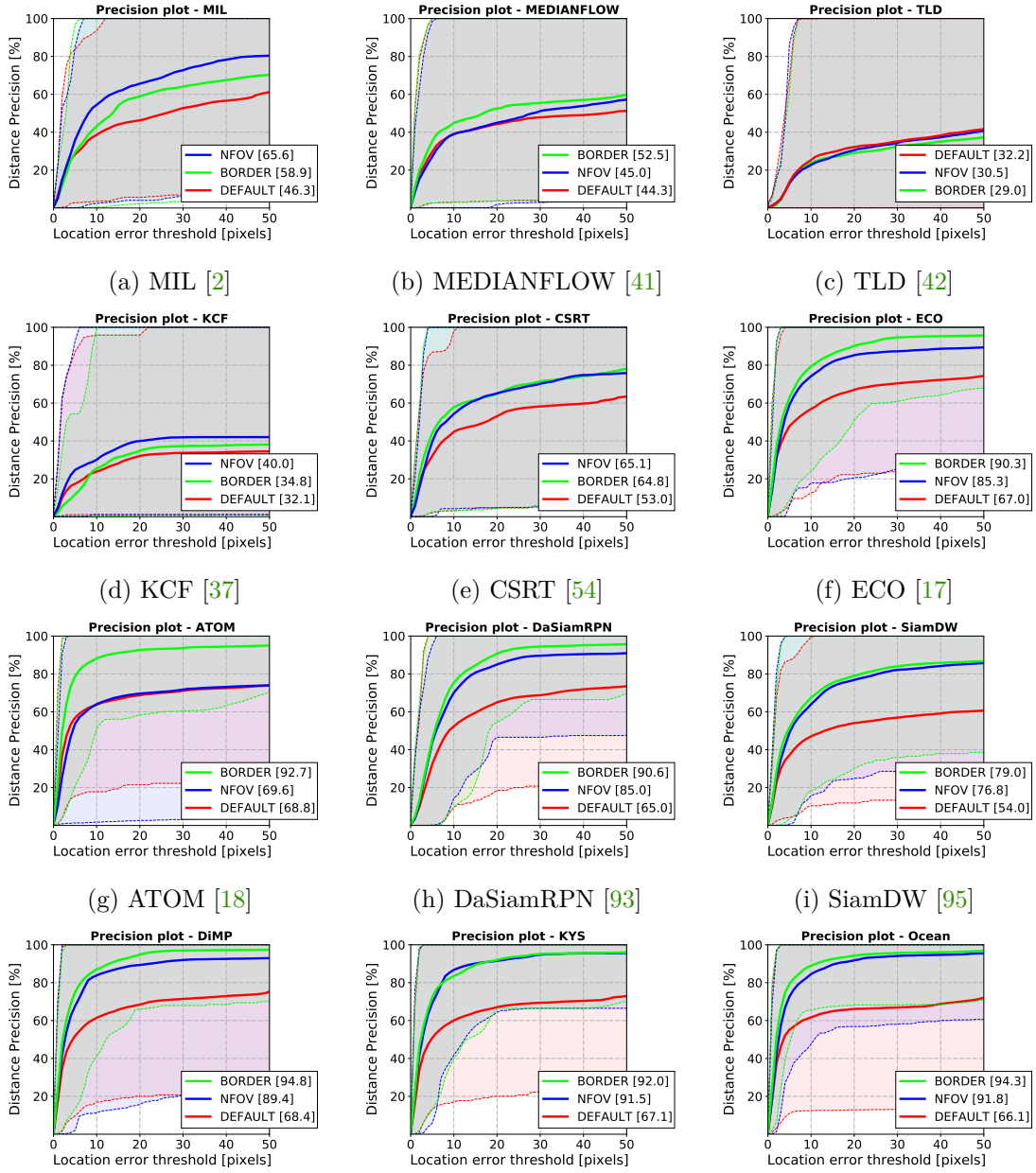
(a) MIL [2] (b) MEDIANFLOW [41] (c) TLD [42]

(d) KCF [37] (e) CSRT [54] (f) ECO [17]

(g) ATOM [18] (h) DaSiamRPN [93] (i) SiamDW [95]

(j) DiMP [5] (k) KYS [6] (l) Ocean [96]

Obrázek 6.6: *Success* grafy jednotlivých trackerů ve třech různých verzích. Červená barva odpovídá přímému využití trackeru v ekvirektangulární projekci (*DEFAULT*), zelená barva představuje vylepšení trackeru založené na sférické rotaci (*BORDER*) a modrá barva reprezentuje vylepšení trackeru postavené na sledování objektu v rektilineární projekci (*NFOV*). V legendě grafů jsou zobrazeny hodnoty *AUC (Area Under Curve)*. Ve všech grafech je formou světlého pozadí znázorněn také rozptyl, jehož vznik byl zapříčiněn nevyrovnanými výsledky pro všech 21 videosekvencí vlastního datasetu.



(a) MIL [2] (b) MEDIANFLOW [41] (c) TLD [42]

(d) KCF [37] (e) CSRT [54] (f) ECO [17]

(g) ATOM [18] (h) DaSiamRPN [93] (i) SiamDW [95]

(j) DiMP [5] (k) KYS [6] (l) Ocean [96]

Obrázek 6.7: *Precision* grafy jednotlivých trackerů ve třech různých verzích. Červená barva odpovídá přímému využití trackeru v ekvirektangulární projekci (*DEFAULT*), zelená barva představuje vylepšení trackeru založené na sférické rotaci (*BORDER*) a modrá barva reprezentuje vylepšení trackeru postavené na sledování objektu v rektilineární projekci (*NFOV*). V legendě grafů jsou pak uvedeny hodnoty, které udávají, kolik procent středů predikovaných rámečků se odchýlilo maximálně o vzdálenost $20px$ od středů *groundtruth* rámečků. Ve všech grafech je formou světlého pozadí znázorněn také rozptyl, jehož vznik byl zapříčiněn nevyrovnanými výsledky pro všech 21 videosekvencí vlastního datasetu.

Již samotné grafy na obrázcích 6.6 a 6.7 poměrně dobře vizualizují vliv provedených vylepšení *BORDER* a *NFOV*. Hodnoty *AUC* a *Precision*, které jsou uvedeny v legendách těchto grafů, je možné nalézt také v tabulce 6.1. Kromě nich jsou v této tabulce uvedeny také hodnoty OP_{50} a OP_{75}. Tyto hodnoty odpovídají procentuálnímu počtu průměrných výsledků (*Overlap Precision*), jejichž hodnota metriky *IoU* přesáhla hodnotu 0.5, respektive 0.75. Pro vylepšení *BORDER* a *NFOV* byly dále vytvořeny tabulky hodnot *AUC* i *Precision* pro jednotlivé videosekvence vlastního datasetu. Tyto tabulky je možné nalézt v příloze C a hodnoty v nich uvedené by mohly být velký přínos při potencionálním vylepšení představeného datasetu. Díky nim lze totiž snadno identifikovat videosekvence, které byly pro konkrétní vylepšení a trackery nejvíce problematické.

	MIL [2]	MF [41]	TLD [42]	KCF [37]	CSR [54]	ECO [17]	ATO [18]	DSR [93]	SDW [95]	DMP [5]	KYS [6]	OCE [96]	∅
AUC-DEFAULT	28.0	32.5	18.0	20.0	34.4	50.7	55.5	43.0	39.9	53.7	52.3	51.3	39.9
AUC-BORDER	30.4	37.5	17.1	19.5	43.5	66.4	74.0	58.2	53.2	73.4	71.3	72.5	51.4
AUC-NFOV	33.9	32.1	16.0	25.0	41.8	61.2	49.7	53.7	52.7	66.7	68.4	66.3	47.2
OP_{50}-DEFAULT	27.2	38.4	20.2	24.2	35.1	63.7	67.2	57.1	48.0	65.1	63.5	62.1	47.7
OP_{50}-BORDER	29.5	44.4	19.6	22.8	45.6	83.1	90.0	77.3	62.9	88.8	86.1	87.4	61.4
OP_{50}-NFOV	32.2	36.7	17.1	28.4	43.2	75.9	61.0	69.9	62.3	83.3	86.1	79.0	56.3
OP_{75}-DEFAULT	13.8	21.0	5.4	13.5	19.6	36.3	52.2	15.1	26.2	47.0	45.9	42.8	28.2
OP_{75}-BORDER	13.2	24.2	3.7	7.4	24.2	47.7	67.5	18.6	34.1	64.6	62.7	62.4	35.9
OP_{75}-NFOV	14.1	20.1	5.4	16.7	20.9	43.3	37.6	17.5	30.9	51.9	51.9	53.7	30.3
Prec-DEFAULT	46.3	44.3	32.2	32.1	53.0	67.0	68.8	65.0	54.0	68.4	67.1	66.1	55.3
Prec-BORDER	58.9	52.5	29.0	34.8	64.8	90.3	92.7	90.6	79.0	94.8	92.0	94.3	72.8
Prec-NFOV	65.6	45.0	30.5	40.0	65.1	85.3	69.6	85.0	76.8	89.4	91.5	91.8	69.6

Tabulka 6.1: Tabulka obsahuje hodnoty AUC (*Area Under Curve*), OP_{50}, OP_{75} a *Precision* pro všechny trackery a všechny tři verze *DEFAULT, BORDER, NFOV*. Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96]. Názvy trackerů byly v tabulce pro přehlednost zkráceny na maximálně 3 znaky.

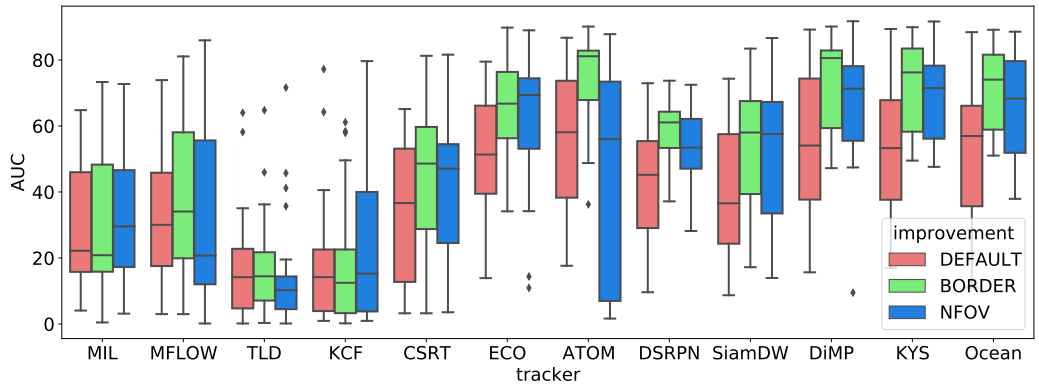
Pro účely správného zobrazení rozptylu byly pro hodnoty *AUC* i *Precision* vytvořeny také krabicové grafy zobrazené na obrázcích 6.8 a 6.9. Na obrázku 6.10 jsou poté zobrazeny *Success* a *Precision* grafy výsledků všech trackerů pro jednotlivá vylepšení. Na základě všech zmíněných vizualizací lze pozorovat výrazný nárůst úspěšnosti procesu sledování, který umožnila obě vylepšení *BORDER* i *NFOV*. Ve všech grafech i v tabulce 6.1 jsou trackery seřazeny podle roku jejich vzniku, přičemž nejstarší tracker MIL [2] a nejnovější tracker Ocean [96] dělí mezi sebou více než 10 let vývoje v oblasti sledování objektů.

Celkově nejhorších výsledků dosahovaly *OpenCV* implementace trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37] a CSRT [54]. Nejnovější *OpenCV* tracker CSRT [54] dosáhl celkově nejlepších výsledků sledování a lze jej tak považovat za aktuálně nejpřesnější tracker dostupný v této knihovně, což potvrdila i dříve zmíněná práce [55]. Je nutné poznamenat, že implementace trackerů MIL [2] a KCF [37] jako jediné neumožňovaly sledování objektů s adaptivní velikostí rámečku. Výsledky těchto trackerů byly touto skutečností silně ovlivněny, jelikož hned v několika videosekvencích sledovaný objekt měnil svoji velikost i tvar. Celkově nejhorším trackerem byla metoda TLD [42], u jejíž *OpenCV* implementace byla pozorována skutečnost, že predikuje velké množství falešně pozitivních výsledků.

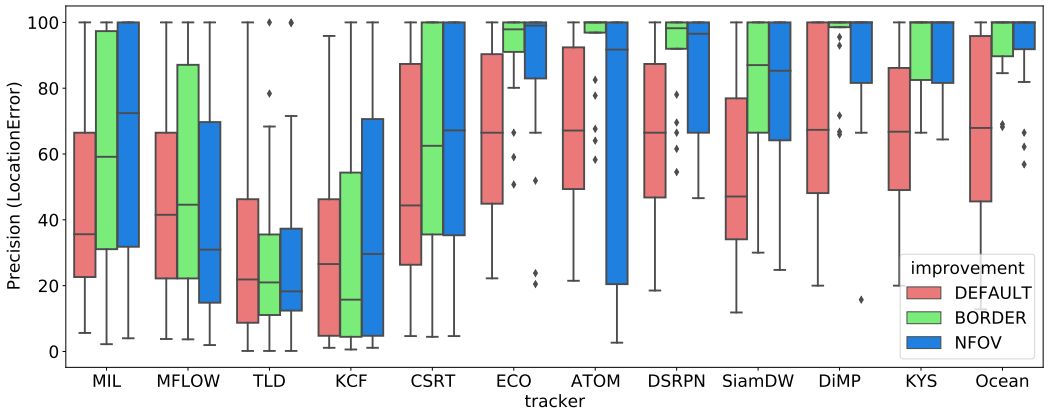
Implementace 7 ostatních trackerů (ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96]) dosáhly výrazně přesnějších výsledků než *OpenCV* trackery. Za nejpřesnější trackery pro sledování objektů přímo v ekvirektangulární projekci

360° videa lze pro verze *DEFAULT i BORDER* považovat *state-of-the-art* metody ATOM [18], DiMP [5], Ocean [96] a KYS [6]. Vyhodnocení ukázalo, že se zmíněné trackery umí velmi dobře vypořádat i s radiálním zkreslením v ekvirektangulární projekci.

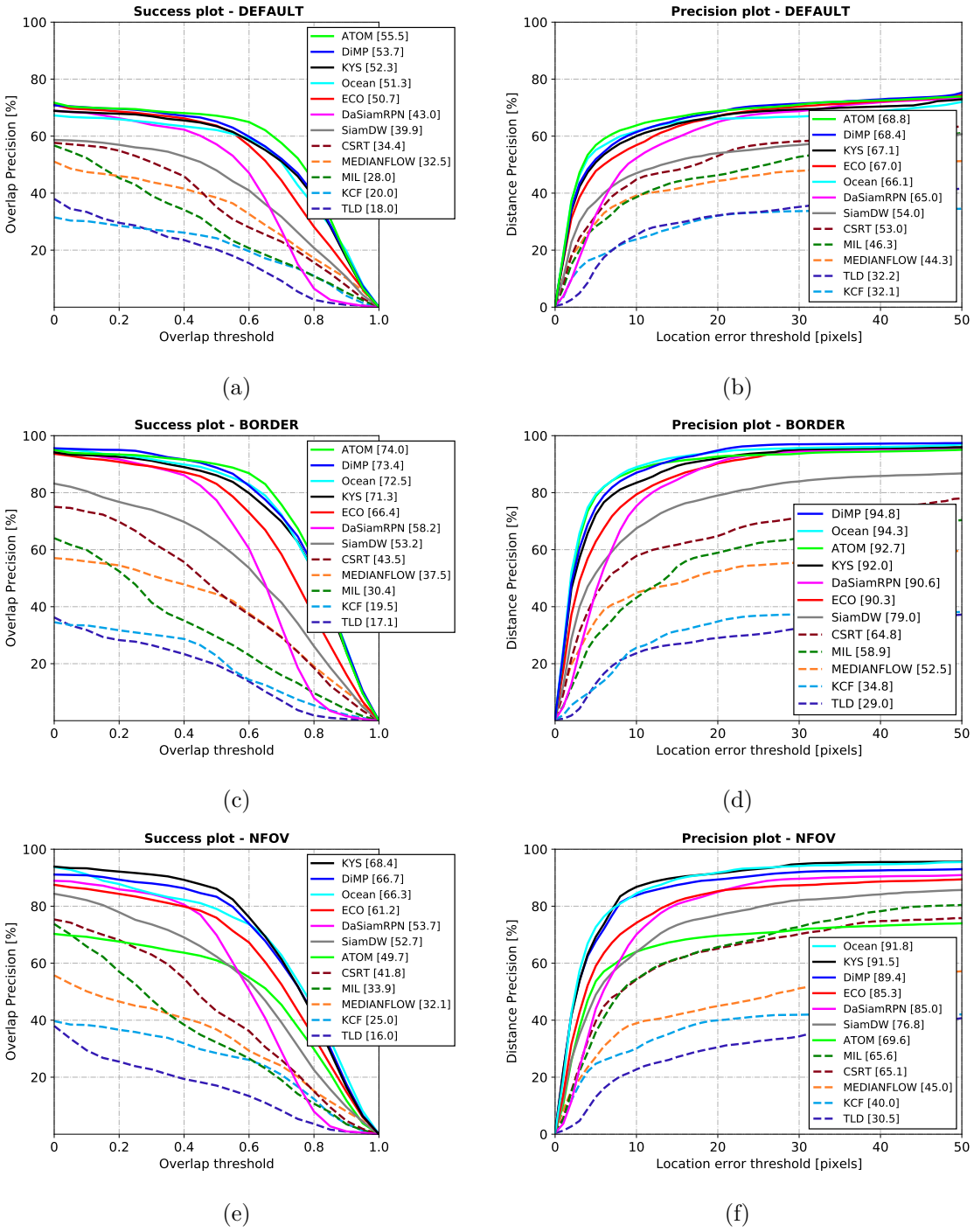
Sledování objektů v rektilineární projekci (*NFOV*) překvapivě dosáhlo v průměru mírně horších výsledků než vylepšení *BORDER*. Konkrétně jinak nejúspěšnější tracker ATOM [18] dosahoval pro verzi vylepšení *NFOV* velmi nepřesných výsledků. Rektilineární projekce může tedy pro některé trackery zapříčinit výrazně horší výsledky, což může být paradoxně způsobeno právě omezeným úhlem záběru. Virtuální kamera se snaží držet sledovaný objekt na středu rektilinárního zobrazení, což může mít negativní dopad na *motion model*, který je pro výsledky některých trackerů velmi důležitý. Obecně nejlepších výsledků tak pro obě provedená vylepšení tedy celkově dosáhly trackery KYS [6], DiMP [5] a Ocean [96].



Obrázek 6.8: Krabicový graf (*boxplot*) založený na hodnotách *AUC*, který názorně demonstruje rozptyl tří verzí modifikací trackerů (*DEFAULT, BORDER, TRACKER*). Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96].



Obrázek 6.9: Krabicový graf (*boxplot*) založený na hodnotách *Precision*, který názorně demonstruje rozptyl tří verzí modifikací trackerů (*DEFAULT, BORDER, TRACKER*). Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96].

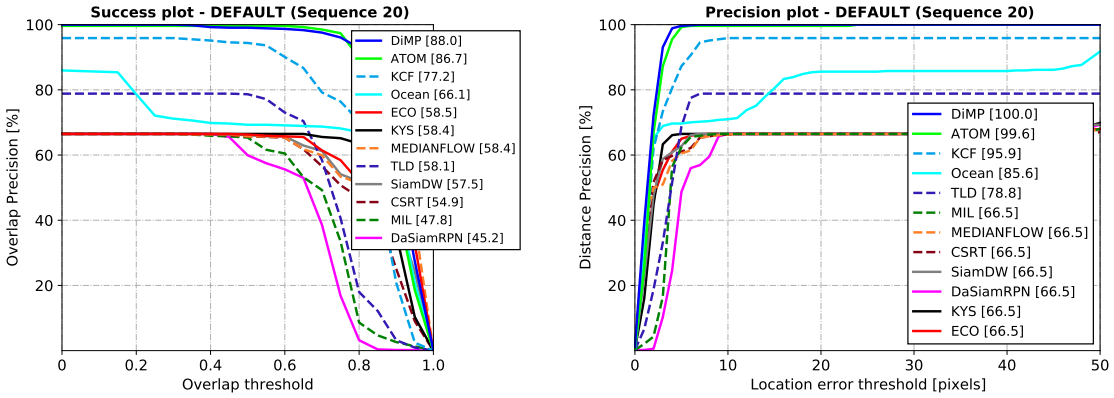


Obrázek 6.10: *Success* a *Precision* grafy pro všech 21 videosekvencí. (a) (b) Popisek *DEFAULT* představuje výsledky trackerů bez modifikace, (c) (d) *BORDER* výsledky trackerů s vylepšením sférické rotace a (e) (f) *NFOV* výsledky trackerů v rektilineární projekci. Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96].

Experiment se zakrytým objektem

Při vyhodnocení byly pozorovány různé problémy pro konkrétní sekvence datasetu. Ve videosekvenci s číslem 20 byl anotován scénář, kde došlo k úplnému zakrytí (*full occlusion*) sledovaného objektu. Díky této situaci je tak možné částečné odhadnout úspěšnost testovaných trackerů při sledování objektů, kde dochází ke zmíněnému jevu *full occlusion*. Pro objektivní vyhodnocení by však bylo nutné tento scénář na více anotovaných videosekvencích. Obrázek 6.11 demonstruje průběh videa, kde došlo ke zkoumanému problému.

Obrázek 6.11: Sledovaný objekt (lidský obličej) je během této videosekvence několikrát částečně i úplně zakryt. Každý snímek na tomto obrázku odpovídá úhlu záběru 90° × 180° z původní ekvirektangulární projekce. Ostatní části původní projekce byly oříznuty za účelem zvýraznění sledovaného objektu.



Obrázek 6.12: *Success* a *Precision* graf s výsledky trackerů pro videosekvenci č. 20.

Z grafů (obr. 6.12) je zřejmé, že v této videosekvenci dosáhly nejvyšší přesnosti trackery DiMP [5], ATOM [18], KCF [37] a Ocean [96]. Především *OpenCV* implementace trackeru KCF [37] a také trackeru TLD [42] zde predikovaly rámeček pro sledovaný objekt mnohem lépe než u jiných videosekvencí. Výsledky ostatní trackerů jsou si v tomto případě velmi podobné, jelikož jejich proces sledování selhal při prvním úplném zakrytí (*full occlusion*).

Jsou zde uvedeny pouze grafy přímého využití trackerů (*DEFAULT*), jelikož navržená vylepšení *BORDER* a *NFOV* by v této videosekvenci nenalezly výrazné uplatnění. Nedochází zde totiž k přechodu objektu přes okraj a sledovaný objekt je umístěn ve vertikálním středu projekce, kde se objevuje jen velmi mírné radiální zkreslení.

6.4 Analýza rozptylu

Ve výsledcích získaných z vlastního datasetu byl pozorován poměrně velký rozptyl úspěšnosti konkrétních trackerů a provedených vylepšení pro každý tracker. Rozptyl byl v minulé sekci vizualizován formou několika grafů, nicméně kromě samotné vizualizace byla provedena i jeho analýza. Analýza rozptylu (*ANOVA – Analysis of variance*) je statistická metoda [79], která umožňuje ověření, zda na hodnotu náhodné veličiny pro určitého jedince má statisticky významný vliv hodnota některého faktoru, jenž je možné u tohoto jedince pozorovat. Konkrétní faktor přitom musí nabývat konečného počtu možných hodnot (alespoň dvou různých) a umožňuje tak rozdělit jedince do vzájemně porovnávaných skupin.

Pro náš případ byly zvoleny celkem tři faktory či znaky, tudíž byla provedena 3-faktorová analýza rozptylu (*3-factor ANOVA / 3-way ANOVA*). Prvním zkoumaným faktorem byla videosekvence (*sequence*) ve vlastním datasetu, který obsahoval celkově 21 anotovaných sekvencí. Dalším faktorem bylo provedené vylepšení (*improvement*) trackerů, přičemž jejich označení je identické jako u již dříve uvedených grafů – *DEFAULT*, *BORDER*, *NFOV*. Třetím faktorem byla samotná metoda sledování objektu (*tracker*).

Všechny kombinace 12 trackerů ve třech verzích byly spuštěny pro 21 videosekvencí, díky čemuž bylo získáno celkem 756 různých výsledků. Tyto výsledky byly následně vyhodnoceny a byly získány hodnoty metrik *AUC* a *Precision*. Pro obě tyto metriky a zmíněné kombinace faktorů byla poté provedena analýza rozptylu. Tyto kombinace, respektive vstupní data metody *ANOVA* ilustruje tabulka 6.2, kde se nachází hodnoty metriky *AUC* jakožto hodnoty náhodné veličiny. Na rozdíl od předchozích grafů a tabulek, kdy byly hodnoty *AUC* i *Precision* v rozmezí $(0, 100)$, zde byly hodnoty normalizovány do intervalu $(0, 1)$.

	AUC	sequence	improvement	tracker
0	0.109	01	DEFAULT	MIL
1	0.176	01	DEFAULT	MFLOW
2	0.307	01	DEFAULT	TLD
3	0.122	01	DEFAULT	KCF
⋮	⋮	⋮	⋮	⋮
752	0.675	21	NFOV	SiamDW
753	0.888	21	NFOV	DiMP
754	0.883	21	NFOV	KYS
755	0.879	21	NFOV	Ocean

Tabulka 6.2: Ukázka vstupních dat pro metodu analýzy rozptylu

Tato vstupní data byla tedy následně použita pro 3-faktorovou analýzu rozptylu. Její přesná implementace proběhla podle dostupných návodů[16] s využitím balíku *statsmodels*[17]. Bylo by jistě možné podrobně popsat statistickou metodu *ANOVA* formou všech rovnic

[16] https://www.pythonfordatascience.org/factorial-anova-python/
[17] https://www.statsmodels.org/stable/anova.html

a principů, které se v případě její 3-faktorové varianty uplatňují. Z hlediska rozsahu zde budou ovšem uvedeny pouze klíčové pojmy, díky kterým bude následně možné prezentovat výsledky této statistické metody formou tabulky. Každý řádek takové tabulky představuje krok od jednoduššího modelu ke složitějšímu modelu. Prakticky lze tedy na implementovanou podobu analýzy rozptylu nahlížet jako na černou skříňku (*black box*), díky které bude možné identifikovat staticky významné znaky, které ovlivnily výsledky procesu sledování.

V prvním sloupci tabulky se obvykle udává konkrétní faktor. V tomto případě se tedy bude jednat o již uvedené faktory *sequence, improvement, tracker*, respektive jejich interakce (např. *sequence*improvement*). Druhý sloupec poté představuje součet čtverců (*sum of squares*), který udává míru celkového rozptylu pro konkrétní faktor a závislou proměnnou (v tomto případě tuto proměnnou představují hodnoty *AUC, Precision*). Je vhodné doplnit, že pro výpočet součtu čtverců (*sum of squares*) je využito tzv. typu III, který byl ve většině zdrojů doporučen[18][19]. Třetí sloupec obsahuje stupně volnosti (*df – degrees of freedom*), které definují kolika různých hodnot konkrétní faktor nebo kombinace faktorů nabývá. Na základě součtu čtverců a stupňů volnosti pak může být vypočtena průměrná hodnota *Mean Square*, která v tabulkách ovšem uvedena nebude. *Mean square* se kromě faktorů a jejich interakce vypočte i pro tzv. reziduální součet čtverců a maximální stupně volnosti, jejichž hodnota je v tomto případě stanovena jako $df(sequence) * df(improvement) * df(tracker)$.

Následně je vypočtena hodnota *F-testu*, která odpovídá poměru mezi *Mean Square* konkrétního faktoru oproti *Mean Square* pro reziduální prvky a představuje tak porovnání faktorů s celkovou odchylkou. Díky hodnotě *F-testu* a konstantě α lze poté vypočíst hodnotu p odpovídající pravděpodobnosti potřebné k určení statisticky významných faktorů. Hodnota konstanty α by měla být velmi malá (typicky 0.05). V případě, kdy je pro konkrétní faktory vypočtená hodnota p menší než konstanta α, lze prohlásit, že tyto faktory mají statisticky významný vliv na hodnotu sledované metriky (*AUC, Precision*). Pokud je však hodnota p vyšší než konstanta α, nelze potvrdit, že by konkrétní faktor nebo interakce faktorů měly statisticky významný vliv na hodnoty sledovaných metrik. V tabulkách 6.3 a 6.4 jsou uvedeny výsledky metody *ANOVA* pro metriky *AUC* a *Precision*.

Pro interpretaci tabulek 6.3 a 6.4 jsou klíčové hodnoty p. V případě, že je pro konkrétní faktor hodnota p menší než konstanta $\alpha = 0.05$, pak lze prohlásit, že je konkrétní faktor statisticky významný. Pro obě sledované metriky *AUC, Precision* byly pro každý faktor i interakce faktorů hodnoty p velmi malé, respektive výrazně menší než $\alpha = 0.05$. Lze tedy prohlásit, že všechny tři faktory (*sequence, improvement, tracker*) včetně jejich interakcí měly významný statistický vliv na výsledky procesu sledování.

	sum_squares	df	F	p(>F)
sequence	11.84	20	63.24	< 0.001
improvement	1.70	2	91.01	< 0.001
tracker	20.38	11	197.95	< 0.001
sequence*improvement	2.69	40	7.18	< 0.001
sequence*tracker	9.98	220	4.85	< 0.001
improvement*tracker	1.32	22	6.39	< 0.001
Residual	4.12	440		

Tabulka 6.3: 3-faktorová analýza rozptylu (*3-way ANOVA*) hodnot metriky *AUC*.

[18] https://towardsdatascience.com/anovas-three-types-of-estimating-sums-of-squares-don-t-make-the-wrong-choice-91107c77a27a

[19] http://www.utstat.toronto.edu/reid/sta442f/2009/typeSS.pdf

	sum_squares	df	F	p(>F)
sequence	19.03	20	43.90	< 0.001
improvement	4.35	2	100.41	< 0.001
tracker	26.21	11	109.94	< 0.001
sequence*improvement	5.67	40	6.54	< 0.001
sequence*tracker	19.55	220	4.10	< 0.001
improvement*tracker	2.07	22	4.34	< 0.001
Residual	9.54	440		

Tabulka 6.4: 3-faktorová analýza rozptylu (*3-way ANOVA*), která je založena na hodnotách metriky *Precision (Location error)*.

Při experimentech s analýzou rozptylu byla mimo jiné zjištěna i souvislost s 8 videosekvencemi, kde nedošlo k přechodu objektu mezi okraji ekvirektangulární projekce. Tabulka 6.5 odpovídá výsledkům metody *ANOVA* aplikované pouze pro metriku *AUC* ve zmíněných 8 videosekvencích. V této tabulce byly indikovány mírně vyšší hodnoty pro faktor vylepšení (*improvement*). Jelikož hodnota $p = 0.0912$ je vyšší než konstanta $\alpha = 0.05$, nelze faktor *improvement* prohlásit za statisticky významný pro výsledky procesu sledování objektu na těchto 8 videosekvencích. Současně s tím ale nelze tvrdit, že faktor *improvement* statisticky významný není, jelikož takový výrok výsledky metody *ANOVA* neumožňují.

	sum_squares	df	F	p(>F)
sequence	2.75	7	63.66	< 0.001
improvement	0.03	2	2.43	0.0912
tracker	9.13	11	134.41	< 0.001
sequence*improvement	0.25	14	2.90	< 0.001
sequence*tracker	3.15	77	6.63	< 0.001
improvement*tracker	0.42	22	3.14	< 0.001
Residual	0.95	154		

Tabulka 6.5: 3-faktorová analýza rozptylu (*3-way ANOVA*) metriky *AUC (Area Under Curve)* provedená pouze pro 8 videosekvencí datasetu, kde nedocházelo k problému přechodu objektu přes okraj ekvirektangulární projekce. Faktor *sequence* zde má tedy pouze 7 stupňů volnosti, na rozdíl od předchozích analýzy, která byla provedena pro celý dataset obsahující 21 videosekvencí.

Byly vytvořeny také *Success* a *Precison* grafy zvlášť pro videosekvence s přechodem objektu a zvlášť také pro videosekvence bez přechodu objektu (viz příloha B). Na základě těchto grafů a výsledků metody *ANOVA* je možné se domnívat, že provedená vylepšení nemají výrazný vliv na výsledky sledování objektů v ekvirektangulární projekci, kde objekt nepřechází přes okraj. Toto tvrzení by však bylo nutné dále zkoumat a potvrdit například při vyhodnocení na rozsáhlejším datasetu, který by obsahoval pouze videosekvence bez přechodu objektu. Nicméně skutečnost, že se proces sledování objektů výrazně nezhorší aplikací provedených vylepšení, je samozřejmě záměrem těchto modifikací. Modifikace *BORDER* je pro taková videa prakticky shodná jako přímé využití trackeru (*DEFAULT*). Pokud se objekt nepřiblíží k okraji, nebude docházet ani k žádné rotaci. U vylepšení *NFOV* by však mohlo být přínosné podrobněji zkoumat jeho vliv.

Kapitola 7

Závěr

Tato diplomová práce se zaměřila na úlohu sledování jediného objektu v 360° videu. Na počátku řešení problematiky této úlohy byla snaha o vylepšení některých existující přístupů pro sledování objektů v 360° panoramatickém videu. Vedle tohoto snažení byla provedena rešerše vývoje metod, která odhalila skutečnost, že dosud bylo představeno velmi málo řešení sledování jediného objektu v ekvirektangulární projekci 360° videa. Směr této práce se tedy následně orientoval především na vlastní vylepšení, která spočívají v adaptaci ekvirektangulárních snímků pro účely sledování objektů. První vylepšení je postaveno na principu sférické rotace a umožňuje tak sledovat konkrétní objekt i na přechodu mezi okraji ekvirektangulárních snímků. Druhé vylepšení je založeno na převodu sledované části do rektilineární projekce, díky čemuž je možné objekty sledovat bez radiálního zkreslení.

Za účelem vyhodnocení zmíněných vylepšení bylo vybráno 12 existujících implementací metod sledování objektů (trackerů). Pro každý tracker byly následně implementována obě zmíněná vylepšení. V rámci tohoto vyhodnocení byl vytvořen i vlastní dataset, který obsahuje 21 videosekvencí v ekvirektangulární projekci s celkově 9909 ručně provedenými anotacemi. Jedná se pravděpodobně o první dataset, který umožňuje přesné vyhodnocení scénáře, kdy se sledovaný objekt může pohybovat na hranici ekvirektangulárních snímků. Vyhodnocení bylo provedeno na základě metrik (*AUC, Precision*), které se v současnosti používají ve výpočetních výzvách zaměřených na úlohu sledování objektů. Toto vyhodnocení ukázalo nezanedbatelný vliv obou implementovaných přístupů, které dosáhly výrazně vyšší přesnosti než trackery bez vylepšení. Nejlepších výsledků dosáhly v součtu metody DiMP [5], KYS [6] a Ocean [96]. Nejvyšší úspěšnost procesu sledování probíhajícího přímo ekvirektangulární projekci měl však tracker ATOM [18], který ale zaznamenal výrazné zhoršení úspěšnosti při využití vylepšení založeného na rektilineární projekci.

Pro budoucí vývoj v oblasti sledování objektů v 360° panoramatickém videu se nabízí řada možností, kterými lze navázat na tuto práci. Implementované přístupy, jež jsou založeny na adaptaci ekvirektangulárních snímků, by mohly být dále optimalizovány pro využití v reálném čase. Další možností je adaptace modelů či sítí moderních algoritmů sledování objektů, které jsou mimo jiné založeny i na hlubokém učení. Takové adaptace by bylo vhodné docílit například pomocí učení na existujících datasetech běžných snímků, které by byly vhodným způsobem radiálně zkresleny. Dále by mohly být adaptovány i samotné implementace moderních algoritmů sledování objektů, například pomocí detekce přechodu sledovaného objektu na okraji ekvirektangulárního snímku. Variantou budoucího vývoje je i možnost rozšíření datasetu, který byl v této práci představen.

Literatura

[1] Agtzidis, I., Startsev, M. a Dorr, M. 360-degree Video Gaze Behaviour: A Ground-Truth Data Set and a Classification Algorithm for Eye Movements. In: *Proceedings of the 27th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2019-10-15, s. 1007–1015. DOI: 10.1145/3343031.3350947. ISBN 978-1-4503-6889-6. Dostupné z: https://dl.acm.org/doi/10.1145/3343031.3350947.

[2] Babenko, B., Yang, M.-H. a Belongie, S. Visual tracking with online Multiple Instance Learning. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, USA: IEEE, 2009, s. 983–990. DOI: 10.1109/CVPR.2009.5206737. ISBN 978-1-4244-3992-8. Dostupné z: https://ieeexplore.ieee.org/document/5206737/.

[3] Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A. a Torr, P. H. S. Fully-Convolutional Siamese Networks for Object Tracking. In: *Computer Vision – ECCV 2016 Workshops*. 1. vyd. Cham: Springer International Publishing, 2016, s. 850–865. DOI: 10.1007/978-3-319-48881-3_56. ISBN 978-3-319-48880-6. Dostupné z: http://link.springer.com/10.1007/978-3-319-48881-3_56.

[4] Bewley, A., Ge, Z., Ott, L., Ramos, F. a Upcroft, B. Simple online and realtime tracking. In: *2016 IEEE International Conference on Image Processing (ICIP)*. Phoenix, AZ, USA: IEEE, 2016, s. 3464–3468. DOI: 10.1109/ICIP.2016.7533003. ISBN 978-1-4673-9961-6. Dostupné z: http://ieeexplore.ieee.org/document/7533003/.

[5] Bhat, G., Danelljan, M., Gool, L. V. a Timofte, R. Learning Discriminative Model Prediction for Tracking. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, 2019, s. 6181–6190. DOI: 10.1109/ICCV.2019.00628. ISBN 978-1-7281-4803-8. Dostupné z: https://ieeexplore.ieee.org/document/9010649/.

[6] Bhat, G., Danelljan, M., Gool, L. V. a Timofte, R. Know Your Surroundings: Exploiting Scene Information for Object Tracking. In: *Computer Vision – ECCV 2020*. Cham: Springer International Publishing, 2020, s. 205–221. DOI: 10.1007/978-3-030-58592-1_13. ISBN 978-3-030-58591-4. Dostupné z: https://link.springer.com/10.1007/978-3-030-58592-1_13.

[7] Biresaw, T. A., Nawaz, T., Ferryman, J. a Dell, A. I. ViTBAT: Video tracking and behavior annotation tool. In: *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Colorado Springs, CO, USA: IEEE, 2016, s. 295–301. DOI: 10.1109/AVSS.2016.7738055. ISBN 978-1-5090-3811-4. Dostupné z: http://ieeexplore.ieee.org/document/7738055/.

[8] Bochkovskiy, A., Wang, C.-Y. a Liao, H.-Y. M. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. 2020. Dostupné z: https://arxiv.org/abs/2004.10934.

[9] Bolme, D., Beveridge, J. R., Draper, B. A. a Lui, Y. M. Visual object tracking using adaptive correlation filters. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, USA: IEEE, 2010, s. 2544–2550. DOI: 10.1109/CVPR.2010.5539960. ISBN 978-1-4244-6984-0. Dostupné z: http://ieeexplore.ieee.org/document/5539960/.

[10] Bouguet, J. yves. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*. 2000. Dostupné z: http://robots.stanford.edu/cs223b04/algo_tracking.pdf.

[11] Cai, C., Liang, X., Wang, B., Cui, Y. a Yan, Y. A Target Tracking Method Based on KCF for Omnidirectional Vision. In: *2018 37th Chinese Control Conference (CCC)*. Wuhan, China: IEEE, 2018, s. 2674–2679. DOI: 10.23919/ChiCC.2018.8483083. ISBN 978-988-15639-5-8. Dostupné z: https://ieeexplore.ieee.org/document/8483083/.

[12] Cheng, S. Y. a Trivedi, M. M. Lane Tracking with Omnidirectional Cameras: Algorithms and Evaluation. *EURASIP Journal on Embedded Systems*. 2007, sv. 2007, s. 1–8. DOI: 10.1155/2007/46972. ISSN 1687-3955. Dostupné z: https://link.springer.com/article/10.1155/2007/46972.

[13] Chou, S.-H., Sun, C., Chang, W.-Y., Hsu, W.-T., Sun, M. et al. 360-Indoor: Towards Learning Real-World Objects in 360° Indoor Equirectangular Images. In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Snowmass Village, CO, USA, USA: IEEE, 2020, s. 834–842. DOI: 10.1109/WACV45572.2020.9093262. ISBN 978-1-7281-6553-0. Dostupné z: https://ieeexplore.ieee.org/document/9093262/.

[14] Comaniciu, D., Ramesh, V. a Meer, P. Real-time tracking of non-rigid objects using mean shift. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*. Hilton Head, SC, USA: IEEE Comput. Soc, 2000, s. 142–149. DOI: 10.1109/CVPR.2000.854761. ISBN 0-7695-0662-3. Dostupné z: http://ieeexplore.ieee.org/document/854761/.

[15] Coors, B., Condurache, A. P. a Geiger, A. SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images. In: *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018, s. 525–541. DOI: 10.1007/978-3-030-01240-3_32. ISBN 978-3-030-01239-7. Dostupné z: http://link.springer.com/10.1007/978-3-030-01240-3_32.

[16] Corbillon, X., Simone, F. D. a Simon, G. 360-Degree Video Head Movement Dataset. In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. New York, NY, USA: ACM, 2017-06-20, s. 199–204. DOI: 10.1145/3083187.3083215. ISBN 9781450350020. Dostupné z: https://dl.acm.org/doi/10.1145/3083187.3083215.

[17] Danelljan, M., Bhat, G., Khan, F. S. a Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017, s. 6931–6939.

DOI: 10.1109/CVPR.2017.733. ISBN 978-1-5386-0457-1. Dostupné z: http://ieeexplore.ieee.org/document/8100216/.

[18] Danelljan, M., Bhat, G., Khan, F. S. a Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, 2019, s. 4655–4664. DOI: 10.1109/CVPR.2019.00479. ISBN 978-1-7281-3293-8. Dostupné z: https://ieeexplore.ieee.org/document/8953466/.

[19] Danelljan, M., Robinson, A., Khan, F. S. a Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In: *Computer Vision – ECCV 2016*. 1. vyd. Cham: Springer International Publishing, 2016, s. 472–488. DOI: 10.1007/978-3-319-46454-1_29. ISBN 978-3-319-46453-4. Dostupné z: http://link.springer.com/10.1007/978-3-319-46454-1_29.

[20] David, E. J., Gutiérrez, J., Coutrot, A., Da Silva, M. P. a Callet, P. L. A Dataset of Head and Eye Movements for 360° Videos. In: *Proceedings of the 9th ACM Multimedia Systems Conference*. New York, NY, USA: Association for Computing Machinery, 2018, s. 432–437. MMSys '18. DOI: 10.1145/3204949.3208139. ISBN 9781450351928. Dostupné z: https://doi.org/10.1145/3204949.3208139.

[21] Delforouzi, A. a Grzegorzek, M. Robust and Fast Object Tracking for Challenging 360-degree Videos. In: *2017 IEEE International Symposium on Multimedia (ISM)*. Taichung, Taiwan: IEEE, 2017, s. 274–277. DOI: 10.1109/ISM.2017.47. ISBN 978-1-5386-2937-6. Dostupné z: http://ieeexplore.ieee.org/document/8241613/.

[22] Delforouzi, A., Tabatabaei, S. A., Shirahama, K. a Grzegorzek, M. A Polar Model for Fast Object Tracking in 360-Degree Camera Images. *Multimedia Tools Appl.* USA: Kluwer Academic Publishers. duben 2019, sv. 78, č. 7, s. 9275–9297. DOI: 10.1007/s11042-018-6525-0. ISSN 1380-7501. Dostupné z: https://doi.org/10.1007/s11042-018-6525-0.

[23] Delforouzi, A., Tabatabaei, S. A. H., Shirahama, K. a Grzegorzek, M. Polar Object Tracking in 360-Degree Camera Images. In: *2016 IEEE International Symposium on Multimedia (ISM)*. San Jose, CA, USA: IEEE, 2016, s. 347–352. DOI: 10.1109/ISM.2016.0077. ISBN 978-1-5090-4571-6. Dostupné z: http://ieeexplore.ieee.org/document/7823644/.

[24] Delforouzi, A., Tabatabaei, S. A. H., Shirahama, K. a Grzegorzek, M. Unknown object tracking in 360-degree camera images. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. Cancun, Mexico: IEEE, 2016, s. 1798–1803. DOI: 10.1109/ICPR.2016.7899897. ISBN 978-1-5090-4847-2. Dostupné z: http://ieeexplore.ieee.org/document/7899897/.

[25] Fan, H., Ling, H., Lin, L., Yang, F., Chu, P. et al. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, 2019, s. 5369–5378. DOI: 10.1109/CVPR.2019.00552. ISBN 978-1-7281-3293-8. Dostupné z: https://ieeexplore.ieee.org/document/8954084/.

[26] Fassold, H. a Ghermi, R. OmniTrack: Real-Time Detection and Tracking of Objects, Text and Logos in Video. In: *2019 IEEE International Symposium on Multimedia (ISM)*. San Diego, CA, USA: IEEE, 2019, s. 245–2451. DOI: 10.1109/ISM46123.2019.00057. ISBN 978-1-7281-5606-4. Dostupné z: https://ieeexplore.ieee.org/document/8959038/.

[27] Fiaz, M., Mahmood, A., Javed, S. a Jung, S. K. Handcrafted and Deep Trackers: Recent Visual Object Tracking Approaches and Trends. *ACM Computing Surveys*. 2019-05-31, sv. 52, č. 2, s. 1–44. DOI: 10.1145/3309665. ISSN 0360-0300. Dostupné z: https://dl.acm.org/doi/10.1145/3309665.

[28] Foldesy, P., Szatmari, I. a Zarandy, A. Moving object tracking on panoramic images. In: *Proceedings of the 2002 7th IEEE International Workshop on Cellular Neural Networks and Their Applications*. Frankfurt, Germany: World Scientific, 2002, s. 63–70,. DOI: 10.1109/CNNA.2002.1035036. ISBN 981-238-121-X. Dostupné z: http://ieeexplore.ieee.org/document/1035036/.

[29] Girshick, R. Fast R-CNN. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015, s. 1440–1448. DOI: 10.1109/ICCV.2015.169. ISBN 978-1-4673-8391-2. Dostupné z: http://ieeexplore.ieee.org/document/7410526/.

[30] Girshick, R., Donahue, J., Darrell, T. a Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, 2014, s. 580–587. DOI: 10.1109/CVPR.2014.81. ISBN 978-1-4799-5118-5. Dostupné z: http://ieeexplore.ieee.org/document/6909475/.

[31] Grabner, H., Grabner, M. a Bischof, H. Real-Time Tracking via On-line Boosting. In: *Procedings of the British Machine Vision Conference 2006*. British Machine Vision Association, 2006, s. 6.1–6.10. DOI: 10.5244/C.20.6. ISBN 1-901725-32-4. Dostupné z: http://www.bmva.org/bmvc/2006/papers/033.html.

[32] Graves, A., Mohamed, A. rahman a Hinton, G. Speech recognition with deep recurrent neural networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada: IEEE, 2013, s. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947. ISBN 978-1-4799-0356-6. Dostupné z: http://ieeexplore.ieee.org/document/6638947/.

[33] Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.-M. et al. Struck: Structured Output Tracking with Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016-10-1, sv. 38, č. 10, s. 2096–2109. DOI: 10.1109/TPAMI.2015.2509974. ISSN 0162-8828. Dostupné z: http://ieeexplore.ieee.org/document/7360205/.

[34] He, K., Gkioxari, G., Dollar, P. a Girshick, R. Mask R-CNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, 2017, s. 2980–2988. DOI: 10.1109/ICCV.2017.322. ISBN 978-1-5386-1032-9. Dostupné z: http://ieeexplore.ieee.org/document/8237584/.

[35] He, K., Zhang, X., Ren, S. a Sun, J. Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016, s. 770–778. DOI: 10.1109/CVPR.2016.90. ISBN 978-1-4673-8851-1. Dostupné z: http://ieeexplore.ieee.org/document/7780459/.

[36] Held, D., Thrun, S. a Savarese, S. Learning to Track at 100 FPS with Deep Regression Networks. In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, s. 749–765. DOI: 10.1007/978-3-319-46448-0_45. ISBN 978-3-319-46447-3. Dostupné z: http://link.springer.com/10.1007/978-3-319-46448-0_45.

[37] Henriques, J. F., Caseiro, R., Martins, P. a Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2015-3-1, sv. 37, č. 3, s. 583–596. DOI: 10.1109/TPAMI.2014.2345390. ISSN 0162-8828. Dostupné z: http://ieeexplore.ieee.org/document/6870486/.

[38] Huang, L., Zhao, X. a Huang, K. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021, sv. 43, č. 5, s. 1562–1577. DOI: 10.1109/TPAMI.2019.2957464. ISSN 0162-8828. Dostupné z: https://ieeexplore.ieee.org/document/8922619/.

[39] Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. 2007, sv. 9, č. 3, s. 90–95. DOI: 10.1109/MCSE.2007.55. ISSN 1521-9615. Dostupné z: http://ieeexplore.ieee.org/document/4160265/.

[40] Janku, P., Koplik, K., Dulik, T., Szabo, I., Mastorakis, N. et al. Comparison of tracking algorithms implemented in OpenCV. *MATEC Web of Conferences*. 2016, sv. 76, s. 4031. DOI: 10.1051/matecconf/20167604031. ISSN 2261-236X. Dostupné z: http://www.matec-conferences.org/10.1051/matecconf/20167604031.

[41] Kalal, Z., Mikolajczyk, K. a Matas, J. Forward-Backward Error: Automatic Detection of Tracking Failures. In: *2010 20th International Conference on Pattern Recognition*. Istanbul, Turkey: IEEE, 2010, s. 2756–2759. DOI: 10.1109/ICPR.2010.675. ISBN 978-1-4244-7542-1. Dostupné z: http://ieeexplore.ieee.org/document/5596017/.

[42] Kalal, Z., Mikolajczyk, K. a Matas, J. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2012, sv. 34, č. 7, s. 1409–1422. DOI: 10.1109/TPAMI.2011.239. ISSN 0162-8828. Dostupné z: http://ieeexplore.ieee.org/document/6104061/.

[43] Kart, U., Kämäräinen, J.-K., Fan, L. a Gabbouj, M. Evaluation of Visual Object Trackers on Equirectangular Panorama. In: *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018-1-27, s. 25–32. DOI: 10.5220/0006526200250032. ISBN 978-989-758-290-5. Dostupné z: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006526200250032.

[44] Kobilarov, M., Sukhatme, G., Hyams, J. a Batavia, P. People tracking and following with mobile robot using an omnidirectional camera and a laser.

In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* Orlando, FL, USA: IEEE, 2006, s. 557–562. DOI: 10.1109/ROBOT.2006.1641769. ISBN 0-7803-9505-0. Dostupné z: http://ieeexplore.ieee.org/document/1641769/.

[45] Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R. et al. A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2016, sv. 38, č. 11, s. 2137–2155. DOI: 10.1109/TPAMI.2016.2516982. ISSN 0162-8828. Dostupné z: http://ieeexplore.ieee.org/document/7379002/.

[46] Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J. et al. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Long Beach, CA, USA: IEEE, 2019, s. 4277–4286. DOI: 10.1109/CVPR.2019.00441. ISBN 978-1-7281-3293-8. Dostupné z: https://ieeexplore.ieee.org/document/8954116/.

[47] Li, B., Yan, J., Wu, W., Zhu, Z. a Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* Salt Lake City, UT, USA: IEEE, 2018, s. 8971–8980. DOI: 10.1109/CVPR.2018.00935. ISBN 978-1-5386-6420-9. Dostupné z: https://ieeexplore.ieee.org/document/8579033/.

[48] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B. et al. Feature Pyramid Networks for Object Detection. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Honolulu, HI, USA: IEEE, 2017, s. 936–944. DOI: 10.1109/CVPR.2017.106. ISBN 978-1-5386-0457-1. Dostupné z: http://ieeexplore.ieee.org/document/8099589/.

[49] Lin, T.-Y., Goyal, P., Girshick, R., He, K. a Dollar, P. Focal Loss for Dense Object Detection. In: *2017 IEEE International Conference on Computer Vision (ICCV).* Venice, Italy: IEEE, 2017, s. 2999–3007. DOI: 10.1109/ICCV.2017.324. ISBN 978-1-5386-1032-9. Dostupné z: http://ieeexplore.ieee.org/document/8237586/.

[50] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P. et al. Microsoft COCO: Common Objects in Context. In: *Computer Vision – ECCV 2014.* Cham: Springer International Publishing, 2014, s. 740–755. DOI: 10.1007/978-3-319-10602-1_48. ISBN 978-3-319-10601-4. Dostupné z: http://link.springer.com/10.1007/978-3-319-10602-1_48.

[51] Liu, K.-C., Shen, Y.-T. a Chen, L.-G. Simple online and realtime tracking with spherical panoramic camera. In: *2018 IEEE International Conference on Consumer Electronics (ICCE).* Las Vegas, NV, USA: IEEE, 2018, s. 1–6. DOI: 10.1109/ICCE.2018.8326132. ISBN 978-1-5386-3025-9. Dostupné z: http://ieeexplore.ieee.org/document/8326132/.

[52] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. et al. SSD: Single Shot MultiBox Detector. In: *Computer Vision – ECCV 2016.* Cham: Springer International Publishing, 2016, s. 21–37. DOI: 10.1007/978-3-319-46448-0_2. ISBN 978-3-319-46447-3. Dostupné z: http://link.springer.com/10.1007/978-3-319-46448-0_2.

[53] Lucas, B. D. a Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, s. 674–679. IJCAI'81. Dostupné z: https://dl.acm.org/doi/10.5555/1623264.1623280.

[54] Lukežič, A., Vojíř, T., Čehovin Zajc, L., Matas, J. a Kristan, M. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. *International Journal of Computer Vision*. 2018, sv. 126, č. 7, s. 671–688. DOI: 10.1007/s11263-017-1061-3. ISSN 0920-5691. Dostupné z: http://link.springer.com/10.1007/s11263-017-1061-3.

[55] Mi, T.-W. a Yang, M.-T. Comparison of Tracking Techniques on 360-Degree Videos. *Applied Sciences*. 2019, sv. 9, č. 16, s. 3336. DOI: 10.3390/app9163336. ISSN 2076-3417. Dostupné z: https://www.mdpi.com/2076-3417/9/16/3336.

[56] Mituyosi, T., Yagi, Y. a Yachida, M. Real-time human feature acquisition and human tracking by omnidirectional image sensor. In: *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI2003*. Tokyo, Japan: IEEE, 2003, s. 258–263. DOI: 10.1109/MFI-2003.2003.1232667. ISBN 0-7803-7987-X. Dostupné z: http://ieeexplore.ieee.org/document/1232667/.

[57] Müller, M., Bibi, A., Giancola, S., Alsubaihi, S. a Ghanem, B. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In: *Computer Vision – ECCV 2018*. 1. vyd. Cham: Springer International Publishing, 2018, s. 310–327. DOI: 10.1007/978-3-030-01246-5_19. ISBN 978-3-030-01245-8. Dostupné z: http://link.springer.com/10.1007/978-3-030-01246-5_19.

[58] Nalwa, V. *A True Omni-Directional Viewer*. Bell Laboratories, Holmdel, NJ 07733, U.S.A., únor 1996. Dostupné z: http://fullview.com/A_True_Omni-Directional_Viewer.pdf.

[59] Nam, H. a Han, B. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016, s. 4293–4302. DOI: 10.1109/CVPR.2016.465. ISBN 978-1-4673-8851-1. Dostupné z: http://ieeexplore.ieee.org/document/7780834/.

[60] Nasrabadi, A. T., Samiei, A., Mahzari, A., McMahan, R. P., Prakash, R. et al. A taxonomy and dataset for 360° videos. In: *Proceedings of the 10th ACM Multimedia Systems Conference*. New York, NY, USA: ACM, 2019-06-18, s. 273–278. DOI: 10.1145/3304109.3325812. ISBN 978-1-4503-6297-9. Dostupné z: https://dl.acm.org/doi/10.1145/3304109.3325812.

[61] Nayar, S. Catadioptric omnidirectional camera. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Juan, PR, USA: IEEE Comput. Soc, 1997, s. 482–488. DOI: 10.1109/CVPR.1997.609369. ISBN 0-8186-7822-4. Dostupné z: http://ieeexplore.ieee.org/document/609369/.

[62] Nayar, S. a Peri, V. Folded catadioptric cameras. In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No*

PR00149). Fort Collins, CO, USA: IEEE Comput. Soc, 1999, s. 217–223. DOI: 10.1109/CVPR.1999.784632. ISBN 0-7695-0149-4. Dostupné z: http://ieeexplore.ieee.org/document/784632/.

[63] Redmon, J., Divvala, S., Girshick, R. a Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016, s. 779–788. DOI: 10.1109/CVPR.2016.91. ISBN 978-1-4673-8851-1. Dostupné z: http://ieeexplore.ieee.org/document/7780460/.

[64] Redmon, J. a Farhadi, A. YOLO9000: Better, Faster, Stronger. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017, s. 6517–6525. DOI: 10.1109/CVPR.2017.690. ISBN 978-1-5386-0457-1. Dostupné z: http://ieeexplore.ieee.org/document/8100173/.

[65] Redmon, J. a Farhadi, A. *YOLOv3: An Incremental Improvement*. 2018. Dostupné z: https://arxiv.org/abs/1804.02767.

[66] Ren, S., He, K., Girshick, R. a Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017-6-1, sv. 39, č. 6, s. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031. ISSN 0162-8828. Dostupné z: http://ieeexplore.ieee.org/document/7485869/.

[67] Scaramuzza, D. Omnidirectional Camera. In: *Computer Vision*. Boston, MA: Springer US, 2014, s. 552–560. DOI: 10.1007/978-0-387-31439-6_488. ISBN 978-0-387-30771-8. Dostupné z: http://link.springer.com/10.1007/978-0-387-31439-6_488.

[68] Shi, J. a Tomasi. Good features to track. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*. Seattle, WA, USA: IEEE Comput. Soc. Press, 1994, s. 593–600. DOI: 10.1109/CVPR.1994.323794. ISBN 0-8186-5825-8. Dostupné z: http://ieeexplore.ieee.org/document/323794/.

[69] Simonyan, K. a Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. Dostupné z: https://arxiv.org/abs/1409.1556.

[70] Su, Y.-C. a Grauman, K. Flat2Sphere: Learning Spherical Convolution for Fast Features from 360° Imagery. In: *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017, s. 529–539. DOI: 10.5555/3294771.3294822. ISBN 978-1-5108-6096-4. Dostupné z: https://dl.acm.org/doi/abs/10.5555/3294771.3294822.

[71] Tang, Y., Li, Y., Ge, S. S., Luo, J. a Ren, H. Parameterized Distortion-Invariant Feature for Robust Tracking in Omnidirectional Vision. *IEEE Transactions on Automation Science and Engineering*. 2016, sv. 13, č. 2, s. 743–756. DOI: 10.1109/TASE.2015.2392160. ISSN 1545-5955. Dostupné z: http://ieeexplore.ieee.org/document/7046440/.

[72] Tomasi, C. a Kanade, T. *Detection and Tracking of Point Features*. International Journal of Computer Vision, 1991. Dostupné z: https://cecas.clemson.edu/~stb/klt/tomasi-kanade-techreport-1991.pdf.

[73] Vondrick, C., Patterson, D. a Ramanan, D. Efficiently Scaling up Crowdsourced Video Annotation. *International Journal of Computer Vision*. Springer Netherlands. s. 1–21. ISSN 0920-5691. 10.1007/s11263-012-0564-1. Dostupné z: http://dx.doi.org/10.1007/s11263-012-0564-1.

[74] Wang, Q., Zhang, L., Bertinetto, L., Hu, W. a Torr, P. H. Fast Online Object Tracking and Segmentation: A Unifying Approach. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, 2019, s. 1328–1338. DOI: 10.1109/CVPR.2019.00142. ISBN 978-1-7281-3293-8. Dostupné z: https://ieeexplore.ieee.org/document/8953931/.

[75] Wikipedia contributors. *Catadioptric system — Wikipedia, The Free Encyclopedia*. 2020. [Online; accessed 10-April-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Catadioptric_system.

[76] Wikipedia contributors. *Fisheye lens — Wikipedia, The Free Encyclopedia*. 2020. [Online; accessed 14-January-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Fisheye_lens.

[77] Wikipedia contributors. *Video tracking — Wikipedia, The Free Encyclopedia*. 2020. [Online; accessed 9-January-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Video_tracking.

[78] Wikipedia contributors. *Aircraft principal axes — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 10-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Aircraft_principal_axes.

[79] Wikipedia contributors. *Analysis of variance — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 12-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Analysis_of_variance.

[80] Wikipedia contributors. *Augmented reality — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 7-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Augmented_reality.

[81] Wikipedia contributors. *Cartesian coordinate system — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 6-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Cartesian_coordinate_system.

[82] Wikipedia contributors. *Euler angles — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 10-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Euler_angles.

[83] Wikipedia contributors. *Omnidirectional (360-degree) camera — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 13-April-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Omnidirectional_(360-degree)_camera.

[84] Wikipedia contributors. *Polar coordinate system — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 6-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Polar_coordinate_system.

[85] Wikipedia contributors. *Spherical coordinate system — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 6-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Spherical_coordinate_system.

[86] Wikipedia contributors. *Virtual reality — Wikipedia, The Free Encyclopedia*. 2021. [Online; accessed 6-May-2021]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Virtual_reality.

[87] Wojke, N., Bewley, A. a Paulus, D. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE International Conference on Image Processing (ICIP)*. Beijing, China: IEEE, 2017, s. 3645–3649. DOI: 10.1109/ICIP.2017.8296962. ISBN 978-1-5090-2175-8. Dostupné z: http://ieeexplore.ieee.org/document/8296962/.

[88] Wu, Y., Lim, J. a Yang, M.-H. Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2015-9-1, sv. 37, č. 9, s. 1834–1848. DOI: 10.1109/TPAMI.2014.2388226. ISSN 0162-8828. Dostupné z: http://ieeexplore.ieee.org/document/7001050/.

[89] Yagi, Y. a Kawato, S. Panorama scene analysis with conic projection. In: *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*. Ibaraki, Japan: IEEE, 1990, s. 181–187. DOI: 10.1109/IROS.1990.262385. Dostupné z: http://ieeexplore.ieee.org/document/262385/.

[90] Yang, F., Li, F., Wu, Y., Sakti, S. a Nakamura, S. *Using Panoramic Videos for Multi-person Localization and Tracking in a 3D Panoramic Coordinate*. 2020. Dostupné z: https://arxiv.org/abs/1911.10535.

[91] Yang, W., Qian, Y., Kamarainen, J.-K., Cricri, F. a Fan, L. Object Detection in Equirectangular Panorama. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. Beijing, China: IEEE, 2018, s. 2190–2195. DOI: 10.1109/ICPR.2018.8546070. ISBN 978-1-5386-3788-3. Dostupné z: https://ieeexplore.ieee.org/document/8546070/.

[92] Zajc, L. C., Lukezic, A., Leonardis, A. a Kristan, M. Beyond Standard Benchmarks: Parameterizing Performance Evaluation in Visual Object Tracking. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, 2017, s. 3343–3351. DOI: 10.1109/ICCV.2017.360. ISBN 978-1-5386-1032-9. Dostupné z: http://ieeexplore.ieee.org/document/8237622/.

[93] Zha, Y., Wu, M., Qiu, Z., Dong, S., Yang, F. et al. Distractor-Aware Visual Tracking by Online Siamese Network. *IEEE Access*. 2019, sv. 7, s. 89777–89788. DOI: 10.1109/ACCESS.2019.2927211. ISSN 2169-3536. Dostupné z: https://ieeexplore.ieee.org/document/8756110/.

[94] Zhang, Y., Xiao, X. a Yang, X. Real-Time Object Detection for 360-Degree Panoramic Image Using CNN. In: *2017 International Conference on Virtual Reality and Visualization (ICVRV)*. Zhengzhou, China, China: IEEE, 2017, s. 18–23. DOI: 10.1109/ICVRV.2017.00013. ISBN 978-1-5386-2636-8. Dostupné z: https://ieeexplore.ieee.org/document/8719156/.

[95] Zhang, Z. a Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, 2019, s. 4586–4595. DOI: 10.1109/CVPR.2019.00472. ISBN 978-1-7281-3293-8. Dostupné z: https://ieeexplore.ieee.org/document/8953458/.

[96] Zhang, Z., Peng, H., Fu, J., Li, B. a Hu, W. Ocean: Object-Aware Anchor-Free Tracking. In: *Computer Vision – ECCV 2020*. 2020. vyd. Cham: Springer International Publishing, 2020, s. 771–787. DOI: 10.1007/978-3-030-58589-1_46. ISBN 978-3-030-58588-4. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-030-58589-1_46.

[97] Zhou, Z., Niu, B., Ke, C. a Wu, W. Static Object Tracking in Road Panoramic Videos. In: *2010 IEEE International Symposium on Multimedia*. Taichung, Taiwan: IEEE, 2010, s. 57–64. DOI: 10.1109/ISM.2010.18. ISBN 978-1-4244-8672-4. Dostupné z: http://ieeexplore.ieee.org/document/5693823/.

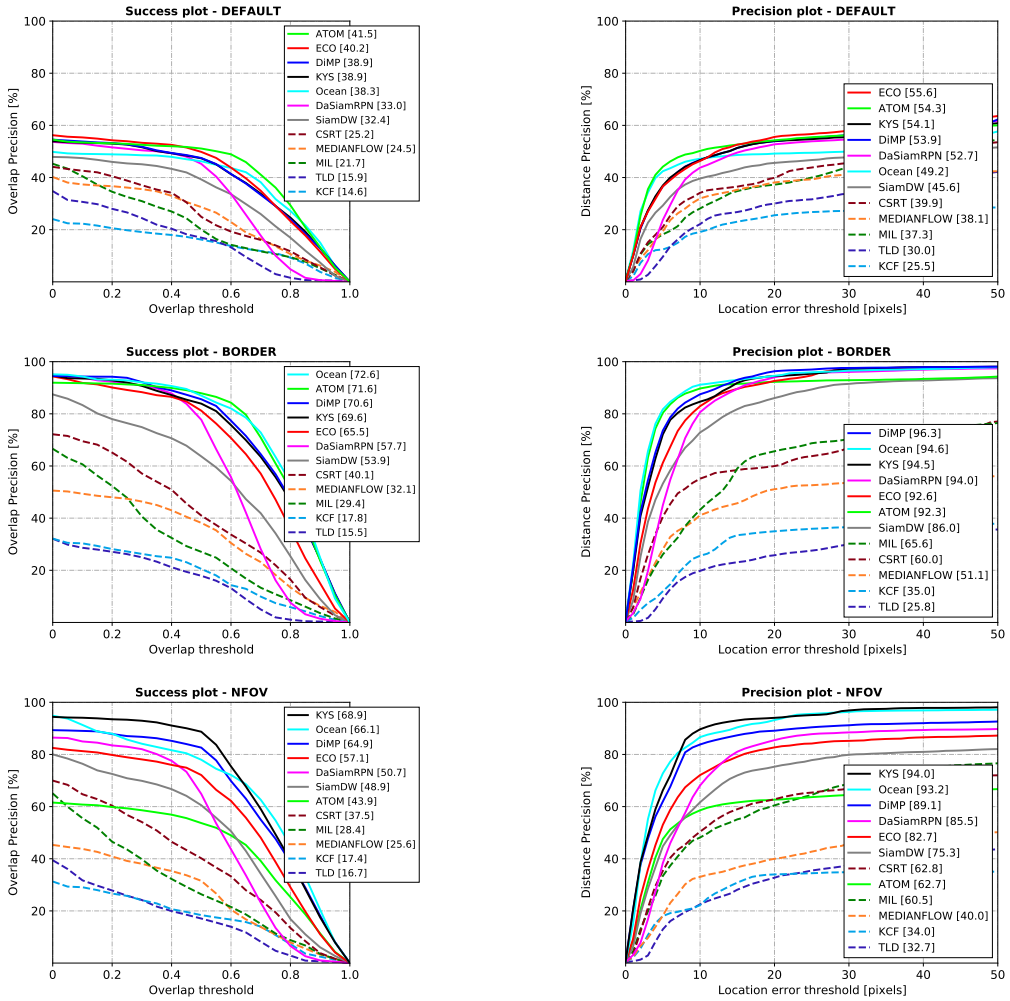
Příloha A

Obsah přiloženého DVD

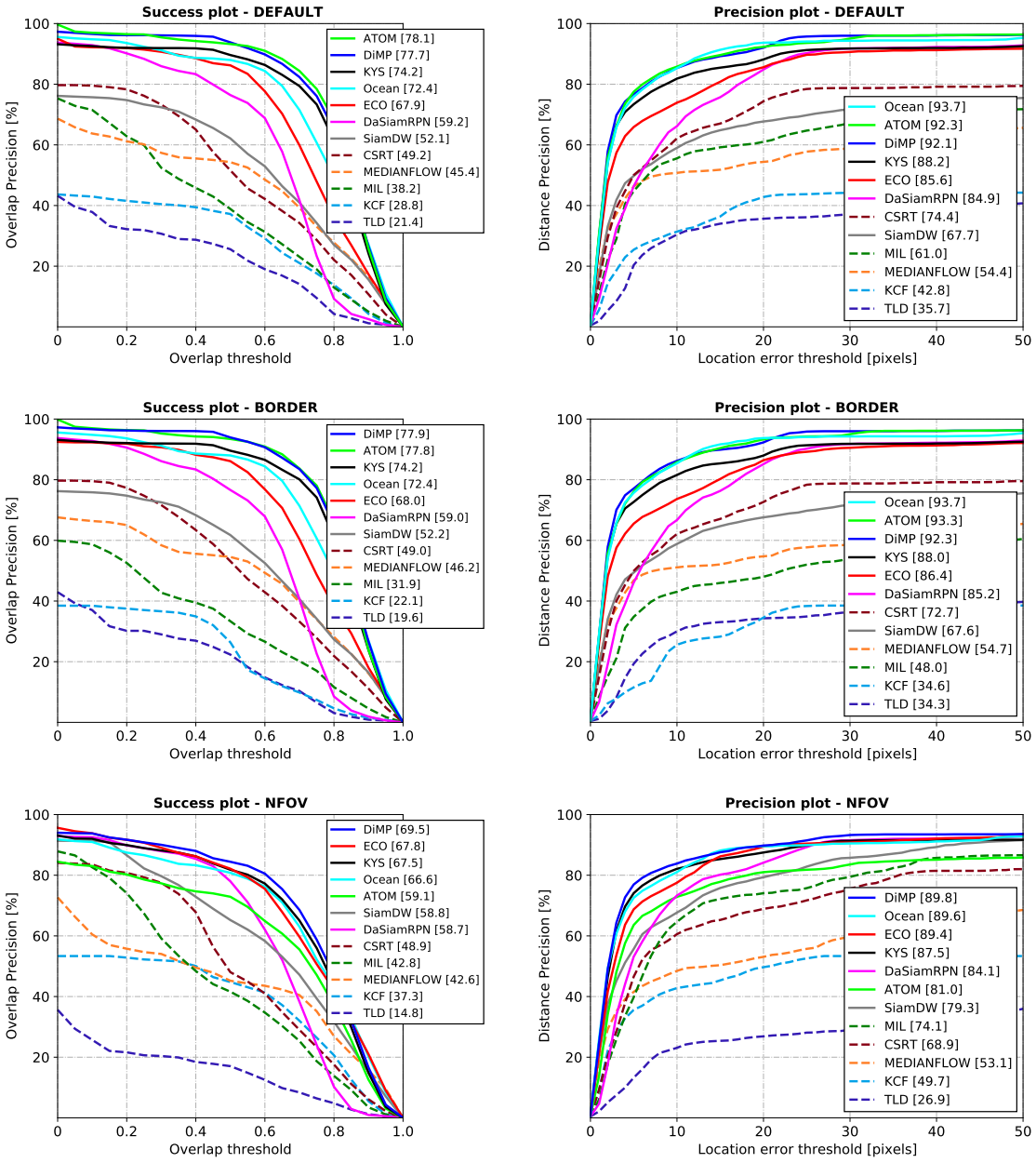
- **code/** - adresář obsahující dataset 360° videí včetně anotací a veškeré zdrojové kódy ve formě Python balíku
- **tech_report/** - adresář obsahující PDF této technické zprávy a zdrojové kódy potřebné pro její vytvoření
- **poster.pdf** - plakát pro stručnou prezentaci této diplomové práce
- **install.sh** - skript umožňující instalaci všech potřebných balíků či knihoven
- **README.md** - soubor obsahující manuál pro instalaci a spuštění

Příloha B

Dodatečné grafy



Obrázek B.1: *Success* a *Precision* grafy pro videosekvence, kde objekt prochází mezi okraji ekvirektangulárních snímků. Popisek *DEFAULT* představuje výsledky trackerů bez modifikace, *BORDER* výsledky trackerů s vylepšením sférické rotace a *NFOV* výsledky trackerů v rektilineární projekci. Jedná se celkem o 13 videosekvencí v představeném datasetu (01, 02, 03, 04, 08, 11, 12, 13, 14, 15, 16, 18, 21).



Obrázek B.2: *Success* a *Precision* grafy pro videosekvence, kde objekt neprochází mezi okraji ekvirektangulárních snímků. Popisek *DEFAULT* představuje výsledky trackerů bez modifikace, *BORDER* výsledky trackerů s vylepšením sférické rotace a *NFOV* výsledky trackerů v rektilineární projekci. Jedná se celkem o 7 videosekvencí v představeném datasetu (05, 06, 07, 09, 10, 17, 19, 20).

Příloha C

Výsledky videosekvencí

video	MIL [2]	MF [41]	TLD [42]	KCF [37]	CSR [54]	ECO [17]	ATO [18]	DSR [93]	SDW [95]	DMP [5]	KYS [6]	OCE [96]	∅
01	20.8	23.3	36.2	18.5	35.8	65.2	36.2	42.8	57.0	47.2	50.6	51.0	40.4
02	10.5	10.2	0.3	5.2	12.8	72.1	80.6	54.9	57.4	80.7	80.5	71.5	44.7
03	14.9	44.0	0.6	2.9	28.8	51.7	53.5	56.4	24.6	55.1	56.2	58.8	37.3
04	73.3	39.0	14.7	5.6	61.5	74.8	81.2	62.2	65.3	51.3	51.2	74.0	54.5
05	0.5	34.1	4.6	0.2	62.7	66.8	73.7	69.6	67.6	74.2	74.2	69.2	49.8
06	16.0	30.0	1.9	14.2	54.2	66.6	82.2	43.7	19.8	80.6	80.8	57.5	45.6
07	41.3	74.1	29.4	22.6	59.7	79.5	85.7	73.7	67.5	82.8	83.5	79.8	65.0
08	16.4	59.9	10.0	18.8	49.0	56.3	67.9	63.9	17.2	68.8	50.5	56.9	44.6
09	17.4	35.5	25.2	3.3	26.3	76.6	82.8	55.2	25.0	82.6	82.7	76.8	49.1
10	50.4	45.3	15.8	61.1	48.6	73.5	69.1	61.1	65.9	68.1	67.3	83.1	59.1
11	27.2	81.1	64.8	58.4	79.1	80.7	65.6	53.3	81.8	59.4	59.8	81.2	66.0
12	56.4	28.1	21.7	57.9	29.0	76.4	82.3	64.3	69.7	81.5	81.9	81.6	60.9
13	15.8	16.1	11.2	12.5	49.4	52.1	48.8	37.2	40.8	49.0	49.5	52.7	36.3
14	28.4	14.9	7.1	3.3	12.0	34.1	81.5	54.9	27.9	85.3	84.5	80.4	42.9
15	11.8	19.9	9.9	5.1	45.6	48.4	81.1	61.7	44.5	82.9	84.4	72.3	47.3
16	44.8	3.0	1.9	0.9	3.2	66.5	76.0	61.9	63.9	77.4	76.2	85.2	46.7
17	16.8	27.4	14.5	14.5	20.1	49.1	59.7	52.3	39.4	57.7	58.3	58.9	39.0
18	57.0	74.8	15.8	40.7	81.2	89.8	90.1	70.8	83.4	90.1	89.9	88.8	72.7
19	64.8	65.3	19.1	11.5	65.1	73.2	83.3	70.9	74.3	89.5	88.8	88.4	66.2
20	48.3	58.1	45.9	49.6	55.2	58.5	86.2	45.2	58.0	88.0	58.4	65.5	59.7
21	5.2	3.0	7.4	1.7	33.3	83.0	86.6	65.7	66.8	89.1	89.1	89.1	51.7
∅	30.4	37.5	17.1	19.5	43.5	66.4	74.0	58.2	53.2	73.4	71.3	72.5	x

Tabulka C.1: Tabulka hodnot metriky *AUC (Area Under Curve)* pro jednotlivé sekvence datasetu. Tyto hodnoty odpovídají prvnímu vylepšení trackerů pomocí sférické rotace ekvirektangulárního snímku (*BORDER*). Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96].

video	MIL [2]	MF [41]	TLD [42]	KCF [37]	CSR [54]	ECO [17]	ATO [18]	DSR [93]	SDW [95]	DMP [5]	KYS [6]	OCE [96]	∅
01	11.0	26.9	35.7	9.5	26.9	56.5	46.3	42.2	57.9	47.4	47.6	50.3	38.2
02	13.9	0.2	0.2	6.1	14.0	71.6	6.9	55.8	54.3	74.0	73.5	77.5	37.3
03	27.0	19.8	1.2	3.3	20.0	14.4	7.0	28.2	24.8	9.5	51.8	57.3	22.0
04	69.4	39.5	41.2	12.6	64.5	74.4	73.7	62.1	66.4	71.3	71.5	73.9	60.0
05	43.5	31.7	8.9	20.1	54.5	73.0	72.0	66.7	66.1	73.8	74.0	68.3	54.3
06	29.6	20.7	2.3	16.2	51.8	69.4	66.3	50.7	41.4	66.9	63.2	51.9	44.2
07	41.7	80.5	14.2	26.0	58.8	77.3	4.1	71.4	67.3	76.7	76.6	79.6	56.2
08	17.3	55.6	11.1	17.9	46.6	50.9	54.9	64.1	47.1	52.2	62.6	56.2	44.7
09	36.9	5.1	5.1	3.3	29.5	53.1	49.5	47.6	50.3	56.7	56.8	58.7	37.7
10	46.6	40.3	19.5	61.1	47.1	72.0	73.4	60.8	67.6	71.6	71.8	77.1	59.0
11	72.7	63.8	71.6	73.6	74.3	74.5	67.6	48.3	69.9	54.4	54.9	79.7	67.1
12	14.4	16.0	10.2	40.6	47.7	70.3	78.9	52.6	30.7	78.5	78.7	74.4	49.4
13	23.0	19.7	14.0	12.5	48.4	52.2	51.0	35.6	31.8	51.4	51.4	47.6	36.5
14	25.5	12.0	8.9	3.2	13.3	34.2	4.3	47.1	26.2	79.6	78.3	37.9	30.9
15	31.0	3.0	1.3	3.8	36.8	10.9	4.1	54.3	13.9	78.1	79.0	43.3	30.0
16	3.1	1.3	3.6	0.9	3.6	62.6	1.6	61.9	62.1	66.7	66.5	85.1	34.9
17	17.9	17.8	14.4	15.3	24.5	53.2	56.0	53.4	33.5	55.5	54.9	49.6	37.2
18	57.0	72.5	13.9	40.0	81.6	89.0	86.9	72.4	83.6	91.7	91.6	88.6	72.4
19	72.7	85.9	7.8	79.6	72.8	87.5	85.1	72.5	86.6	86.7	86.5	87.6	75.9
20	53.9	58.6	45.7	76.9	52.3	57.3	66.6	46.2	57.6	68.4	56.1	60.2	58.3
21	4.3	3.1	4.5	1.9	9.3	80.4	87.8	34.7	67.5	88.8	88.3	87.9	46.5
∅	33.9	32.1	16.0	25.0	41.8	61.2	49.7	53.7	52.7	66.7	68.4	66.3	x

Tabulka C.2: Tabulka hodnot metriky *AUC (Area Under Curve)* pro jednotlivé sekvence datasetu. Tyto hodnoty odpovídají druhému vylepšení, které využívá pro proces sledování rektilineární projekci (*NFOV*). Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96].

video	MIL [2]	MF [41]	TLD [42]	KCF [37]	CSR [54]	ECO [17]	ATO [18]	DSR [93]	SDW [95]	DMP [5]	KYS [6]	OCE [96]	∅
01	59.2	51.5	65.3	45.5	61.0	96.2	58.3	78.0	90.2	92.9	82.5	89.7	72.5
02	39.4	21.3	0.2	6.3	35.6	97.9	99.4	100	86.0	98.5	98.2	84.6	63.9
03	25.8	94.1	0.9	4.5	30.6	80.1	77.7	100	41.8	95.5	80.7	86.4	(59.8
04	100	69.7	29.6	6.2	100	100	100	100	100	99.3	99.6	100	83.7
05	2.2	42.3	11.9	0.6	100	100	100	100	100	100	100	100	71.4
06	42.1	40.0	2.1	33.8	100	91.0	100	92.4	32.4	100	100	95.9	69.1
07	100	100	46.3	34.8	100	100	100	100	100	100	100	100	90.1
08	100	100	28.5	100	100	100	100	100	87.0	100	100	100	93.0
09	31.1	44.6	78.4	4.1	37.8	100	100	100	37.8	100	100	100	69.5
10	34.3	35.8	21.0	75.7	61.4	91.4	82.5	69.6	81.1	71.7	69.4	100	66.2
11	100	100	100	100	100	100	100	100	100	100	100	100	100
12	97.3	53.9	10.3	97.9	4.4	100	100	98.1	96.7	100	100	100	79.9
13	35.6	22.2	21.9	26.6	62.5	59.1	64.1	61.6	63.8	65.9	67.2	69.1	51.6
14	80.7	17.8	11.1	3.7	23.0	80.7	100	99.3	74.8	100	100	100	65.9
15	23.8	26.2	16.7	6.0	100	91.7	100	97.6	100	100	100	100	71.8
16	84.6	3.8	3.1	1.1	4.7	100	100	92.0	85.3	100	100	100	64.6
17	15.7	21.6	11.1	15.7	23.6	50.7	67.6	54.5	30.0	66.8	67.9	68.2	41.1
18	100	100	33.6	54.4	100	100	100	100	100	100	100	100	90.7
19	92.2	87.1	35.6	14.9	92.0	91.8	96.9	98.2	92.9	100	100	100	83.5
20	66.5	66.5	68.4	96.8	66.5	66.5	99.4	66.5	66.5	100	66.5	85.8	76.3
21	6.9	3.7	13.7	2.3	57.6	98.6	100	95.6	93.0	100	100	100	64.3
∅	58.9	52.5	29.0	34.8	64.8	90.3	92.7	90.6	79.0	94.8	92.0	94.3	x

Tabulka C.3: Tabulka hodnot metriky *AUC (Area Under Curve)* pro jednotlivé sekvence datasetu. Tyto hodnoty odpovídají prvnímu vylepšení trackerů pomocí sférické rotace ekvirektangulárního snímku (*BORDER*). Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96].

video	MIL [2]	MF [41]	TLD [42]	KCF [37]	CSR [54]	ECO [17]	ATO [18]	DSR [93]	SDW [95]	DMP [5]	KYS [6]	OCE [96]	∅
01	23.7	45.1	71.6	32.7	57.5	84.9	91.7	87.0	91.4	81.6	81.6	88.3	69.8
02	50.2	1.9	0.2	7.3	35.6	99.0	20.5	100	79.2	94.5	95.0	94.8	56.5
03	71.2	29.7	2.4	3.6	35.3	20.5	13.1	46.6	42.4	15.7	79.5	81.9	36.8
04	100	69.7	99.8	14.9	100	100	100	100	100	100	100	100	90.4
05	100	41.9	16.7	29.6	100	100	100	100	100	100	100	100	82.4
06	72.4	40.0	12.4	35.2	96.6	100	100	96.6	79.3	99.3	99.3	93.1	77.0
07	100	100	37.3	39.3	100	100	18.4	100	100	100	100	100	82.9
08	99.8	100	58.5	100	100	100	100	100	99.8	99.8	100	100	96.5
09	100	28.4	28.4	4.1	48.6	100	100	100	100	100	100	100	75.8
10	31.8	30.9	23.1	70.6	7.7	97.0	85.7	51.2	64.2	72.2	69.8	100	58.7
11	100	100	100	100	100	100	100	100	100	100	100	100	100
12	9.4	14.0	2.8	93.5	79.2	97.5	99.9	90.3	54.9	99.7	99.8	92.3	69.5
13	52.8	26.9	24.7	26.9	67.2	68.1	69.4	55.0	37.8	66.9	66.6	62.2	52.0
14	80.0	14.8	13.3	3.7	23.7	83.0	6.7	97.8	67.4	100	100	91.9	56.9
15	100	11.9	2.4	4.8	100	23.8	10.7	91.7	26.2	100	100	100	56.0
16	4.0	2.2	18.3	1.1	4.7	100	2.7	92.2	85.3	100	100	100	50.9
17	22.7	16.9	16.0	22.7	31.8	51.9	63.3	58.9	24.8	66.5	64.4	56.9	41.4
18	88.5	100	22.6	51.0	100	100	100	100	100	100	100	100	88.5
19	99.1	100	14.0	100	100	100	100	100	100	100	100	100	92.8
20	66.5	66.5	67.0	95.9	66.5	66.5	80.7	66.5	66.5	80.3	66.5	66.5	71.3
21	6.5	3.3	9.1	2.3	13.4	98.2	100	50.8	94.4	100	100	100	56.5
∅	65.6	45.0	30.5	40.0	65.1	85.3	69.6	85.0	76.8	89.4	91.5	91.8	x

Tabulka C.4: Tabulka hodnot metriky *AUC (Area Under Curve)* pro jednotlivé sekvence datasetu. Tyto hodnoty odpovídají druhému vylepšení, které využívá pro proces sledování rektilineární projekci (*NFOV*). Jedná se o vyhodnocení trackerů MIL [2], MEDIANFLOW [41], TLD [42], KCF [37], CSRT [54], ECO [17], ATOM [18], DaSiamRPN [93], SiamDW [95], DiMP [5], KYS [6] a Ocean [96].