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

SPRACOVANIE RADAROVÉHO SIGNÁLU A FÚZIA S OBRAZOM

RADAR SIGNAL PROCESSING AND FUSION WITH IMAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JURAJ KULA
AUTHOR

VEDOUCÍ PRÁCE prof. Dr. Ing. PAVEL ZEMČÍK,
SUPERVISOR

BRNO 2019

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

12616

Student: **Kula Juraj**
Program: Informační technologie
Název: **Zpracování radarového signálu a fúze s obrazem**
Radar Signal Processing and Fusion with Image
Kategorie: Zpracování obrazu
Zadání:

1. Nastudujte literaturu a existující algoritmy zpracování radarového signálu se zaměřením na Dopplerovy radary se 2D směrovou anténou. Nastudujte též existující postupy slučování senzorové informace radarů a kamer.
2. Vyberte vhodnou metodu spojování informací z radaru a kamery a vyhodnoťte dosažitelné vlastnosti algoritmu spojení.
3. Navrhněte postup implementace a implementační platformu a zvažte vlastnosti dosažitelné navrženým řešením.
4. Implementujte spojování radarové a kamerové informace a demonstrujte výsledek.
5. Vyhodnoťte dosažené výsledky a možnosti dalšího pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 6. listopadu 2018

Abstrakt

Cieľom tejto bakalárskej práce bolo vytvoriť systém pre spočítanie ľudí v miestnosti a zobrazení informácií o nich vo videozázname. Systém využíva fúziu dát získaných zo senzorov kamera a radar. Užívateľom by mal pomôcť pri zisťovaní obsadenosti miestností, alebo pri získavaní informácii o pohybe chodcov na frekventovaných úsekoch. Výsledkom je systém implementovaný v jazyku Python využívajúci voľne dostupné knižnice.

Abstract

The goal of this bachelor thesis was to create a system for counting people in the room and visualizing information about them in the video. The system uses the fusion of data obtained from camera and radar sensors. Users should be able to help in determining the occupancy of the room or obtaining information on pedestrian traffic on frequented streets. Final result is the system implemented in Python language using open source libraries.

Kľúčové slová

Spracovanie obrazu, Spracovanie radarového signálu, Fúzia dát, Detekcia ľudí, Detekcia objektov, Klasifikácia objektov, Sledovanie objektov

Keywords

Image processing, Radar signal processing, Data fusion, Pedestrian detection, Object detection, Object Classification, Object tracking

Citácia

KULA, Juraj. *Spracovanie radarového signálu a fúzia s obrazom.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík,

Spracovanie radarového signálu a fúzia s obrazom

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Prof. Dr. Ing. Pavla Zemčíka. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Juraj Kula
15. mája 2019

Poďakovanie

Veľmi rád by som sa tu poďakoval svojmu vedúcemu práce pánovi Prof. Dr. Ing. Pavlovi Zemčíkovi za ochotu pri rozbore témy, čas venovaný konzultáciám a ústretový prístup pri vedení mojej bakalárskej práce. Tiež by som sa chcel poďakovať pánovi Ing. Lukášovi Maršíkovi za čas venovaný konzultáciám pri riešení tejto práce, poskytnuté informácie a testovacie dáta z kamery a radaru.

Obsah

1 Úvod **5**

2 Zhrnutie doterajšieho stavu **7**
2.1 Dopplerov radar 7
2.2 Detekcia objektov v obraze 23
2.3 Fúzia senzorov 30
2.4 Prehľad existujúcich riešení 33

3 Analýza a návrh riešenia **35**
3.1 Analýza konkrétneho problému 35
3.2 Zhodnotenie existujúcich systémov 35
3.3 Návrh riešenia problému 36
3.4 Technická špecifikácia 40

4 Implementácia **41**
4.1 Logické celky 41
4.2 Rozvrhnutie modulov systému 46
4.3 Vyhodnotenie funkčnosti systému 47
4.4 Testovanie a experimenty 47

5 Záver **50**

Literatúra **52**

Zoznam obrázkov

2.1 Doska IRW1642Boost s mmWave senzorom IRW1642[3] 8
2.2 Princíp radaru 10
2.3 Dopplerov jav 12
2.4 Rozdielne bunky posuvného okna algoritmu CFAR[27] 15
2.5 Všeobecná logika algoritmu MHT 18
2.6 Stromová štruktúra algoritmu MHT 18
2.7 Graf trás algoritmu MWIS 21
2.8 Stacionárny (a) a Skrytý (b) Markovský model 22
2.9 Všeobecná schéma detekcie v obraze 23
2.10 Princíp detekcie objektov 24
2.11 SIFT deskriptor 25
2.12 Detekcie objektov v obraze na základe šablóny 26
2.13 Princíp algoritmu Background subtraction [10] 27
2.14 Princíp a rozdelenie sledovacích metód [18] 29
2.15 Všeobecný prehľad fúznych architektúr 30
2.16 Koncept fúzie senzorov typu kamera a radar.[36] 32
2.17 Koncept frameworku pre fúziu kamery s radarom.[36] 32

3.1 Blokové schéma systému - návrh 37
3.2 Ilustrácia detekcie ľudí v obraze 38
3.3 Ilustrácia detekcie ľudí v obraze po vykonaní fúzie 40

4.2 Ilustrácia uloženia radarových dát 42
4.3 Non-maxima suppresion 43
4.4 Ilustrácia detekčnej mapy 44
4.5 Implementácia - rozvrhnutie modulov a ostatných častí systému 46
4.1 Implementačná schéma systému 49

Slovník

ADAS	Pokročilé systémy navrhnuté pre asistenciu šoféra (z en. Advanced driver-assistance systems)	33, 35, 36
ADT	Automatická detekcia a sledovanie (z en. Automatic detection and tracking)	9
CIR	Impulzná odozva kanáli (z en. channel impulse response)	15
COV	Deskriptor prvkov používaný pri detekcii objektov v obraze (z en. region covariance descriptor)	28
CW	Vlnový radar (z en. Continuius-wave radar)	8
EKF	Nelienárna vezia Kalmanovho filtru(z en. Extended Kalman filter)	21
FM-CW	Frekvenčno-vlnový radar (z en. Frequency-modulated continuius-wave radar)	8
HOG	Deskriptor prvkov používaný v spracovaní obrazu pri detekcii objektov (z en. Histogram of Oriented Gradients)	25, 28, 43
ISAR	Radar s inverznou syntetickou apertúrou (z en. inverse synthetic aperture radar)	9, 13
MHT	Algoritmus vhodný pre riešenie problému združovania sa objektov (z en. Multiple Hypothesis Tracking)	2, 17–19, 21
MTI	Radar detekujúci pohyblivé ciele (z en. Moving target indication)	8
MWIS	Jedná sa o problém nastávajúci pri optimalizácii kombinatoriky(z en. Maximum weight independent set)	2, 20, 21
NRL	Výskumný ústav námorníctva Spojených Štátov (z en. U.S. Naval Research Laboratory)	7

PRF	Opakovacia frekvencia impulzov (z en. Pulse repetition frequency)	8
ROI	Výber vzorkov z datovej sady určených pre konkrétny účel (z en. Region of interest)	23
SAR	Radar so syntetickou apertúrou (z en. synthetic aperture radar)	9, 13
SIFT	Algoritmus pre detekciu a popis prvkov v snímke. Používa sa v softvéroch pre spracovávanie obrazu (z en. Scale-Invariant Feature Transform)	2, 24, 25
SLAR	Palubný radar (z en. Sidelooking airborne radar)	9
STT	Sledovač jedného cieľa (z en. Single target tracker)	9
SVM	Jedna z metód pre vytvorenie diskriminačného klasifikátora pri detekcii objektov (z en. Support Vector Machines)	27, 28
TO-MHT	Algoritmus pre sledovanie objektov(z en. Track-Oriented Multiple Hypothesis Tracking)	18
TPR	Hodnota správne detekovaných objektov (z en. True positive Rate)	28
TWS	Sledovanie počas skenovania (z en. Track-shile-scan)	9

Kapitola 1

Úvod

Za posledné roky je čoraz vyšší záujem využitia viacerých senzorov súčasne a skombinovať ich dáta. Senzory použité samostatne majú obmedzené využitie, pretože nedokážu zaznamenávať všetko. Dokážu len to, na čo sú stvorené. Napríklad zistenie vzdialenosti objektu od senzora nie je vhodná úloha pre kameru, pretože danú vzdialenosť nedokáže zmerať presne. Pre klasifikáciu objektov pomocou ich rozpoznávania je však ideálna. Oproti tomu radarové senzory nemajú takú vysokú rozpoznávaciu schopnosť, avšak dokážu poskytnúť presnejšie informácie o polohe objektu, alebo jeho vzdialenosti. Samostatne tieto systémy majú nedostatky, avšak pomocou fúzie je možné spojiť ich funkcionalitu a výsledný systém nakoniec používa to najlepšie z oboch senzorov. Na podobnom princípe vníma svet každý živý organizmus.

Fúzia senzorov je stále vo vývoji. Väčšina popredných automobilových spoločností vyvíja vlastné systémy využívajúce fúziu viacerých senzorov. Zväčša sa jedná o bezpečnostné systémy. Fúzia však nemusí byť použitá len v automobilovej doprave, a preto som sa rozhodol spraviť systém fungujúci na fúzii zo senzorov typu radar a kamera. Ten by umožnil spočítať ľudí v miestnosti a do videozáznamu vložiť dodatočné informácie k nim zistené, ako je ich vzdialenosť alebo rýchlosť.

V dnešnej dobe sa často stáva, že niekam ideme, napríklad k lekárovi a nevieme, koľko čakajúcich ľudí sa tam vyskytuje. Taktiež na vysokých školách je ťažké odhadnúť počet ľudí vyskytujúcich sa na prednáške. Aj cestná premávka môže byť sledovaná podobným systémom. Veľakrát sa stane nehoda, pri ktorej sa len odhaduje, kto ju spôsobil. Na rizikových úsekoch by bolo možné mať takýto systém. Sledoval by cestnú premávku a na videozázname by boli informácie o rýchlosti pohybujúcich sa vozidiel a ich vzdialenosť od systému. To by mohlo pomôcť pri vyšetrovaní nehody. Fúzia v technológiách je stále vo vývoji, aj keď každý rok sa tým spoločnosti začínajú zaoberať stále viac a viac. To má za následok, že aj takéto systémy budú v budúcnosti možné.

Úlohou tejto práce je teda preštudovať oblasť, ktorá sa týka spracovania radarového signálu, spracovania obrazu, detekcie objektov v dátach získaných zo senzorov a spôsoby fúzie spracovaných dát. Vďaka získaným informáciám je potrebné zvoliť a použiť vhodné knižnice a algoritmy pre prácu so senzormi, ako aj navrhnúť spôsob ako výstupné dáta spojiť a vizualizovať ich. Následne je potrebné navrhnúť systém schopný detekcie ľudí v miestnosti pre zistenie počtu osôb. Túto tému som si vybral, pretože ma zaujíma spracovanie obrazu, ako aj samotná fúzia z viacerých senzorov. Podľa môjho názoru bude mať táto téma v budúcnosti čoraz väčší význam.

V kapitole 2 je uvedený súhrn aktuálneho stavu technológii v oblasti spracovania radarového signálu, spracovania obrazu a fúzie viacerých senzorov.

V kapitole 3 je popísaný návrh systému, porovnanie môjho systému s už existujúcim podobným riešením, ako aj technické parametre systému.

V poslednej kapitole 4 je popis implementácie na základe návrhu, rozvrhnutie do modulov, vyhodnotenie funkčnosti systému a popis testovania.

Kapitola 2

Zhrnutie doterajšieho stavu

Na začiatku tejto kapitoly je vysvetlený senzor radar. Stručná história, typy radarov, jeho funkčnosť, ako aj spôsoby detekcie a sledovania objektov. Za ním nasleduje časť týkajúca sa detekcie objektov v obraze. Ku koncu sú spísané spôsoby fúzie z týchto dvoch zariadení, ako aj už existujúce systémy riešiacich rovnaký alebo podobný problém ako je fúzia odlišných senzorov.

Jednotlivé spôsoby a algoritmy tu popísané sú len časťou možností ako riešiť súvisiace problémy a teda sú len úvodom do jednotlivých oblastí.

2.1 Dopplerov radar

Radar je elektromagnetické zariadenie pre detekciu, lokáciu, sledovanie a rozpoznávanie objektov. Funguje na princípe vysielania elektromagnetických vĺn smerom k cieľovým objektom a spracovaním ozveny z týchto detekovaných objektov. Objektami môžu byť napríklad lietadlá, lode, automobilové zariadenia, vesmírne telesá, dokonca ľudia, vtáky alebo dážď.[33, 31] Radar samotný síce je elektronické zariadenie, avšak má silný vzťah k informačným technológiám.

Cieľom tejto kapitoly je uviesť fakty o radarových systémoch a spracovaním ich signálov v súvislosti s touto bakalárskou prácou a teda sa jedná len o výňatok z celkovej teoretickej časti týkajúcej sa radarov.

Vývoj radaru

Vývoj radaru bol rozšírený v 30-tych rokoch 20. storočia, ale technológia samotná vychádza z experimentov nemeckého fyzika Heinricha Hertza, ktorý experimentoval s Maxwellovými teóriami v roku 1886. [33] V roku 1904 nemecký inžinier Chrisitan Hülsmeier získal patent pre zariadenie schopné detekcie prekážok a lodnej navigácie. [15]

V roku 1922 bola detekovaná prvá drevená loď inžiniermi z NRL a v roku 1930 prvé lietadlo.[33] Jedným z najslávnejších radarových systémov tej doby bol pravdepodobne britský radarový systém včasného varovania, nazývaný „Chain Home“, slúžiaci od roku 1938 do konca 2. svetovej vojny.[33]

Po vojne výskum radaru pokračoval a v súčasnosti sa okrem vojenských účelov používa napríklad pri riadení letovej prevádzky, námornej a leteckej navigácie alebo pri meteorologickom pozorovaní. [33]

Typy radarov

Neexistuje jednoduchý spôsob, ako môžme odlíšiť jeden radar od druhého. Radary odlišujeme na základe ich použitia, vlastností alebo na akom frekvenčnom pásme pracujú.

Obr. 2.1: Doska IRW1642Boost s mmWave senzorom IRW1642[3]

Radary podľa jednoznačnej vlastnosti:[31]

- **Pulzný radar** - opakovane vysiela takmer obdĺžnikové impulzy. Základný radar, ktorý si môžme predstaviť bez znalostí o iných typoch radarov.
- **Radar s vysokou rozlišovacou schopnosťou** - vysoká rozlišovacia schopnosť môže byť dosiahnutá vo vzdialenosti, uhloch, alebo rýchlosti. Zvyčajne sa však jedná o vzdialenosť, ktorú dokáže zmerať ako v metroch, tak aj v centimetroch.
- **Pulzný kompresný radar** - využíva dlhý impulz s vnútornou moduláciou pre získanie výkonu dlhého impulzu s rozlišovacou schopnosťou krátkeho impulzu.
- **CW** - vysiela spojitú sínusovú vlnu. Pre detekciu pohybujúcich sa objektov alebo zmeranie rýchlosti objektu využíva Dopplerov posun.
- **FM-CW** - meria vzdialenosť s frekvenčnou moduláciou vlny.
- **Rádiolokátor** - detekuje objekty (napríklad, lietadlá alebo lode) a určuje ich vzdialenosť a uhol. Taktiež je schopný postupom času jednotlivé objekty sledovať.
- **MTI** - jedná sa o pulzný radar detekujúci pohybujúce sa objekty z odrazov použitím nízkeho PRF, ktorého dosah je obmedzený na 50 kilometrov, ale nedokáže jednoznačne určiť rýchlosť.
- **Pulzný Dopplerov radar** - rozlišujeme radar so stredným PRF a vysokým PRF. Oba typy využívajú Dopplerov posuv. Radar s vysokým PRF jednoznačne určí rýchlosť objektu, ale nezmeria presne vzdialenosť a radar so stredným PRF nezmeria presne ani rýchlosť, ani vzdialenosť.

- **Sledovací radar** - dokáže sledovať objekt a poskytuje jeho trajektóriu. Rozlišujeme:
  - **STT** - sleduje jeden objekt a poskytuje jeho presnú trajektóriu. Priebežný výpočet je za 0,1s.
  - **ADT** - sleduje veľké množstvo objektov. Priebežný výpočet je za 1s až 12s.
  - **TWS** - obsahuje informácie o všetkých objektoch v obmedzenej pozorovanej oblasti.
  - **Sledovač fázového poľa** - je schopný nepretržite sledovať objekty s vysokou rýchlosťou prenosu dát. Taktiež je schopný sledovať objekty s nízkou rýchlosťou prenosu dát podobne ako ADT.
- **Zobrazovací radar** - poskytuje dvojrozmerný obraz cieľa alebo scény (napríklad povrch zeme).
- **SLAR** - poskytuje vysokú presnosť vzdialenosti a slušnú presnosť uhla použitím antény vysielajúcej úzky paprsok lúča.
- **SAR** - jedná sa o zobrazovací radar na pohybujúcom sa vozidle poskytujúci vysokú presnosť vo vzdialenosti a priečnom rozsahu.
- **ISAR** - poskytuje vysoké rozlíšenie priečneho rozsahu využitím vzdialenosti a relatívneho pohybu cieľa. Môže byť pripevnený staticky, alebo na pohybujúcom sa vozidle.
- **Radar pre kontrolu zbraní** - jedná sa o radar používaný na obranu pred vzdušným útokom. Jedná sa v podstate o STT radar.
- **Navádzací radar** - je umiestený na rakete, ktorú navádza k cieľu.
- **Radar pre pozorovanie počasia** - použitie pri meteorologických účeloch. Jeho úlohou je napríklad detekovať, rozpoznávať a merať mieru zrážok alebo rýchlosť a smer vetra.
- **Dopplerov radar pre pozorovanie počasia** - používa posun Dopplerovej frekvencie pre meranie vetra, hlavne pre detekciu tornád alebo prúdenie vetra.
- **Radar pre rozpoznávanie objektov** - radar, ktorý je schopný určiť, aký objekt detekoval (napríklad auto, lietadlo, človek).
- **Multifunkčný radar** - spojenie funkčností z radarov popísaných vyššie do jedného radaru.

Radary podľa ich použitia[7]:

- **Vojenské**:
  - detekcia a rozpoznávanie cieľov, ovládanie zbraní
  - navádzacie systémy
  - identifikácia objektov v oblasti
- **Kontrola leteckej prevádzky**
  - detekcia a zobrazenie lietadiel v blízkosti letísk

  - skenovanie povrchu pristávacej plochy
  - navádzanie lietadla pri pristávaní

- **Vzdialené sledovanie objektov**
  - informácie o počasí
  - poloha vesmírnych telies
  - sledovanie morského ľadu

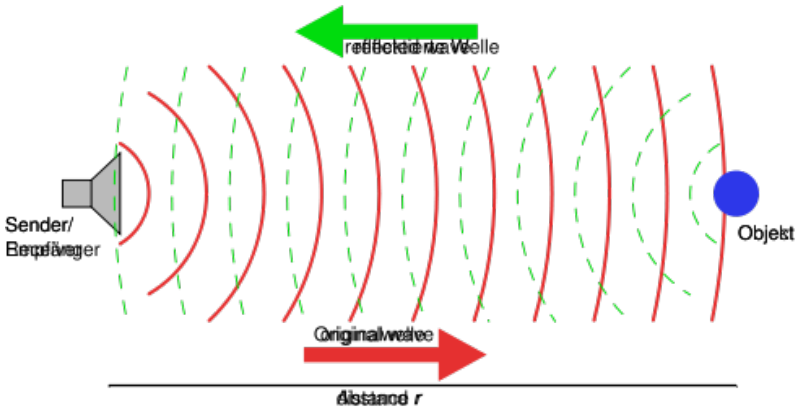
- **Riadenie pozemnej dopravy**
  - určenie rýchlosti vozidiel
  - varovanie pri výskyte prekážky pred alebo za vozidlom

- **Vesmír**
  - detekcia, sledovanie a monitorovanie vesmírnych telies
  - detekcia a sledovanie satelitov

Princíp radaru

Princíp radaru môže byť zhrnutý v nasledujúcich krokoch[29, 32], ktoré sú ilustrované na obrázku 2.2

- anténa radaru vysiela elektromagnetické vlny
- toto vlnenie sa šíri priestorom (smerom, ktorým sú nasmerované antény)
- vlnenie môže byť zachytené objektom (nazývaný taktiež cieľ) umiestneným v určitej vzdialenosti od radaru
- odchytené vlnenie sa odrazí od cieľa a šíri sa do všetkých strán
- časť vĺn (echo) zachytia radarové antény
- prijímač zosilní signály a rozhoduje, či sa jedná o echo sledovaného objektu alebo nie
- spracovanie informácii získaných z echo signálu



Obr. 2.2: Princíp operácie radaru. Vľavo môžme vidieť zdroj (radar) vysielaných vĺn, ktoré po zasiahnutí objektu sú ďalej odrazené. [6]

Radarová rovnica

Rovnica radarového rozsahu (skrátene radarová rovnica) je užitočná pri rátaní dosahu radaru, ale taktiež pri návrhu radarového systému. Môže byť zapísaná rovnicou[32, 14]:

$$P_r = P_d \times R_a \times A_e \tag{2.1}$$

kde:

- P_d - Výkonová hustota pri určitej vzdialenosti od antény
- R_a - Plocha v $[m^2]$, z ktorej sa signál odrazí naspäť k radaru
- A_e - Plocha antény, na ktorej sa zachytí sila echo signálu smerujúca k radaru

Výkonová hustota P_d:

$$P_d = \frac{P_t G_t}{4\pi R^2} \tag{2.2}$$

- P_t - výkon vysielača v [W]
- G_t - zosilnenie signálu v [dBi]
- R - Vzdialenosť od radara v [m]

Odrazová plocha R_a, z ktorej sa signál odrazí naspäť k radaru:

$$R_a = \frac{\sigma}{4\pi R^2} \tag{2.3}$$

- σ - Efektívna odrazová plocha objektu, ktorá určuje výkonovú hustotu odrazenú od objektu v $[m^2]$
- R - Vzdialenosť k radaru v [m]

Po upravení Radarovej rovnice dostaneme:

$$P_r = P_d \times R_a \times A_e = \frac{P_t G_t}{4\pi R^2} \times \frac{\sigma}{4\pi R^2} \times A_e = \frac{P_t G_t \sigma A_e}{(4\pi)^2 R^4} \tag{2.4}$$

Maximálny dosah radaru je vzdialenosť, pri ktorej sa ešte dokáže detekovať príslušný objekt so stanovenou kvalitou pri minimálnej výkonovej hustote $P_{r_{min}}$. Pre maximálny dosah radaru je možné upraviť rovnicu 2.4 na tvar[32]:

$$R_{max} = \sqrt[4]{\frac{P_t G_t \sigma A_e}{(4\pi)^2 P_{r_{min}}}} \tag{2.5}$$

Vo väčšine prípadov radary používajú anténu ako pre vysielanie, tak aj pre prijímanie. V takom prípade G_t je vzťah medzi zosilnením signálu a efektívnej odrazovej plochy objektu.[14]

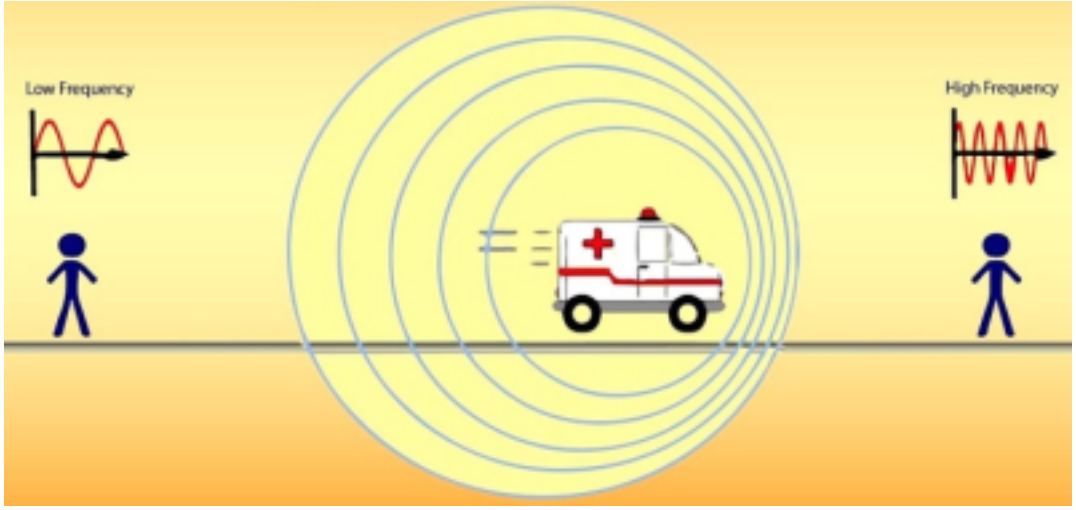
$$G_t = \frac{4\pi A_e}{\lambda^2} \tag{2.6}$$

Po dosadení vzťahu 2.6 do rovnice 2.5 dostaneme rádiolokačnú rovnicu:

$$R_{max} = \sqrt[4]{\frac{P_t G_t^2 \sigma \lambda^2}{(4\pi)^3 P_{r_{min}}}} \tag{2.7}$$

Dopplerov jav

V prípade, že zdroj vysielajúci elektromagnetické vlny je v relatívnom pohybe s pozorovateľom, dochádza k zmene medzi odoslanými a prijatými signálmi. Tento efekt sa nazýva Dopplerov jav a zmenená frekvencia Dopplerov posun alebo Dopplerova frekvencia.[14] Jav, pomenovaný po rakúskom fyzikovi Ch. Dopplerovi, nie je vyhradený len pre radarové vlny, ale dá sa aplikovať s použitím všetkých vĺn, vrátane zvukových. Pravdepodobne každý z nás už zažil Dopplerov jav, aj keď si to neuvedomoval. Napríklad si môžme predstaviť približujúce sa auto so sirénou (sanitka, hasiči a pod.). Pri približovaní sa zdá, že tón sirény je silnejší a v okamihu keď prejde okolo nás, sa tón stíši. [22, 35] Ukážku tohto príkladu môžme vidieť na obrázku 2.3.



Obr. 2.3: Na obrázku sa vyskytuje zdroj (sanitka) pohybujúca sa smerom vpravo a dvaja pozorovatelia (postavy vľavo a vpravo). Pravý pozorovateľ počuje hlasnejší tón ako pozorovateľ vľavo. Je to spôsobené tým, že ho zasiahne väčší počet zvukových vĺn. Každá z vĺn má kratšiu vzdialenosť medzi zdrojom (sanitka) a cieľom (pravý pozorovateľ) a tým pádom sa znižujú intervaly medzi jednotlivými zvukovými vlnami. [19]

Ak R je vzdialenosť medzi radarom a cieľom, konečná hodnota vlnovej dĺžky λ získaná z obojsmernej vzdialenosti medzi radarom a cieľom je $\frac{2R}{\lambda}$. Vzdialenosť R a vlnová dĺžka λ sú merané v rovnakých jednotkách. Keďže jedna vlnová dĺžka zodpovedá uhlovej odchýlke 2π rad, celková uhlová odchýlka Φ je $\frac{4\pi R}{\lambda}$. Ak je cieľ v pohybe, R a fáza Φ sa budú neustále meniť. Zmena Φ v závislosti od času je ekvivalent frekvencie.[31] Týmto je popísaná Dopplerova uhlová frekvencia ω_d:

$$\omega_d = 2\pi f_d = \frac{d\Phi}{dt} = \frac{4\pi}{\lambda} * \frac{dR}{dt} = \frac{4\pi v_r}{\lambda}$$

$$\omega_d = 2\pi f_d = \frac{d\Phi}{dt} = \frac{4\pi}{\lambda} * \frac{dR}{dt} = \frac{4\pi v_r}{\lambda} \tag{2.8}$$

kde:

- f_d = Dopplerov frekvenčný posun
- v_r = relatívna rýchlosť cieľa z pohľadu radaru

Dopplerov frekvenčný posun f_d:

$$f_d = \frac{2v_r}{\lambda} = \frac{2v_r f_0}{c} \tag{2.9}$$

kde:

- f_o = prenosová frekvencia
- c = rýchlosť šírenia = $3 \times 10^8 m/s$

Ak f_d je v hertzoch, v_r bude v uzloch a λ v metroch.

$$f_d = \frac{1.03v_r}{\lambda} \tag{2.10}$$

Informácie získavané z radarového signálu

Detekovanie objektov by nemalo zmysel, ak by sme z toho nič nezískali. Z radarového signálu môžme získavať niekoľko informácií o detekovanom objekte. Nasledujúce riadky sú prevzaté z [32]:

- **Vzdialenosť** - je určenie vzdialenosti medzi radarom a objektom na základe výpočtu dĺžky trvania šírenia radarového signálu rýchlosťou svetla medzi radarom a cieľovým objektom. Presnosť merania je závislá od šírky vlnového pásma radaru. Čím je vlnové pásmo širšie, tým je výpočet vzdialenosti presnejší.
- **Rýchlosť** objektu - je získaná z rozdielu vzdialeností v čase. Taktiež môže byť zmeraná z Dopplerovho posunu. Kvalita výpočtu je ovplyvnená časom potrebným pre samotný výpočet.
- **Smer** - je jedna z metód pre zistenie smeru objektu je na základe zistenia, kde je maximálny rozsah echo signálu zo skenovacej antény. Presnosť merania smeru je závislá na elektrickom výkone antény.
- **Veľkosť a tvar** - ak má radar dobrú schopnosť rozpoznávať vzdialenosť a uhol objektu, je možné zistiť veľkosť a tvar objektu. Využitím oblasti z Doplerovej frekvencie založenej na SAR alebo ISAR sa získa rozmer objektu. Je potrebné, aby boli v relatívnom pohybe k sebe navzájom cieľ s radarom, a taktiež radar musí mať dostatočne rozlíšenú vzdialenosť a priečny rozsah objektu.
- **Šírka pásma v radare** - rozlišujeme 2 druhy:
  - Šírka pásma signálu - je stanovená šírkou signálneho pulzu alebo vnútornou moduláciou signálu. V podstate sa jedná o jednoduchý impulz sínusovej vlny o trvaní $t => \frac{1}{t}$. Väčšia šírka pásma je potrebná pre rozpoznanie cieľov v dosahu, presné zmeranie vzdialenosti cieľa, alebo schopnosť odlíšiť jeden typ cieľa od iného.
  - Laditeľná šírka pásma - umožňuje zmeniť frekvenciu radarového signálu v širokom rozsahu dostupného spektra. Je to užitočné pre zredukovanie množstva vzájomne sa rušiacich radarov pôsobiacich na tej istej frekvencii, ako aj pri pokuse o zníženie obranyschopnosti nepriateľských protiopatrení.

Po 2. svetovej vojne radar mohol operovať približne na tretine mikrovlného spektra. Toto spektrum bolo v priebehu rokov redukované kvôli pribúdajúcim službám, ktoré taktiež využívajú túto časť spektra. Radar preto operuje len na menšom rozsahu mikrovlného spektra a je extrémne dôležité vybrať správnu šírku pásma pre správne fungovanie radarových aplikácií.

- **Pomer signálu a šumu** - je dôležitý pre presnosť výpočtov a meraní. Závisí od vzťahu $\frac{E}{N_o}$ kde E je celková energia obdržaného signálu, ktorá je spracovaná radarom a N_o je sila šumu na jednotku šírky pásma prijímača.

- **Dopplerov posun v radare** sa používa pri rozlišovaní pohybujúcich sa objektov od statických. Všetky moderné radary pre riadenie letovej prevádzky, vojenské pozemné radary, ako aj radary používané vo vojenských stíhačkách využívajú Dopplerov jav. Ďalšie významné uplatnenie radara je pri pozorovaní počasia.

- **Obsluha viacerých frekvencií** - Radar môže mať schopnosť operovať na viacerých frekvenciách. Rozoznávame frekvenčnú obratnosť a frekvenčnú rozmanitosť. Frekvenčná obratnosť využíva viaceré frekvencie na základe pulz-pulz, avšak nie je kompatibilná s dopplerovým spracovávaním (rozpoznávanie pohybujúcich sa objektov v odrazených signáloch). Frekvenčná rozmanitosť sa spája s použitím viacerých frekvencií, ktoré sú v širokých rozostupoch a dokonca niekedy vo viacerých radarových pásmach. Táto rozmanitosť je kompatibilná s dopplerovským spracovávaním. Frekvenčný rozsah vo frekvenčnej obratnosti aj rozmanitosti je oveľa vyšší ako je základná šírka pásma impulzu šírky τ.

- **Nulová výška** - Operovanie radara na jednej frekvencii môže mať za výsledok, že bude znížené pokrytie niektorých výškových uhlov a u iných bude zvýšená sila signálu. Zmena vo frekvencii zmení polohu týchto uhlov tak, aby sa operovalo nad širším frekvenčným rozsahom. Výsledkom bude menšia pravdepodobnosť straty echo signálu.

- **Zvýšenie detekcie cieľa** - V prípade, že radar operuje len na jednej frekvencii, je možné, že z detekovaného cieľa sa bude vracať slabá odozva, a tým pádom daný objekt nedetekuje. Pravdepodobnosť detekovania takéhoto objektu môžme zvýšiť tým, že necháme radar operovať na viacerých frekvenciách. Takáto situácia sa však týka len zložitejších objektov (napríklad lietadlo). To je dôvodom, prečo radary pre sledovanie letovej prevádzky zvyčajne operujú na viacerých frekvenciách.

- **Zníženie účinnosti nepriateľských protiopatrení** - Hocijaký úspešný vojenský radar môže byť rušený iným nepriateľským radarom. Preto radary zvyčajne operujú na viacerých frekvenciách. Taktiež zmena frekvencie je nepredvídateľná a teda úspešná obrana proti rušeniu. Frekvenčná rozmanitosť v širokom pásme zasa spôsobuje, že je obtiažne určiť polohu vysielača.

Detekcia objektov

Dôležitou úlohou pri detekcie objektov radarom je nájdenie výrazných vrcholov v odrazenom signále a súčasne odstránenie šumu a vrcholov nepatriacich žiadnemu objektu.[27] V nasledujúcich riadkoch bude popísaných niekoľko metód pre detekciu objektov.

Algoritmus CLEAN

Základnou myšlienkou je odhad CIR $h(t)$. Na základe prenášaného signálu so znalosťou $h(t)$ by znamenalo, že je možné vybrať šum cieľového signálu. V rovnici 2.11 získaný signál môže byť popísaný ako konvolúcia CIR s prenosovým signálom, kde $*$ značí operátor konvolúcie.[27]

$$\xi(t) = h(t) * s(t) + n(t) \tag{2.11}$$

Zistenie $h(t)$ je rovnaké ako dekonvolúcia CIR s prenosovým signálom. Fourierova transformácia CIR môže byť spočítaná ako $H(f) = \frac{\xi(f)}{S(f)}$, čo je nestabilné ak $S(f) \longrightarrow 0$. Algoritmus CLEAN aplikuje iteračné prístupy pre vyhnutie sa zlého postavenia na dosiahnutie aproximácie CIR.[27] Počas dekonvolúcie vzorkovaný CIR h_k je nájdený prvými výpočtami vzájomnej koreláce rovnice 2.11 medzi prenášaným a získaným signálom v cykle. Je inicializovaná mapa ekvivalentná s prijatým signálom, ktorá je použitá pre odobratie vzoriek z vysoko korelovaných vzoriek. Iterácie pokračujú, kým sa nedosiahne predom určený prah. Počas každej iterácie je spočítaná maximálna hodnota vzájomnej korelácie a vzorky sú pridané do vektora obsahujúceho odhadované CIR. Akonáhle je zistený vrchol, odstráni sa zo signálu a taktiež všetky susediace vzorky sú nahradené nulami. Tento proces pokračuje, až kým nie je možné nájsť žiadny vrchol. Výsledkom algoritmu je vektor vzoriek s hodnotami, kde ξ_k a s_k sú najviac korelované.[27]

$$R_{\xi_s} = \xi_k * s_k \tag{2.12}$$

Algoritmus CFAR

Detekčný prispôsobivý algoritmus používajúci sa v radarových technológiách pre detekciu návratnosti z cieľov proti pozadiu šumu a rušeniu. CFAR algoritmy využívajú metódu posuvného okna pre odhad vrcholov v prijatom signále. Toto je spravené pod podmienkou, že hodnota falošnej signalizácie nemôže presiahnuť predom definovanú hodnotu. Táto vlastnosť robí algoritmy CFAR užitočné v kritických systémoch, ako je napríklad proti-raketový obranný radarový systém. CFAR metóda očakáva ohraničený vstupný signál, čo znamená, že musí byť vypočítaný výkon spojením radarových skenov. Dôležité parametre CFAR sú CUT (testované bunky), G (strážne okná) obklopujúce CUT a vedúce a oneskorené referenčné okná ($N_{lead,lag}$) obklopujúce ochranné bunky. Koncept vzoriek reprezentujúci tieto parametre je znázornený v obrázku 2.4.[27]

Obr. 2.4: Rozdielne bunky posuvného okna algoritmu CFAR[27]

Táto metóda odhaduje silu šumu z každého referenčného okna $N_{lead,lag}$ a porovnáva CUT výkon Z s prahom T. Všeobecný výpočet prahu je daný rovnicou 2.13.

$$T = \hat{g}(N_{lead}, N_{lag})\alpha \tag{2.13}$$

kde:

- $\hat{g}(N_{lead}, N_{lag})$ - odhadovaná sila šumu

- α - CFAR konštanta

Detekcia môže byť zhrnutá do binárneho hypotetického testu 2.14, kde 1 znamená detekciu a 0 žiadnu detekciu.

$$Z \underset{0}{\overset{1}{\lessgtr}} T \tag{2.14}$$

Algoritmus CA-CFAR

Jedná sa o jeden z najrozšírenejších CFAR algoritmov. V tomto algoritme je sila šumu vypočítaná spriemerovaním výkonu v referenčných bunkách. Predpoklady o štatistikách šumu sa robia pri derivovaní α pre algoritmus CA-CFAR. Na odvodenie výrazu pre konštantu CA-CFAR algoritmu sa predpokladá, že sila prijatého signálu je exponenciálne distribuovaná podľa rovnice 2.15[27]

$$p_x(X) = \frac{1}{\sigma_i^2} e^{\frac{-x}{\sigma_i^2}}, X \geq 0 \tag{2.15}$$

kde:

- X - štvorcový ohraničený výstup (napríklad výkon)
- σ_i^2 - priemerná sila šumu získaná z referenčných buniek

Integrovaním rozdelenia pravdepodobnosti z rovnice 2.15 sa získa výraz pre pravdepodobnosť detekcie P_D. Výsledným prahom je teda rovnica 2.16, v ktorej konštanta CA-CFAR algoritmu je daná v 2.17.[27]

$$T_{CA} = \alpha_{CA} \hat{\sigma}_i^2 \tag{2.16}$$

$$\alpha_{CA} = N[P_{FA}^{-\frac{1}{N}} - 1] \tag{2.17}$$

$$\sigma_i^2 = \frac{1}{N} \sum_{n=1}^{N} r_n \tag{2.18}$$

kde:

- N - celková dĺžka referenčných okien $N_{lead,lag}$
- σ_i - odhadovaný interferenčný výkon v referenčných bunkách daných rovnicou 2.18
- $R = \{r_1, r_2, \ldots, r_n\}$ - vzorky v referenčných bunkách

Algoritmus CA-CFAR je vhodný pre použitie v homogénnych prostrediach, alebo pri detekcii len jedného objektu. V heterogénnych prostrediach, poprípade pri detekcii viacerých objektov bude zvýšený prah, zvýši sa počet nesprávnych detekcií a vo výsledku samotná detekcia zlyhá.[27]

Algoritmus GCMLD-CFAR

Algoritmus GCMLD-CFAR je jedným z algoritmov riešiacich problém vyskytujúci sa v algoritme CA-CFAR a to je detekcia viacerých cieľov. Patrí medzi takzvané Robustné CFAR algoritmy. Myšlienkou je cenzúrovanie silných rušení z referenčných buniek pre dosiahnutie presnejšieho zmerania sily šumu v pozadí. Úroveň cenzúry je daná pravdepodobnosťou klamlivej cenzúry P_{FC}. Zvyčajne je táto hodnota rovnaká ako P_{FA}.[27]

P_{FA} je derivovaná z P_D, čo je získané integrovaním rozdelenia pravdepodobnosti predpokaldanej sily signálu. Výsledná konštanta α a prah T sú zobrazené v rovnici 2.19 kde:

- N - dĺžka referenčných okien
- $\hat{\sigma}^2_{i,censored}$ - predpokladaná spriemerovaná sila šumu v referenčných bunkách po cenzúre

$$T_{GCMLD} = \hat{\sigma}^2_{i,censored} \tag{2.19}$$

$$\alpha_{GCMLD} = (N - m)P_{FA}^{\frac{1}{m-N}} - 1 \tag{2.20}$$

Sledovanie objektov

V tejto podkapitole je opísaný jeden z algoritmov pre sledovanie objektov vhodný na sledovanie ľudí. Jedná sa o algoritmus MHT, ktorý je schopný sledovať súčasne viacero objektov. V nasledujúcich častiach bude podrobnejšie vysvetlený.

Model merania a pohybu

Pulzy vysielané radarom sa šíria a odrážajú od objektov. Odrazené pulzy sú merané jedným alebo viacerými prijímačmi radaru.[27] Hodnoty získané z radaru sú reprezentované rovnicou merania

$$\xi = C(r) + n \tag{2.21}$$

kde:

- ξ - meranie
- n - šum merania
- r - vzdialenosť k objektu
- $C(r)$ - matica merania, získaná v diskrétnom čase rovnicou 2.22, v ktorej k znamená index vzorky

$$C(r_k) = \sqrt{x_k^2 + y_k^2} \tag{2.22}$$

Rovnica merania je nelineárna a stavy sledovaných objektov sú skryté, preto nie je možné použiť niektorú zo štandardných techník ako je napríklad Kalmanov filter. Je nutné rovnicu zlinearizovať okolo (x_k, y_k) pomocou 1. rádu Taylorovho rozvoja za predpokladu, že sledované objekty sú v pohybe. Výsledkom je linearizovaná rovnica merania 2.23.[27]

$$H(r_k) = \begin{bmatrix} \frac{x_k}{\sqrt{x_k^2+y_k^2}} & \frac{y_k}{\sqrt{x_k^2+y_k^2}} \end{bmatrix}^T \tag{2.23}$$

Hoci presný model pohybu sledovaného objektu je neznámy, môžme predpokladať model konštantnej rýchlosti Brownianovho pohybu, ktorý je popísaný v jednej dimenzii rovnicou 2.24[27]

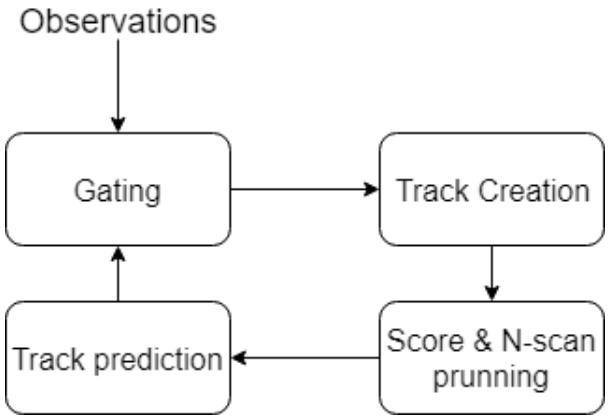
$$r_{k+1} = r_k + \sigma_{v_k} \tag{2.24}$$

kde:

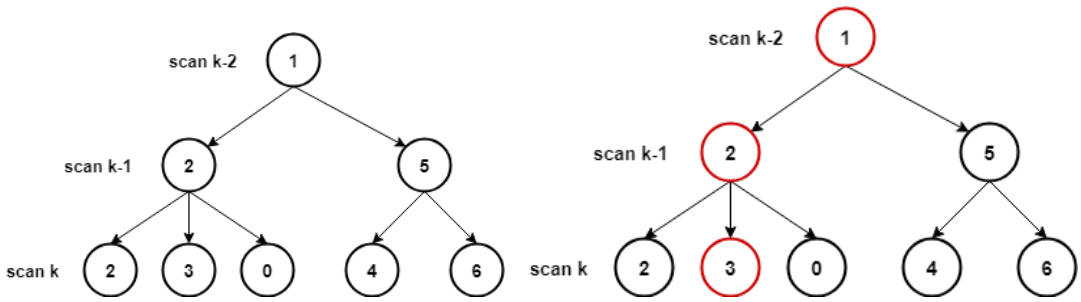
- r - dosah
- σ_v - produkovaný šum

TO-MHT

Jedná sa o obľúbený algoritmus pre sledovanie objektov v náročnom prostredí. Stopa objektu je sekvencia hypotéz vytvorených po získaní záznamov z viacerých pozorovaní. Jadrom MHT je získanie záznamov v určitom kroku, vytvorením hypotéz a ich uložením pre použitie v neskoršej časovej inštancii. Algoritmus TO-MHT vytvára stromovú štruktúru, kde uzlami v tomto strome sú hypotézy. Ako sú získavané nové záznamy, vytvárajú sa nové vetvy alebo rozširujú už existujúce. Jednotlivé stopy sú v podstate „cestou“ hypotéz v strome. Logika algoritmu MHT je zobrazená v obrázku 2.5 a štruktúra stromu v obrázku 2.6.[27]



Obr. 2.5: Všeobecná logika algoritmu MHT



Obr. 2.6: Na obrázkoch sa nachádza stromová štruktúra algoritmu MHT. Čísla vnútri uzlov sa viažu k pozorovacím záznamom. Číslo 0 znamená „dummy“ pozorovací záznam (záznam, ktorý nie je spojený so žiadnym meraním). V pravom obrázku je vyznačená trasa sledovaného objektu.

Procedúry algoritmu:

Gating - procedúra spájajúca merania so stopami. Pre existujúce stopy je spojenie založené na štatistike vypočítanej z predchádzajúcej predpovedi stavu. Ak umocnená Mahalanobis vzdialenosť d^2 je nižšia ako určený prah, meranie je považované, že sa nachádza vo vnútri brány.[27]

$$d^2 = (\hat{x}_k^l - y_{i_k})^T (\sum_k^l)^{-1} (\hat{x}_k^l - y_{i_k}) \lesseqgtr G_{th} \tag{2.25}$$

kde:

- $\hat{x}_k^l$ - predpoveď stavu
- y_{i_k} - záznam z pozorovania i^{th}
- G_{th} - veľkosť brány
- $\sum_k^l$ - inovatívna kovariancia stopy l^{th} v čase k vypočítaná rovnicou 2.26.

$$\sum_k^l = HP_{k-1}^l H^T + R \tag{2.26}$$

kde:

- R - kovariantná matica merania
- P_{k-1} - stav kovariancie stopy odhadnutý na základe posledného merania

Veľkosť inovatívnej brány G_{th} je χ distribuované s n stupňami slobody, pričom n je rozmer stavov. V prípade, že sa jedná o dvojdimenzionálny rozmer, rovnica 2.25 zodpovedá sústredenej kružnice na predpovedanej polohe stôp.[27]

Ak meranie spadá pod validačnú bránu, inicializuje sa hypotéza spojená s meraním. Totožné merania môžu teoreticky generovať viacero hypotéz k rôznym stopám, avšak daná situácia je riešená v neskoršej fáze algoritmu MHT. Ak meranie nespadá pod validačnú bránu niektorej zo stôp, alebo stopa neexistuje, vytvorí sa nový strom stôp s hypotézami spojenými s konkrétnym meraním.[27]

Track Scoring - každý uzol v strome je ohodnotený. Toto hodnotenie je založené na pravdepodobnostnom pomere LR, čo je pravdepodobnosť validity stopy. Všeobecná definícia LR je uvedená v rovnici 2.27

$$LR = \frac{p(D|\mathcal{H}_1)P_0(\mathcal{H}_1)}{p(D|\mathcal{H}_0)P_0(\mathcal{H}_0)} \tag{2.27}$$

kde:

- $\mathcal{H}_{1,0}$ - hypotéza skutočnosti cieľového objektu
- D - prijaté dáta
- $p(D|\mathcal{H}_i)$ - funkcia hustoty pravdepodobnosti prijatých dát za predpokladu, že $\mathcal{H}_i$ je správne
- $P_0(\mathcal{H}_i)$ - prioritná pravdepodobnosť $\mathcal{H}_i$

V praxi sa používa logaritmus LR, ktorý je definovaný ako

$$LLR \triangleq \ln\left(\frac{P_T}{P_F}\right) \tag{2.28}$$

LLR skóre S_k^l v čase k sa počíta rekurzívne ako

$$S_k^l = S_{k-1}^l + \Delta S_k^l \tag{2.29}$$

kde rozdiel v skóre je daný:

$$\Delta S_k^l = \begin{cases} \approx \ln(1 - P_D), & i = 0 \\ \ln \frac{V_s}{2\pi} - \frac{1}{2}\ln(|\sum_k^l|) - \frac{d^2}{2}, & i \neq 0 \end{cases} \tag{2.30}$$

kde V_s je merací priestor, ktorý v prípade dvojdimenzionálneho sledovania je sledovanou oblasťou. Pri vytvorení novej stopy, skóre je inicializované s rovnakou hodnotou ako pre nezachytenú stopu. Ako sa pridávajú nové hypotézy, skóre narastá. To znamená, že v každom spodnom uzle stromu hypotéz sa bude nachádzať súčet LLR skóre každej zodpovedajúcej vetvy. LLR môže byť taktiež skonvertované do pravdepodobnosti, či sú stopy validné rovnicou 2.31.[27]

$$P_{track} = \frac{e^{LLR}}{1 + e^{LLR}} \tag{2.31}$$

Pravdepodobnosť P_{track} sa používa pri vyhodnotení, či by mala by+t stopa zmazaná, alebo nie. Typickou hranicou pre zmazanie stopy je $\approx 1 \times 10^{-3}$[27].

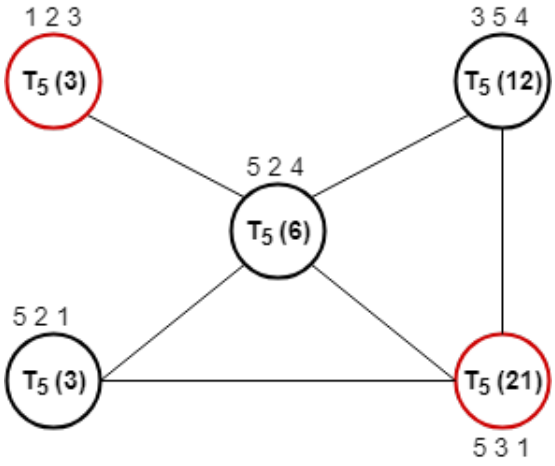
Track pruning - keďže počet generovaných hypotéz rastie exponencionálne, priebežne systematickým spôsobom sa musia mazať stopy. Mažú sa nepravdepodobné stopy. Nižšie sú popísané dva spôsoby, ako sú stopy odstraňované.[27]

- **Score pruning** - v každej iterácii sú stopy ohodnotené podľa rovnice 2.30 a pravdepodobnosť, či je stopa validná je získaná z rovnice 2.31. Stopy, ktorých pravdepodobnosť P_{track} je nižšia ako určená hranica budú zmazané.[27]
- **Global pruning** - sa vykonáva hľadaním súboru trás s maximálnym LLR skóre, ktoré sú kompatibilné. Aby trasa bola kompatibilná s ďalšou, nesmú mať spoločné pozorovacie záznamy s inou trasou v tom istom čase. Trasy vychádzajúce z globálnych trás N krokov zo stromu sú mazané zo stromov hypotéz. Toto je nazývané „N-sken mazanie“, kde N je integer hodnota vybraná zvyčajne v rozmedzí 3-5.[27]

Metóda nájdenia kompatibilného súboru trás s maximálnym skóre môže byť formulovaná ako MWIS problém. Stromová štruktúra hypotéz trás je reprezentovaná ako neorientovaný graf G. Každá trasa T_l v grafe je reprezentovaná vrcholom $v_1 \in V$ kde V je súbor všetkých trás. Ak trasy T_l a T_j majú spoločný záznam z pozorovania, sú spojené hranami $(l, j) \in E$ v grafe. Každý vrchol má vlastnú váhu, čo je v podstate LLR skóre trasy. Nezávislý súbor maximálnych váh je súborom trasových hypotéz, ktoré spolu poskytujú sumu hmotností bez toho, aby mali zdielané pozorovacie záznamy v tom istom čase. Tento problém môžme formulovať ako

$$\max_v \sum_l w_l v_l, \quad v_l + v_j \leq 1, \quad \forall (l, j) \in E, \quad v_t \in \{0, 1\} \tag{2.32}$$

kde nerovnosť vrcholov znamená, že daná trasa môže byť použitá len raz.[27] Na obrázku 2.7 je zobrazený príklad trás s váhami a sekvenciou pozorovacích záznamov.



Obr. 2.7: Graf vytvorený zo stromovej reprezentácie. MWIS je znázornený červenou farbou. V uzle sa nachádza jeho váha, nad uzlom sa vyskytujú jeho pozorovacie záznamy.[27]

Predpovedanie trás - časť MHT, ktorá predpovedá, kde sa cieľ bude nachádzať v určitom čase. Predpovede sú potrebné pre určenie polohy cieľa v čase a taktiež sa používajú v ďalšej iterácii MHT pre spočítanie validačnej brány pre združenie nových meraní. Technika, ktorá je využitá v MHT implementácii je EKF.[27]

Keďže výsledkom EKF nie je presný model, je možné ho nahradiť filtrovaním častíc, ktoré netrpí rovnakými problémami ako EKF. Prezentované stavy sú dvojdimenzionálne s meraniami v jednej dimenzii a nelienárnou rovnicou. Tieto stavy sú reprezentované Skrytým Markovským modelom, ktorý môže byť riešený filtrovaním častíc.[27]

Každá inicializovaná trasa v MHT predpovedá jej stav s oddeleným filtrom častíc. Procedúra Gating je ovplyvnená presnosťou týchto stavov, pri nízkej presnosti sa zvyšuje veľkosť brány. Koncept filtrovania častíc je podrobnejšie vysvetlený nižšie.[27]

Zlučovanie a delenie trás - zlučovaním nazývame situáciu, pri ktorej sa dva ciele zlúčia do jedného. Delenie je zasa rozdelenie jedného cieľa do menších skupín. Systémy, ktoré detekujú trasy viacerých objektov musia korektne zaobchádzať s jednotlivými udalosťami. Trasy s podobným stavom alebo pozorovacími záznamami sú zlučované. Keďže radary zvyčajne merajú len vzdialenosť, delenie trás je veľmi náročné, keďže po rozdelení by sa mohli stratiť informácie.[27]

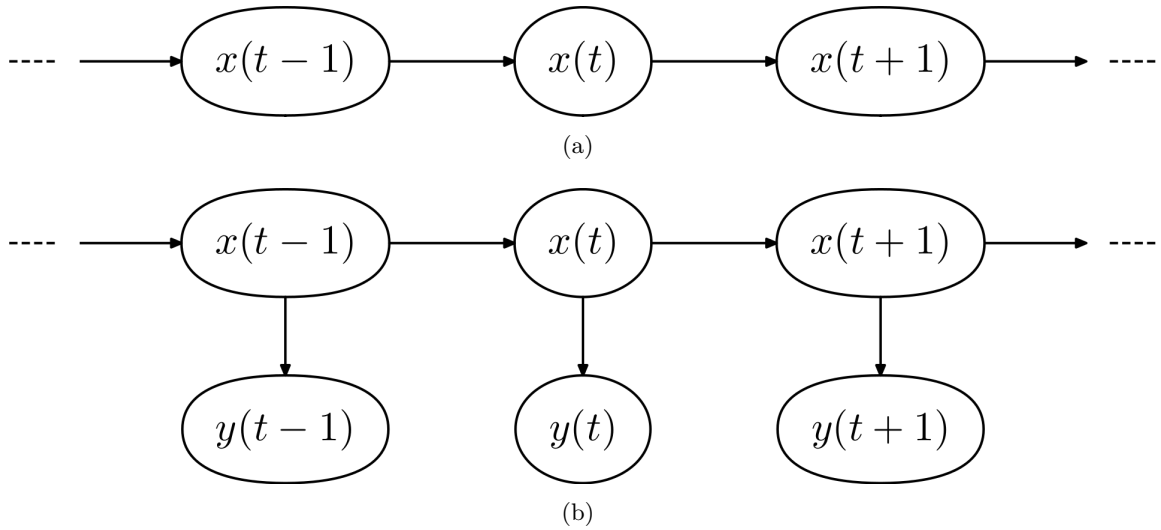
Zhlukovanie trás - pre dosiahnutie väčšej účinnosti algoritmu MHT je potrebné vytvoriť zhluky hypotéz. Princípom je zjednotiť všetky trasy, ktoré sú nekompatibilné. Prirodzene táto situácia nastáva pri vytváraní alebo rozširovaní stromu, keďže vetvy toho istého stromu vychádzajú z rovnakého koreňa. Ďalšou možnosťou je vytvárať zhluky z tých trás, ktoré zdieľajú pozorovacie záznamy. Keď sú zhluky vytvorené, je možné časti algoritmu ako gating a scoring robiť nezávisle od zhlukov, čiže je možné paralelné spracovanie.[27]

Skryté Markovské modely

Jedná sa o spôsob modelovania zmien v stavoch stochastického systému. Model predpokladá, že systém splňuje Markovovu vlastnosť, teda že budúce stavy sú podmienene nezávislé na všetkých predchádzajúcich stavoch okrem aktuálnych. Keď označíme sekvenciu diskrétnych stavov ako $\{x_1, x_2, \dots\}$, potom $X = \{X_1, X_2\}$ je možná sekvencia stavov náhodného systému nadobudnutá v momentálnom čase. Markovova vlastnosť je potom vyjadrená ako[27]:

$$P(X_{k+1} = x_{k+1}|X_0 = x_0, X_1 = x_1, \dots, X_{k-1} = x_{k-1}, X_k = x_k) = P(X_{k+1} = x_{k+1}|X_k = x_k) \tag{2.33}$$

Nadobudnutá sekvencia stavov $X_{i:k}$ náhodného systému sa nazýva Markovova reťaz. V stacionárnom Markovovom modely sú stavy systému poznamenané priamo, avšak v Skrytom Markovom modely sú zaznamenané len tie pozorovacie záznamy, ktoré sú príbuzné s aktuálnym stavom systému. Rozdiel medzi obyčajným a skrytým Markovským modelom je znázornený na obrázku 2.8 .[27]



Obr. 2.8: Stacionárny (a) a Skrytý (b) Markovský model

Filtrovanie častíc

Typ metódy Monte-Carlo pre aproximáciu PDF stavu systému popísaného modelom HMM. Stav $(x, y, \dot{x}, \dot{y})$ je prezentovaný sadou N vzorkou $\{x^i_{k-1}\}^N_{i=1}$ rôznych hmotností nazývaných častice, kde i ukazuje na i-tu časticu. Častice sú pre-vzorkované na základe aktualizovaných hmotností, aby sa predišlo tomu, že niektoré častice budú mať zanedbateľné váhy. Kroky filtrovania častíc môžme rozdeliť nasledovne:[27]

- **Predpoveď** - šírenie častíc použitím predchádzajúceho PDF ako návrhového rozdelenia $q(x^i_k|x^i_{k-1}, z_k) = p(x^i_k|x^i_{k-1})$. Týmto sa získa sada šírených častíc $\{x^i_k\}^N_{i=1}$.
- **Aktualizácia hmotností** - spočítanie hmotností w^{-i}_k každej častice. Váhy sú spočítané ako pravdepodobnosť vyhodnotenia merania danej častice získanej $w^{-i}_k = p(z_k|x^i_k)$. Toto rozdelenie je nazývané taktiež ako senzorový model a zobrazuje hodnoty, ktoré stavy pravdepodobne budú mať. Váhy sú normalizované.

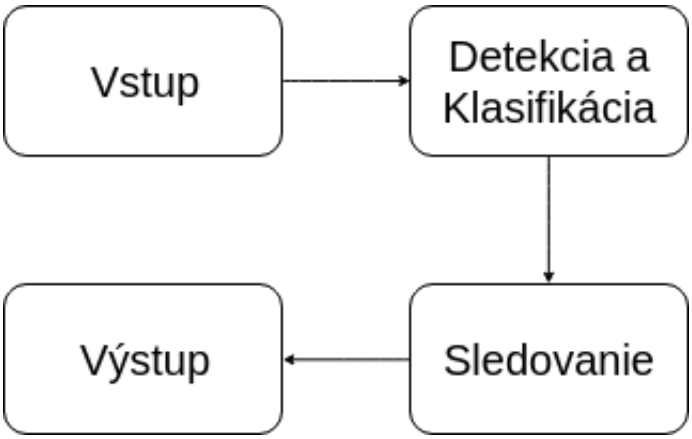
- **Pre-vzorkovanie** - pravdepodobnosťami získanými z normalizovaných váh je N-krát prekreslená nová sada častíc $\{x_k^{i*}\}_{i=1}^N$. Tieto častice budú skôr distribuované podľa $\{x_k^{i*}\}_{i=1}^N \sim p(x_k|z_{1:k})$, čo je rozloženie vyhľadávania.

Hlavným parametrom filtrovania častíc je množstvo častíc, ktoré sa použije.

2.2 Detekcia objektov v obraze

V tejto sekcii sú popísane spôsoby detekcie objektov v obraze, ako aj sledovanie týchto objektov. Jedná sa však len o výber algoritmov a spôsobov z danej oblasti pre detekciu a sledovanie objektov v obraze.

Na obrázku 2.9 je znázornená schéma princípu detekcie. Vstupom je zosnímaný obrázok, alebo iná forma relevantných dát obsahujúcich hľadaný objekt z oblasti záujmov. Následne sa v týchto dátach detekuje a klasifikuje objekt na základe niektorého z algoritmov vhodných pre detekciu objektov v obraze. Objekty následne môžu byť identifikované na základe ich trás, avšak len počas tej doby, čo sa daný objekt nachádza v sledovanej oblasti. Výstupom sú dáta s informáciami o týchto detekovaných objektov.



Obr. 2.9: Všeobecná schéma detekcie v obraze

Výber oblasti záujmov

Namiesto skenovania celého snímku sa vyberie len určitá časť, prezývaná oblasť záujmov, ktorá je následne poslaná do klasifikačného modulu. Tento krok sa nazýva ROI krok, alebo segmentácia v popredí. Je to dôležité, aby sa vylúčili časti neobsahujúce objekty, napríklad obloha.[28, 18]

Najjednoduchšou metódou pre získanie počiatočnej scény, v ktorej sa vyskytuje objekt alebo objekty, je algoritmus Sliding Window. V podstate sa jedná o posun detekčného okna cez všetky pozície objektov v obrázku v rôznych mierkach. Keďže obraz môže obsahovať tisíce detekčných okien, tento proces je vysoko náročný pre výpočetný výkon. Zrýchlenie sa môže dosiahnuť tým, že sa spojí algoritmus Sliding Window s klasifikátorom kaskády rastúcej zložitosti.[18]

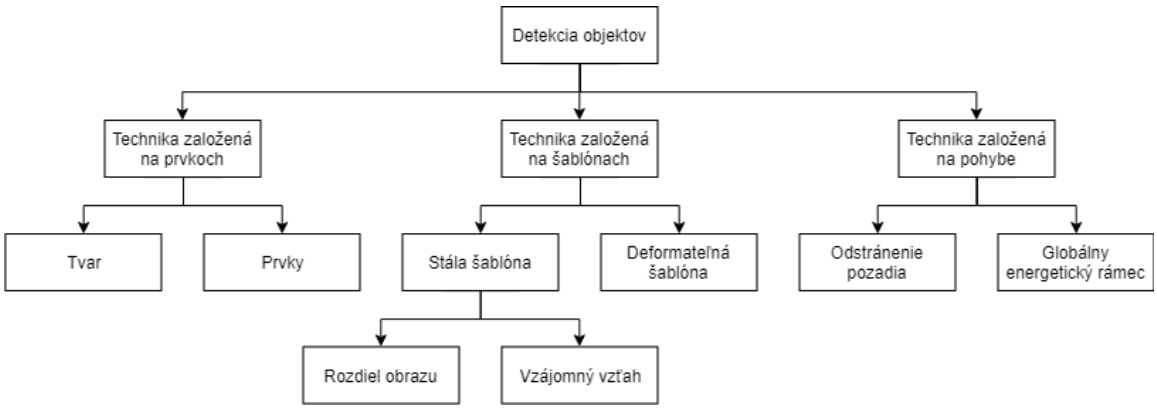
Ak poznáme klasifikáciu hľadaného objektu a geometriu kamery, ďalšou možnosťou je obmedzenie vyhľadávacieho miesta v snímke. Je to založené na predpoklade plochého sveta, alebo určenej veľkosti hľadaného objektu. Čiže, aby bol vyhľadávaný objekt detekovaný, musí spĺňať podmienku obmedzenia veľkosti, čo výrazne zvýši rýchlosť počítania.[18, 28]

Detekcia objektov

Techniky pre detekciu objektov sa môžu rozdeliť do troch kategórií:

- na základe prvkov
- na základe šablóny
- na základe pohybu

Princíp všetkých troch kategórií je znázornený na obrázku 2.10



Obr. 2.10: Blokový diagram princípov detekcie objektov. [10]

Na základe prvkov

Základným prvkom týchto algoritmov je získanie vlastností a ich následná klasifikácia v pred-trénovaných rozpoznávacích systémoch. Tieto algoritmy by mali byť použité na jednotlivých snímkach ako na ich sekvencii. Taktiež ich jadro môžme rozdeliť na dve kategórie:

- výber prvkov - klasifikácia týchto prvkov a trénovanie systému pre rozpoznávanie
- klasifikácia týchto prvkov z natrénovaného rozpoznávacieho systému

Ideálny detektor by mal byť schopný rozpoznať vlastnosti objektu nezávisle od zmien vo svetle, perspektívy, alebo mierky. Hlavným problémom je výber prvkov, na základe ktorých sa bude rozpoznávať. Napríklad rozpoznávanie založené na farbách musí byť schopné rozlišovať všetky možné farebné kombinácie na oblečení, kým rozpoznávanie podľa tvaru musí byť schopné rozlíšiť na základe tvarov, umiestenia daného objektu a taktiež pridané prvky, ako sú šperky. Tieto kroky sú zvyčajne výpočtovo náročné. [18, 10]

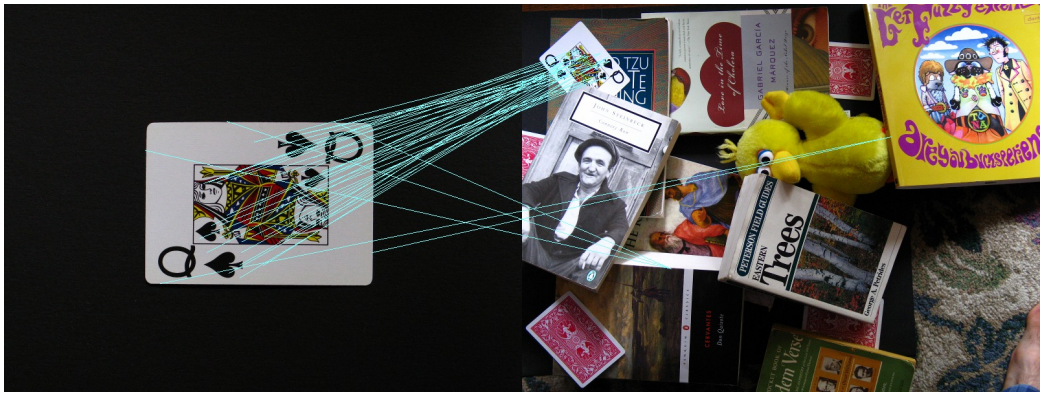
Najbežnejšie používané funkcie a ich popis:

- **Bodové detektory** - cieľom je nájsť kľúčové body objektu, na základe ktorých môže byť daný objekt klasifikovaný. Bežne používané detektory sú napríklad Harris corner detektor[2, 30], Harris-Laplace[20] detektor, alebo Difference of Gaussians[9].[18]

  Jeden z najpoužívanejších krokov je SIFT deskriptor. V tejto metóde je obsah obrázka transformovaný do lokálneho súradnicového systému a využíva orientáciu a mierku získanú z detektora kľúčových bodov, aby vytvoril orientačný histogram s váhami

založenými na rozsahu a tak získal preklad, rotáciu a mierku pre SIFT deskriptor.[18, 16]

HOG predstavený Dalal a Triggs[25] bol populárny vďaka jeho zlepšeniam vo funkciách založených na intenzite. Kvôli jeho vysokej úspešnosti detekovania sa jedná o jednu z najpopulárnejších metód pre detekovanie objektov, hlavne sa využíva v oblasti detekcie chodcov, ale je možné ho použiť i v oblastiach detekcie zvierat, alebo vozidiel. Vstupný snímok je spracovávaný v štyroch krokoch: normalizácia gamy a farieb, výpočet sklonu, zaznamenanie váh do priestorových a orientačných buniek, normalizácia kontrastu cez prekrývajúce sa bloky v priestore. Nevýhodou je však pomalý výpočtový výkon, kvôli čomu sa nemôže použiť pre detekciu v reálnom čase.[18, 28]



Obr. 2.11: Znázornený SIFT deskriptor. Na obrázku v ľavej časti sa nachádza karta, ktorá má byť detekovaná v predmetoch v pravej časti obrázka.[5]

- **Detektor častí a končatín** - používa sa pri detekcii chodcov. Detekcia chodcov je rozdelená na detekciu jednotlivých častí (nohy, ruky, hruď, hlava). Pre každú časť sa trénuje klasifikátor a taktiež sú tieto časti detekované samostatne až vznikne výsledná detekcia.[18] To znamená, že ak osoba zmení polohu a niektorá z končatín nie je detekovateľná, systém stále dokáže na základe ostatných končatín určiť, či sa jedná o chodca, alebo nie. [16]

- **Detektor založený na vlnke (z en. wavelet)** - rovnice sú vypočítané z rozdielu medzi súčtom hodnôt intenzít pre oblasť pixelov a orientáciou, mierkou a pozíciou.. Funkcie sú jednoduché a vypočítané rýchlo vďaka použitiu integrálnej obrazovej technike.[18]

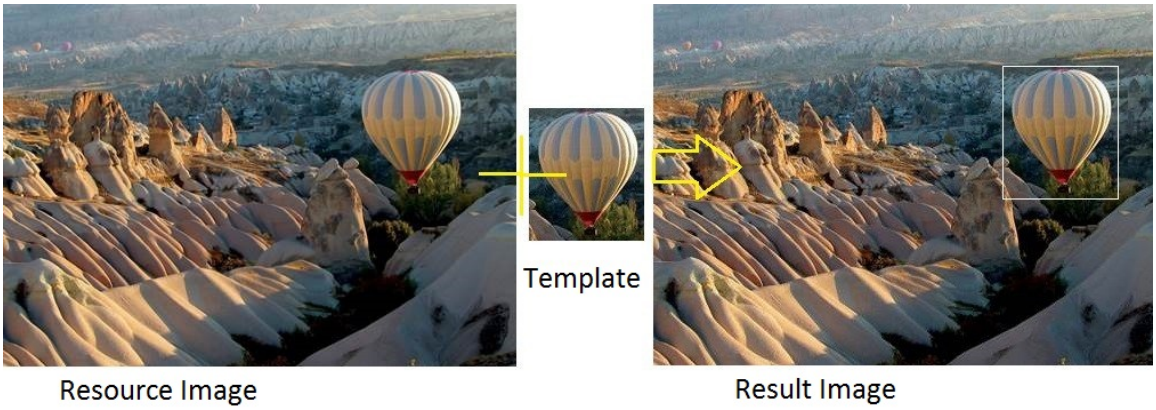
  V roku 2002 boli tieto funkcie rozšírené Lienhartom a Maydtom[21], aby bolo možné lepšie zachytiť napríklad ruky pohybujúceho sa človeka.[18]

- **Detektor založený na hranách a sklone** - ďalšou technikou je výpočet nezrovnalostí vo funkcii jasu obrazu. Gavrila a Philomin[13] spopularizovali systém pre detekciu chodcov založenom na extrahovaných okrajových obrazoch a ich spojením a využitím vzdialenosti skosenia. Neskôr sa to upravilo na systém schopný detekovať chodcov v reálnom čase. [18]

- **Modely** - algoritmom je spojenie priestorovej transformácie s už vypočítanými hierarchickými modelmi. Výhodou je detekcia v reálnom čase, nevýhodou je však zvýšená miera nesprávnych vyhodnotení pri detekcii sledovaných objektov.[18]

Na základe šablóny

Táto technika je založená na princípe nájsť prvky v obraze, ktoré sa zhodujú so šablónou (pozri obrázok 2.12). Princípom detekcie je spárovanie reprezentácie obrázka ako dvojrozmerné pole s hodnotami intenzít so šablónou hľadaného objektu s vhodnou metódou (napríklad Euklidovská vzdialenosť). Algoritmy nie sú vhodné pre sledovanie viacerých objektov. Taktiež nevýhodou je vysoká výpočtová zložitosť. [18, 10]



Obr. 2.12: Na obrázku vľavo sa nachádza vstupný obrázok. V strede sa nachádza šablóna, ktorá je reprezentovaná balónom. Pravý obrázok je výstupom, kde je balón v obrázku detekovaný na základe šablóny[4]

Na základe pohybu

Základom je proces detekovať pohybujúci sa objekt v obraze v závislosti od odchýlky medzi objektom a referenčného rámca pozadia. Dostatočne veľká odchýlka značí pohybujúci sa objekt. Algoritmus Background subtraction (algoritmus oddeľujúci objekty v popredí, čiže hľadané objekty od potenciálne rušivého pozadia) využívajúci fixnú kameru pre detekciu objektov je najefektívnejší spôsob oddelenia objektov v popredí od pozadia.[18, 10]

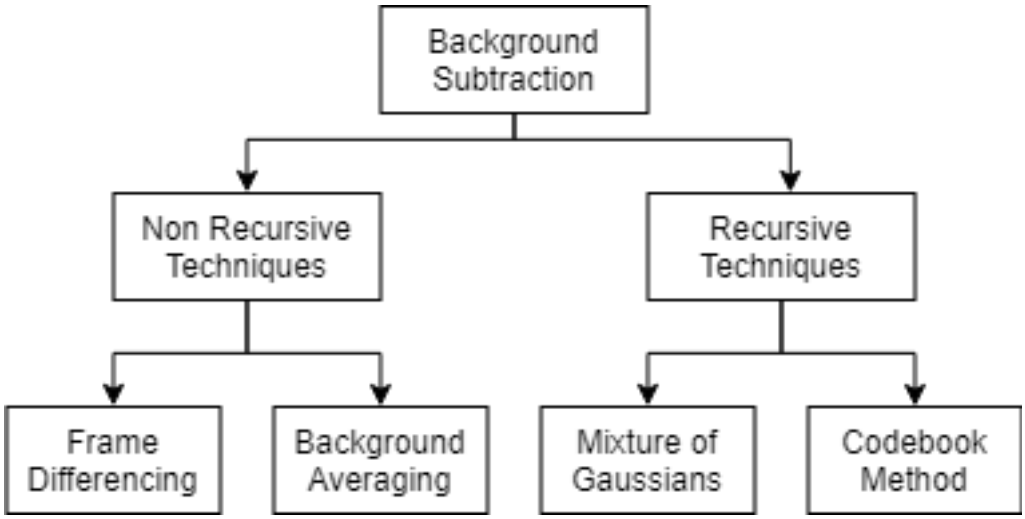
Tento princíp môžme rozdeliť do štyroch bodov:

- Algoritmus Background subtraction by mal segmentovať hľadané objekty v momente zobrazenia v scéne
- Pixely, ktoré spĺňajú kritériá sú určené za pozadie a teda ignorované
- Model pozadia musí byť schopný sa postupne prispôsobovať aj náhlym zmenám v pozadí
- Model pozadia musí byť schopný vziať do úvahy zmeny v odlišných priestorových mierkach

Algoritmus Background subtraction pozostáva zo štyroch hlavných krokov: predspracovávanie, modelovanie pozadia, detekcia objektov na popredí a overovanie dát. Toto je spravené uložením modelu pozadia a lokalizáciou oblastí v novom snímku, ktorý má dostatočný rozdiel od pozadia vzhľadom na hraničnú hodnotu.[18, 10] Tento algoritmus však

má niekoľko obmezení, ktorými sú - použitie len statickej kamery, neschopnosť odlíšiť ľudí v skupine, alebo od ďalších pohybujúcich sa objektov, napríklad áut.[16]

Algoritmus Background subtraction môže byť rozdelený do dvoch kategórií - rekurzívne a nerekurzívne, ako je znázornené na obrázku 2.13. Najjednoduchšou metódou je rozlišovanie naivného rámca, kde posledný snímok reprezentuje model pozadia.[18]



Obr. 2.13: Princíp algoritmu Background subtraction [10]

Segmentácia sa robí porovnaním rozdielu farebných hodnôt medzi predchádzajúcim a aktuálnym snímkom. Jedná sa o nenáročnú metódu z hľadiska pamäti, ale nefunguje dobre v scénach, kde sa vyskytuje premenlivé pozadie, alebo sa sledovaný objekt prestane pohybovať. Pre druhú kategóriu sa berie do úvahy história snímkov a tým je možné mať dynamickejšie pozadie.[18, 10]

Ďalším prístupom je využitie rekurzívneho učenia pozadia multimodálnymi funkciami hustoty pravdepodobnosti. Hodnota intenzít každého pixela je vyjadrená Gaussovským rozložením a jednoduchou heuristikou sa rozhodne, ktoré intenzity patria pozadiu. Pixely, ktoré sa nebudú zhodovať intenzitou s pozadím, budu označené za nezaujímavú oblasť.[18]

Klasifikačné metódy

V momente ako sa získal objekt na základe niektorej z detekčných metód, musí byť klasifikované, o aký objekt sa jedná. Klasifikátory sú rozdelené do dvoch hlavných kategórií učiacich modelov:[18]

- diskriminačné prístupy, ako SVM, ktoré sa rozhodujú či sa jedná o objekt alebo nie na základe natrénovaných príkladov
- všeobecné prístupy, ako sú grafické modely, kde objekt je vymodelovaný na základe jeho funkcie podmienenej hustoty

Diskriminačné modely

Cieľom je oddeliť triedy vzorov od hraničných bodov. Techniky patriace do tohto modelu sú populárne kvôli vynikajúcemu výkonu automaticky vybrať zodpovedajúce prvky z veľkého množstva tréningových prvkov.[18] Týmito technikami sú:

- **SVM** - najpoužívanejšia technika pre postavenie diskriminačného klasifikátora pre detekciu objektov. Maximalizuje medzeru medzi priestormi, aby vznikol maximálny rozostup medzi hranicami značiacich triedu objektov. Dalal použil lineárne SVM pre klasifikáciu hustoty prvkov spočítanú z HOG. Každá vrstva v SVM kaskáde je trénovaná pre získanie TPR úpravou rozhodovacej plochy SVM v aktuálnej vrstve a odmietaním správne klasifikovaných negatív. Postupom času je náročnejšie správne predpovedať správnu triedu objektu. Pre zlepšenie sa musí zvýšiť počet prvkov spolu so zvyšujúcimi sa vrstvami. Pridaním takzvaného Kernelovho triku sa lineárne SVM rozšíri do nelineárneho. Využitie nelineárnych SVM ako klasifikátorov s polynómovými alebo radiálnymi základnými funkciami pre zmapovanie vstupných prvkov do viacdimenzionálnych priestorov sa prejavilo zvýšenou kvalitou detekcie. Avšak v porovnaní s lineárnymi majú nelineárne vyšší výpočtový výkon a taktiež vyžadujú viac pamäte.[18, 28, 16, 37]

- **Kaskádový AdaBoost** - generujú sa slabé klasifikátory, ktoré sú kombinované s náhodne vybraným klasifikátorom, ktorý minimalizuje chyby v tréningových sadách. Touto kombináciou vznikne silný klasifikátor. Každým ďalším priechodom generovania sa skombinuje slabý klasifikátor s výstupom z predchádzajúceho generovania.[37] Táto metóda je často použitá pri generovaní sérií klasifikátorov, kde každý nasledujúci člen tejto série má silnejší klasifikátor, ktorý má vyšší výkon, rýchle vyhodnotenie a slúži ako základ pre moderné detektory. Pre natrénovanie tohto algoritmu sa používajú vektory prvkov HOG a COV.[18, 28] Viola a Jones vytvorili algoritmus pre zosilňujúce kaskády Haar-like vlnky schopné docieliť vysokú rýchlosť v spracovávaní obrazu na 15 snímkov za sekundu.[18] Taktiež tento algoritmus sa môže skombinovať s ďalšími učiacimi sa algoritmami pre dosiahnute zrýchlenia vo výpočtoch.[16]

- **Neurónové siete** - viacnásobné neurónové vrstvy sú taktiež použité pri vytváraní hraničných bodov medzi triedami. V oblasti detekcie chodcov, viacnásobné neurónové siete sú použité s prispôsobivými funkciami receptívnej oblasti ako sú napríklad nelineárne skryté neurónové siete.[18] Jedná sa o efektívne riešenie v systémoch pre rozpoznávanie ľudí.[37]

Všeobecné prístupy

Učí sa spoločné rozdelenie $p(x, y)$ a využíva Bayesovo pravidlo na výpočet $p(y|x)$, aby sa predpovedalo y, ktoré je pravdepodobne opakom diskriminačného klasifikátora učiaceho sa podmienené rozdelenie pravdepodobnosti $p(x|y)$. Najpoužívanejšie algoritmy sú LDA a klasifikátor Naive Bayes.[18, 16]

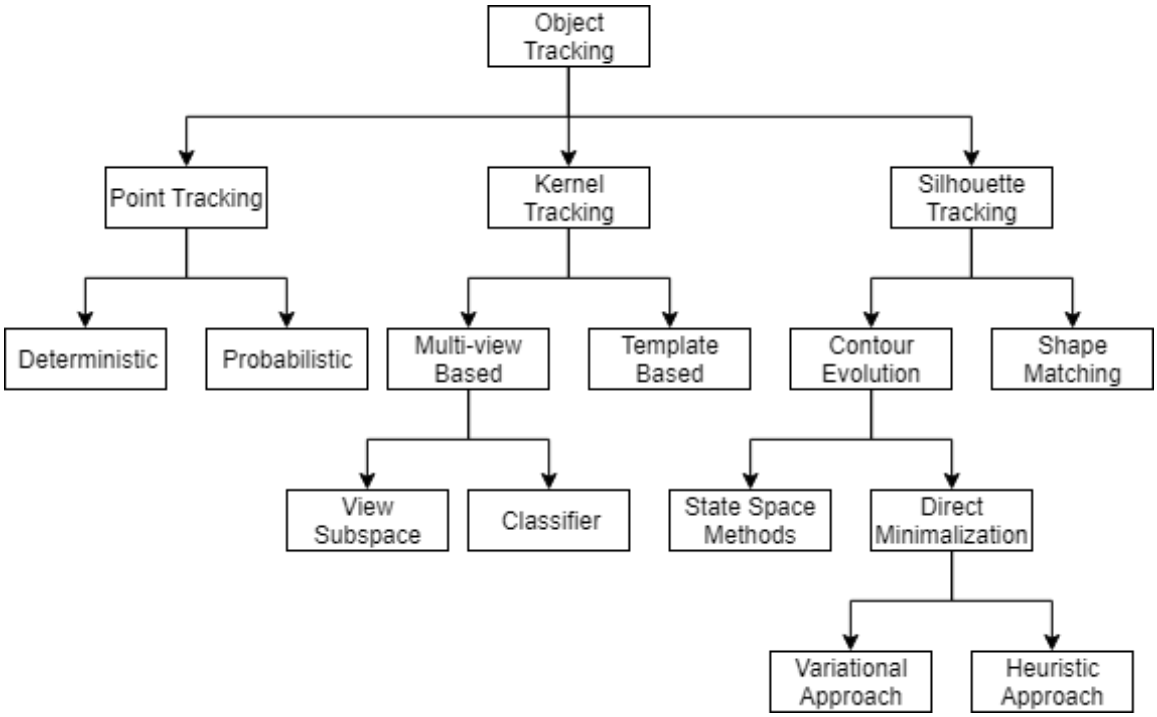
Detekcia viacnásobnej orientácie

Metóda kaskádového Haar klasifikátora bola optimalizovaná pre detekciu objektov s podobnými rotáciami ako predtrénovaný objekt. Objekty môžu byť v rozličných pózach a preto rotácie rozlišujeme na dve kategórie: rotácia v rovine a rotácia mimo roviny. Je možné trénovať jednoduchý klasifikátor pre všetky možnosti rotácie, ale výsledkom by bol slabý a nepresný výpočetný výkon. Riešením je natrénovanie viacerých klasifikátorov s rozostupom napríklad 30° a skombinovaním ich výstupov.[18]

Sledovanie objektov

Sledovanie objektov v reálnom čase sa dá nájsť v rôznych aplikáciách ako monitorovanie dopravy, navigácia vozidiel a rozpoznávanie pohybu objektu. Cieľom je produkovať trajektóriu sledovaného objektu v čase umiestenia jeho polohy vo všetkých snímkoch. Kompletný detekčný systém na základe sledovania môže taktiež predpovedať kolízie s inými objektami. Ako príklad sa môže uviesť vozidlo, ktoré monitoruje okolité objekty, napríklad vozidlá, alebo chodcov. Jedná sa o náročnú úlohu a systém musí byť schopný sa vysporiadať s náhlou zmenou pohybu objektu, vzorom objektu a scény i s možným pohybom samotnej kamery.[18]

Detekcia a vytvorenie zhody objektov medzi snímkami sa vykonáva dvomi spôsobmi: oddelene alebo spoločne. Metóda používajúca oddeľujúci spôsob detekuje a spája objekty naprieč snímkami, zatiaľ čo druhá metóda priebežne aktualizuje polohu objektu a informácie o oblasti s predchádzajúcich snímok. V obrázku 2.14 sa nachádza rozdelenie sledovania objektov do niekoľkých sekcií.[18]



Obr. 2.14: Princíp a rozdelenie sledovacích metód [18]

- **Point Tracking** - algoritmy využívajúce túto metódu reprezentujú objekt ako body alebo skupinu bodov, získavaných z predchádzajúcej polohy a pohybu objektov. Aby bolo možné detekovať objekt v každom snímku, vyžaduje táto metóda ďalší systém pre detekciu. Pre sledovanie objektov, respektíve bodov sa používa Kalmanov filter, ktorý je schopný predpovedať polohu a rýchlosť objektu v trojdimenziálnom priestore. [18, 10]

- **Kernel Tracking** - algoritmus pracuje na princípoch príbuznosti prvkov. Podstatou objektu je jeho tvar a výzor. Stopy objektov sa získavajú výpočtom pohybu podstaty zo snímkov idúcich za sebou. Pohyb objektu je vo všeobecnosti forma parametrického

pohybu alebo výpočtom hustoty prietoku oblasti v nasledujúcich snímkoch. Pre optimalizáciu sa používa procedúra posunu za priemerom. [18, 16, 10]

- **Silhouette Tracking** - v tejto metóde oblasť objektu je vyhodnotená v každom snímku. Informácie získané z objektovej oblasti, ako hustota vzhľadu a tvar modelu, sú použité pre vytvorenie hranových máp. Sledovanie prebieha potom na základe zhody tvaru alebo evolúcie obrysov z modelu objektu. Výzor objektu je zvyčajne modelovaný parametrickou alebo neparametrickou funkciou hustoty. Tvar objektu sa môže vymodelovať z obrysu, ktorý je získaný zo sady obrysov daného objektu v rozličných pozíciách.[18, 10]

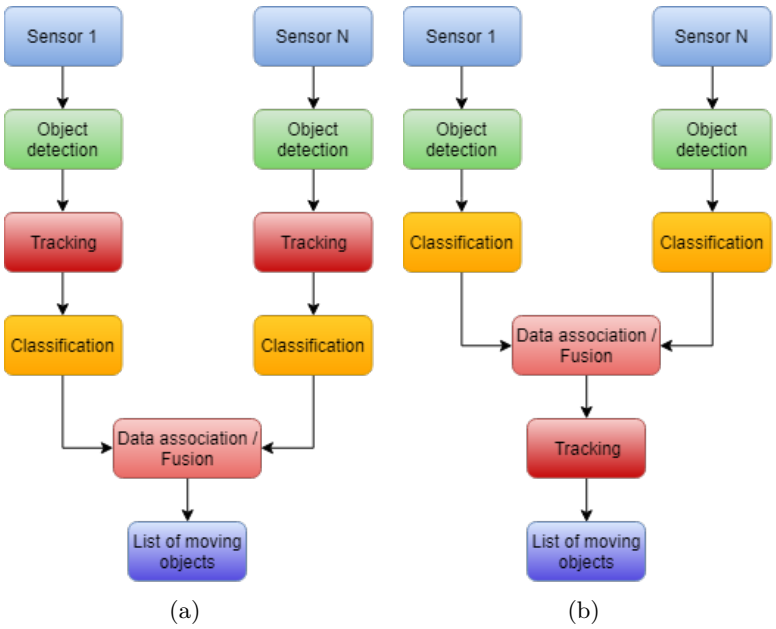
2.3 Fúzia senzorov

Pre ľudí je prirodzené zbieranie a spájanie informácií z odlišných zmyslov, avšak aplikovanie podobného správania je v technike náročné. V podstate sa jedná o kombináciu informácií získaných z odlišných zariadení pre dosiahnutie presnejších výsledkov. V tejto kapitole sú popísané spôsoby fúzie, manipulácia získaných dát, ako ich vizualizácia a uchovávanie. Informácie tu popísané sú však len výňatkom z celkového odvetia týkajúceho sa fúzie odlišných zariadení.

Architektúra fúzie

Na obrázku 2.15 vidíme všeobecnú architektúru fúzie, ktorá sa delí na dve úrovne:

- fúzia založená na sledovaní objektov
- fúzia založená na detekcii objektov



Obr. 2.15: Všeobecný prehľad fúznych architektúr. Vľavo (a) sa nachádza architektúra založená na sledovaní, vpravo (b) architektúra založená na detekcii

Fúzia založená na sledovaní objektov

Jedná sa o klasický prístup fúzie, ktorý berie do úvahy generický počet vstupných senzorov na úrovni sledovania objektov, ako môžme vidieť na obrázku 2.15(a). V tejto architektúre, sledovanie a detekcia objektov sú vykonané pred samotnou fúziou. Po sledovaní objektov sú vyňaté informácie o triede objektu pre zlepšenie reprezentácie objektu v senzorovom zozname trás. Potom je vykonaná fúzia medzi jednotlivými zoznamami trás senzorov. Reprezentácia finálneho objektu obsahuje informácie o triede objektu, ktoré pomáhajú zlepšiť sledovanie pohybujúceho sa objektu a skombinované informácie o objekte získané z jednotlivých senzorov.[11]

Fúzia založená na detekcii objektov

Fúzia, v ktorej sú dostupné informácie pre zlepšenie detekcie a sledovania objektu začleňujú skorej pri aktualizácii reprezentácie objektu. Na obrázku 2.15(b) je znázornená všeobecná architektúra tejto fúzie, ktorá berie do úvahy generický počet vstupných senzorov. Pre každý senzor, detekcia a klasifikácia pohybujúceho sa objektu je vykonaná pred procesom fúzie. Informácie o triede sú vyňaté z detekcií objektu každého senzoru. Sledovanie je vykonané použitím finálneho zoznamu reprezentácie objektu získaných z fúzie. Reprezentácia neobsahuje len kinetické dáta, ale taktiež informácie o geometrii objektu a triedu záujmov objektov.[11]

Fúzia radaru s kamerou

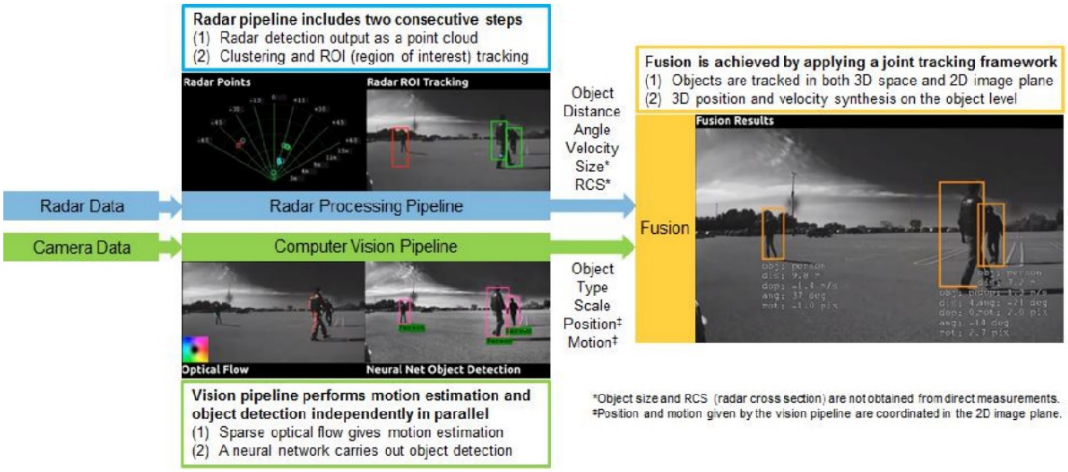
Radar a kamera majú k sebe navzájom doplnkové vlastnosti, čiže je vhodné spraviť fúziu z týchto dvoch senzorov. Ako je ilustrované v tabuľke 2.1, samostatne má každý zo senzorov určité obmedzenie, ktoré však neplatí pri ich fúzii. Benefitom fúzie je to, že sa môžu použiť tie dáta zo senzora, ktoré sú presnejšie.[36]

Perception	**Camera**	**Radar**	**Fusion**
Distance	Fair	Excellent	Excellent
Angle	Excellent	Fair	Excellent
Velocity (Radial)	Limited	Excellent	Excellent
Velocity (Lateral)	Fair	Limited	Excellent
Boundary	Excellent	Limited	Excellent
Obstacle	Fair	Excellent	Excellent
Classification	Excellent	Fair	Excellent
Weather/Lighting/Dirt	Limited	Excellent	Excellent

Tabuľka 2.1: Fúzia kamery a radaru - prehľad úloh[36]

Na obrázku 2.16 vidíme príklad fúzie. V ľavej časti sa nachádzajú výsledky zo spracovania radarových a kamerových dát. Konkrétne, horná časť sa týka radaru, ktorá obsahuje detekčné body objektov a výsledky zo sledovania. V dolnej časti sú výsledky z optického toku a neurónovej siete.

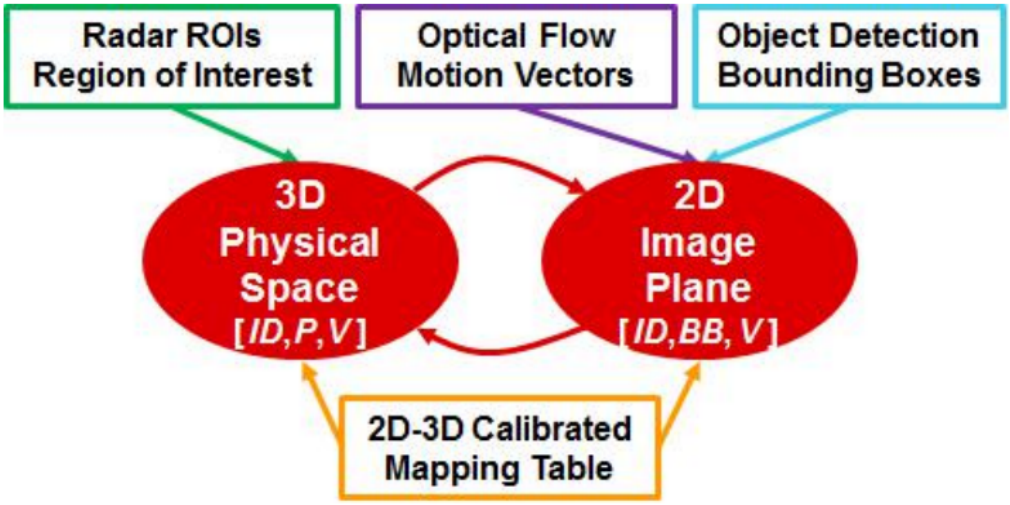
Pravá časť obrázka 2.16 obsahuje výsledky fúzie získané skombinovaním výsledkov z jednotlivých senzorov nachádzajúcich sa v ľavej časti. Na obrázku môžme vidieť zoznam informácii získaných fúziou, ako je typ, vzdialenosť, rýchlosť, uhol a uhlová rýchlosť.[36]



Obr. 2.16: Koncept fúzie senzorov typu kamera a radar.[36]

CRF fúzia

Fúziou sa očakáva, že zmiešaním výstupov z radaru a kamery sa získa presnejšia pozícia a rýchlosť objektov. Hlavným problém je skutočnosť, že výsledky zo senzorov sa nachádzajú v odlišných dimenziách. Radarová oblasť záujmov je definovaná v 3D, kým vektory pohybu a ohraničujúce rámce objektu sú v 2D obrazovej rovine. Podľa [36] sa vytvoril framework (z en., softwarová knižnica, alebo kostra programu, ktorá je použitá ako základný modul nového programu), ktorý sleduje trasy snímaných objektov, sleduje ako v 3D tak aj v 2D obrazovej rovine ako je znázornené na obrázku 2.17.[36]



Obr. 2.17: Koncept frameworku pre fúziu kamery s radarom.[36]

Každý objekt je modelovaný v 3D priestore použitím klasického šesť-parametrového modelu pohybu a 2D obrazovej roviny ako ohraničujúcich rámcov s 2D rýchlosťami. Tento framework zabezpečuje, že oba modely zodpovedajúce vlastným cieľom sa neustále zho-

dujú. Toto je dosiahnuté neustálou aktualizáciou modelov, ktorej mechanizmus je podobný Kalmanovmu filtru.[36]

2.4 Prehľad existujúcich riešení

Systémov zaoberajúcich sa fúziou senzorov nie je veľa. V podstate sa jedná o technológiu, ktorá nie je moc rozšírená a väčšinou sa jedná o teoretickú a testovaciu oblasť. Jedným z existujúcich riešení zaoberajúci sa touto oblasťou je systém ADAS, ktorý tu bude bližšie popísaný, nasledovaný fúznym systémom v robotike.

Systém ADAS

Systém ADAS je definovaný ako súbor inteligentných dopravných systémov, ktoré priamo zasahujú do chodu vozidla. Pomaly každý výrobca automobilov má vlastný systém, avšak v podstate sa stále jedná o ten istý systém. Systém ADAS pre automobilové spoločnosti je videný ako nástroj pre zredukovanie počtu havárií s cieľom pre vytvorenie plne funkčného samostatného vozidla.[34, 8]

Systémy patriace pod technológiu ADAS[34]:

- Asistent núdzového zastavenia
- Automatické núdzové zastavenie
- Adaptívny tempomat
- Detekcia slepého uhla
- Asistent zmeny jazdného pruhu
- Upozornenie na zmenu jazdného pruhu
- Rozpoznávanie dopravných značení
- Asistent pri predbiehaní

Táto technológia je stále vo vývoji a neustále je rozširovaná ďalšími systémami. Spoločnosť NVIDIA prišla s riešením pre klasifikáciu, ako schopnosť odlíšiť jednotlivé vozidlá cestnej premávky, ako sú napríklad vozidlá pohotovostných služieb.[1] V posledných rokoch je snaha o vytvorenie plne nezávislého automobilu, v podstate vozidlo, ktoré by nepotrebovalo pre pohyb osobu. Pre vytvorenie takéhoto vozidla je potrebné spraviť fúziu z jednotlivých senzorov, pretože senzory samotné nie sú schopné splniť všetky požiadavky.[36] Spolu s fúziou sú použité podľa [24] aj predtrénované neurónové siete.

Robotika

Možnosť fúzovať dáta z odlišných zariadení je kľúčová pre systémy zamerané na robotiku. V týchto systémoch nie je dôležitý len počet senzorov, ktoré obsahujú, ale tiež presnosť a úplnosť dát, ktoré pomocou nich sú schopné získať. Rozdielne senzory majú odlišné merítka a práve táto odlišnosť získaných informácií je hodnotná pre presnosť dynamiky a kinematiky autonómnych robotov. Autonómne roboty, ktoré sú dôležitou súčasťou celkového odvetvia robotiky, sú v neustálom vývoji. Dôležité je, aby boli schopné získať čo najpresnejšie dáta z prostredia a spravovať ich v reálnom čase. Presnosť, nutná pre schopnosť robota

bezpečného pohybu, je získavaná koordináciou senzorov. Hlavným prvkom robota je jeho prispôsobivosť nepriaznivým podmienkam. Výzkum robotov stále pokračuje, na základe experimentov na nich vznikla napríklad virtuálna realita.[38, 12, 23]

Kapitola 3

Analýza a návrh riešenia

V tejto kapitole popisujem návrhu systému, ktorým sa zaoberá táto práca. Pri návrhu systému pre fúziu dát z radaru a kamery som v počiatočnej fáze zhodnotil existujúce aplikácie zamerané na podobnú tematiku, aby som zistil, čo poskytujú, čo im chýba a aké problémy riešia. Táto analýza dopomôže pri návrhu systému. Ďalej je v tejto kapitole špecifikovaný určitý problém, ktorý som riešil fúziou a navrhol riešenie tohto problému. Na konci som spísal technické parametre systému, ako aj použité knižnice potrebné pre samotný chod systému.

3.1 Analýza konkrétneho problému

Zadaním tejto práce je vytvoriť systém schopný v reálnom čase detekovať objekty z dvoch senzorov a spojiť ich do jedného výstupu. Existujúce systémy sú navrhnuté pre použitie v automobilových systémoch a preto po preskúmaní týchto existujúcich systémov som sa rozhodol vytvoriť aplikáciu, ktorá by bola schopná detekovať ľudí v miestnosti. Úlohou tejto aplikácie je detekovať ľudí, zistiť počet detekovaných ľudí a do obrazu vložiť informácie týkajúce sa detekovaných ľudí. Konkrétne sa jedná o informácie, ako vzdialenosť detekovaných ľudí od kamery a ich rýchlosť. V obraze som vypísal počet detekovaných ľudí, tento počet je v dvoch tvaroch. Prvá informácia hovorí o počte ľudí, pri ktorých bolo možné vykonať fúzia, v druhej informácii bude celkový počet detekovaných osôb. Výsledkom je teda rozptyl týchto dvoch číselných údajov.

3.2 Zhodnotenie existujúcich systémov

V dnešnej dobe je systémov využívajúcich fúziu viacerých senzorov málo. Takmer všetky patria pod automobilové systémy. Známym je systém ADAS, ktorý je implementovaný a upravený mnohými automobilovými spoločnosťami. Tento systém je však ešte stále vo vývoji, hoci ako jediný má použitie v praxi. Ďalšou vetvou zaoberajúcou sa fúziou je odvetvie robotiky, avšak jedná sa zatiaľ hlavne o vývoj. Ostatné systémy sú zatiaľ založené len na teoretickom a testovacom základe. V tejto sekcii porovnávam systém ADAS, systém robotiky a môj systém. Hoci systémy nie sú určené pre riešenie rovnakého problému, oba používajú fúziu dát a na to sa budem v porovnaní zameriavať. Samotné porovnanie môžte nájsť v tabuľke 3.1.

Systém	Detekcia vozidiel	Detekcia osôb	Uloženie detekcie	Autonómne správanie
ADAS	ÁNO	ÁNO/NIE	NIE	ÁNO
Robotika	ÁNO/NIE	ÁNO/NIE	NIE	ÁNO
Môj systém	NIE	ÁNO	ÁNO	NIE

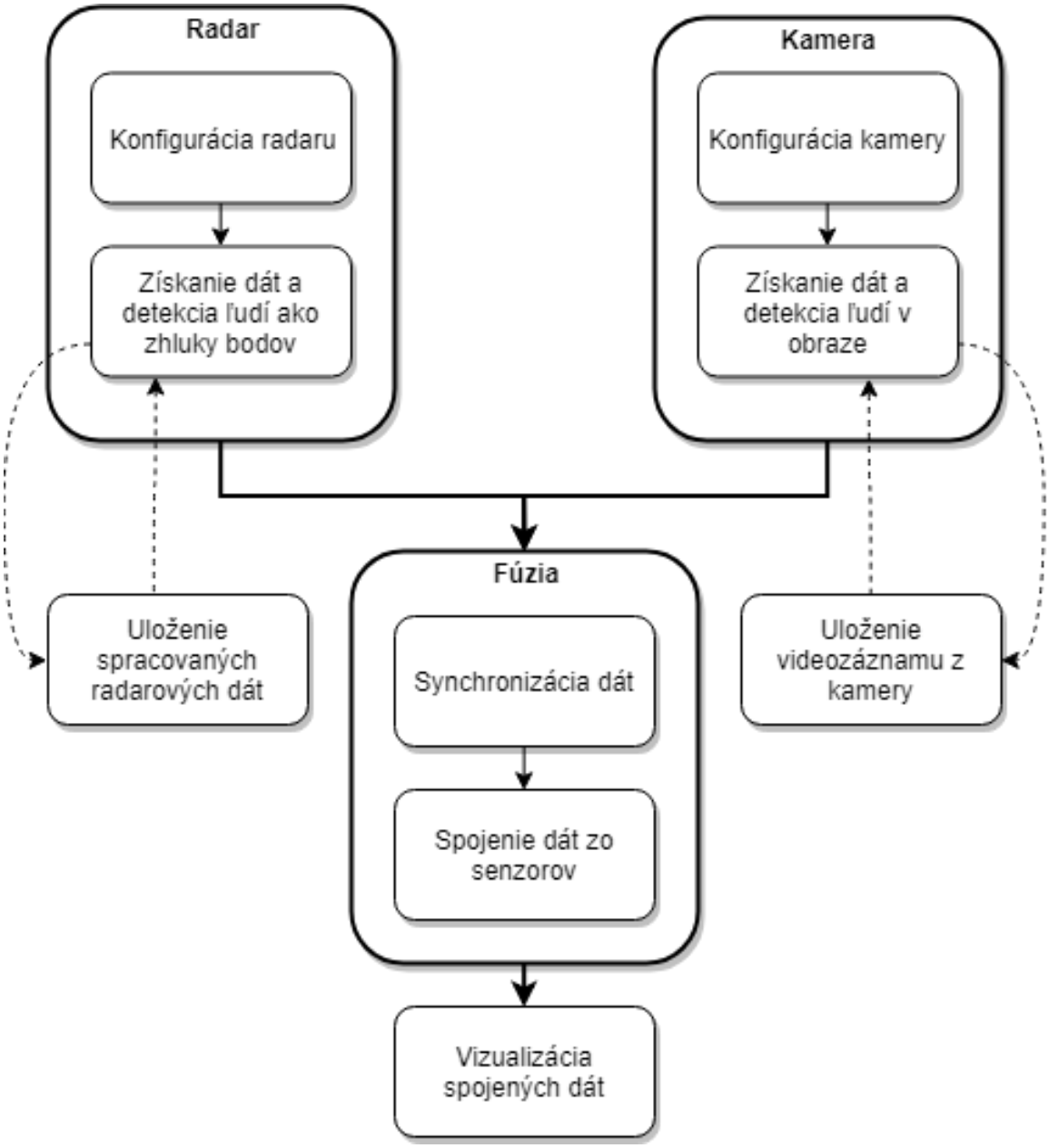
Tabuľka 3.1: Tabuľka porovnávajúca systém ADAS, systém zameraným na robotiku a systém popísaný v tejto bakalárskej práci

Rozdiel systémov je v schopnosti detekcie. Systém, ktorým som sa zaoberal je schopný len detekcie ľudí. Systém ADAS je schopný ako detekcie vozidiel tak aj ľudí, avšak detekcia osôb závisí konkrétne od automobilovej spoločnosti vyvíjajúcej daný systém, niektoré detekciu ľudí nepodporujú. Podobné je to aj u robotiky, keďže daný systém to môže a nemusí mať implementované. Ďalším porovnávacím prvkom je možnosť uloženia záznamov. V aplikácii popisovanej v tejto práci je možné záznamy uložiť a použiť neskôr aj bez použitia senzorov. Na rozdiel od toho systém ADAS, ako aj robotický systém takýmto spôsobom dáta neukladá. Ďalšou výhodou systému ADAS oproti môjmu je podpora autonómneho správania, keďže tieto systémy môžu byť použité v samo jazdiacich autách. V tom je tiež výhoda aj systému pre robotov, keďže sú postavené na základe toho, že podporujú autonómne správanie.

3.3 Návrh riešenia problému

Najprv by mal systém byť schopný spraviť konfiguráciu radaru a kamery, ako aj prepočet ich umiestnenia v priestore voči sebe. Následne spustiť senzory, získavať a spracovávať z nich získané dáta. Po spracovaní dát a ich vzájomnej synchronizácii je potrebné spojiť dáta patriace k rovnakému objektu. Systém by mal byť taktiež schopný spracované dáta uložiť a použiť ich neskôr. Tieto požiadavky som rozdelil do niekoľkých bodov pre uľahčenie návrhu 3.1:

- Práca s radarom
- Práca s kamerou
- Fúzia získaných dát
- Uloženie získaných dát
- Vizualizácia získaných dát



Obr. 3.1: Blokové schéma systému - návrh

Horné dva bloky znázorňujú konfigurovanie senzorov, získavanie a spracovanie dát z týchto senzorov. Výstupom z radaru je zhluk bodov reprezentujúci sledované objekty a výstupom kamery je aktuálny nasnímaný snímok s detekovanými objektami v ňom. Tieto dáta je možné uložiť, čo je reprezentované blokmi pod nimi. Ďalším hlavným blokom je samotná fúzia, ktorá prijíma dáta, synchronizuje ich a spája dáta patriace k sebe. Po tomto spracovaní je možné výstupné dáta zobraziť vo videozázname.

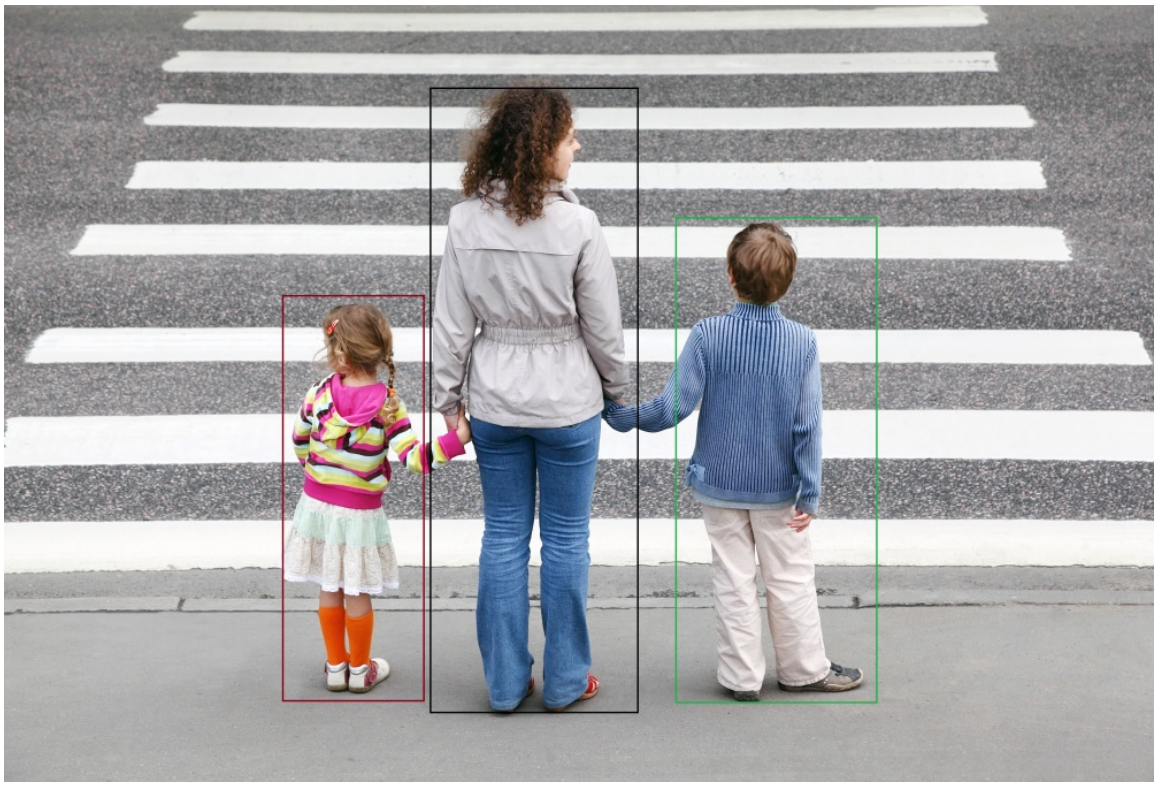
Práca s radarom

Zo súborového systému sa načíta konfigurácia pre nakonfigurovanie senzora. Tiež sú načítané informácie o polohe radaru v priestore. Pre naprogramovanie radaru, aby bol schopný

detekcie ľudí, som sa rozhodol použiť demo aplikáciu People Tracking and Counting od výrobcu radaru Texas Instruments. Analýza dát získaných z radaru je taktiež spravená na základe tejto aplikácie, aj keď pre potreby práce som sa ju rozhodol upraviť, aby môjmu systému vyhovovala. Výsledkom tohto modulu by mali byť objekty reprezentujúce danú osobu v zornom poli radaru. Tieto objekty budú obsahovať informácie o ľuďoch, ako je ich rýchlosť a pozícia v priestore, ako aj čas, kedy boli detekovaní.

Práca s kamerou

Pre prácu s kamerou som sa rozhodol použiť knižnicu OpenCV [26], ktorá už obsahuje funkcie a algoritmy potrebné pre prácu s videozáznamom. Samotný senzor, čiže kamera, musí byť umiestený vo výške aspoň 2,2m a naklonený k zemi, aby bolo možné detekovať aj osoby nachádzajúce sa za sebou. Zo súborového systému sa načíta konfigurácia s informáciami o kamere a jej poloha v priestore. Pri spracovaní obrazu sa vykonáva detekcia ľudí, ako je uvedené na ilustračnom obrázku 3.2. Výstupom tohto modulu je snímok z videozáznamu s detekovanými ľuďmi.



Obr. 3.2: Ilustrácia detekcie ľudí v obraze

Fúzia získaných dát

Po spracovaní dát získaných zo záznamov je potrebné ich spojiť. Ako prvú som spravil synchronizáciu, aby som docielil toho, že sa spájajú dáta získané v rovnakom čase. Po synchronizácii nasledovala ich korekcia, keďže senzory nesnímajú priestor z toho istého miesta. Výstupné dáta sú v rôznych dimenziách, radarové v 3D priestore a vizuálne v 2D obrazovej rovine. Preto som sa rozhodol prispôsobiť dáta z radaru ku kamere. Keďže ka-

mera je umiestnená vo výške v náklone, detekovateľné objekty nachádzajúce sa bližšie ku kamere budú v snímke nižšie ako objekty nachádzajúce sa ďalej. Spojenie závisí od spoločnej osi senzorov, presnej vzdialenosti detekovanej radarom a blízkosti objektov ku kamere zistenej z obrazu. Výstupom sú objekty s informáciami o ich pozícii v obraze ako aj dátami doplnenými z radaru.

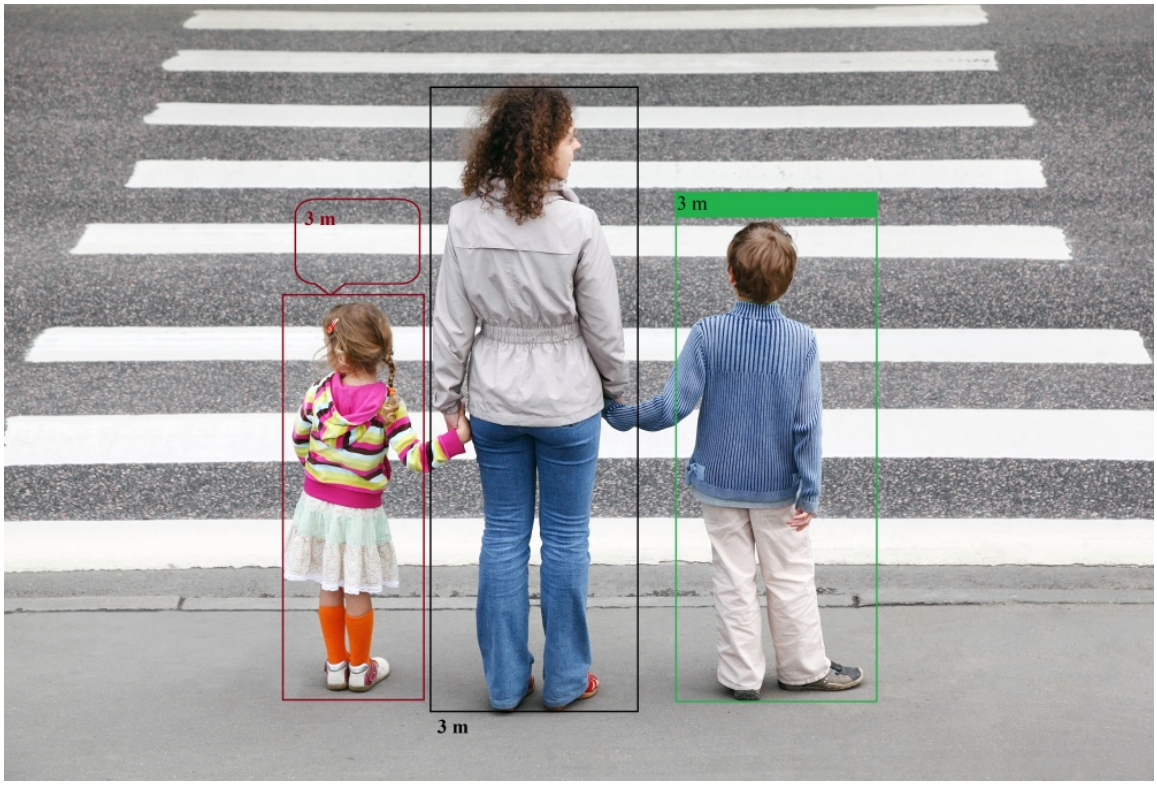
Uloženie získaných dát

Pri ukladaní dát je potrebné ukladať aj čas, kedy sa dané dáta získali. Dáta z kamery ukladám do videozáznamu v určitej perióde. Radarové dáta som sa rozhodol ukladať po vyhodnotení správnosti daného záznamu vo formáte JSON v nasledujúcom tvare:

- **Záznam**
  - Čas
  - Hlavička
  - Zhluk bodov reprezentujúci objekty
- **Záznam**
  - ...
- **...**

Vizualizácia získaných dát

Dáta po vykonaní fúzie sú takto pripravené na vizualizáciu. Vykonám ju taktiež pomocou knižnice OpenCV. Do videozáznamu pridám na základe dát z fúzie informácie o detekcii objektov, ako aj ich vzdialenosť od kamery a rýchlosť. V obraze tak uvidíme počet detekovaných ľudí po fúzii, ako aj celkový počet objektov zachytených v miestnosti senzormi, u ktorých nie je isté či sa jedná naozaj o človeka. To preto, lebo niektoré osoby môžu byť v zákryte a kamera ich nebude schopná detekovať, ale radar áno, prípadne naopak. K takýmto osobám nebudú uvedené žiadne informácie, len budú zahrnuté v celkovom počte osôb nachádzajúcich sa v obraze. Ilustráciu finálneho snímku po fúzii môžte vidieť na obrázku 3.3.



Obr. 3.3: Ilustrácia detekcie ľudí v obraze po vykonaní fúzie

3.4 Technická špecifikácia

V tejto časti sa nachádza tabuľka s technickými parametrami systému.

Technické špecifikácia	
Operačný systém	Linux
Programovací jazyk	Python v3.6
Knižnica pre prácu s kamerou	OpenCV v3.4.3
Radar - výrobca	Texas Instruments
Radar - typ	IWR1642
Radar - naprogramovaná knižnica	Demo Counting and Tracking People
Podporné nástroje	knižnica NumPy

Tabuľka 3.2: Technická špecifikácia aplikácie

Kapitola 4

Implementácia

V tejto kapitole sa venujem popisu implementácie systému pre detekciu ľudí fúziou senzorov radar a kamera. V prvej časti popisujem spôsob implementácie systému rozdeleného do logických celkov. V ďalšej časti sa zaoberám implementáciou jednotlivých modulov a na konci uvádzam spôsob testovania a vyhodnotenie funkčnosti systému.

4.1 Logické celky

Logické rozvrhnutie systému som spravil na základe návrhu z kapitoly 3. Každý logický celok je tvorený množinou programových modulov, ktoré definujú potrebné funkcie pre funkčnosť systému. Implementačné schéma môžme nájsť v obrázku 4.1

Radar

V tejto časti som popísal implementáciu radarového modulu. V úvode je postup konfigurácie radaru, ďalej detekcia ľudí radarom, získavanie dát, ich spracovanie a na záver ich ukladanie.

Konfigurácia radaru

Pri spustení systému sa načíta konfigurácia z konfiguračných súborov. Tie sa delia:

- hlavný súbor obsahujúci všeobecné informácie k systému a v rámci radaru obsahuje informácie o jeho umiestnení v priestore
- radarový konfiguračný súbor obsahujúci konfiguráciu pre samotný senzor

Po úspešnom pripojení k senzoru sa mu posielajú konfiguračné záznamy. Senzor po spracovaní záznamu vracia reťazec `Done`. V prípade, že senzor nepošle daný reťazec, je mu znova poslaný ten istý záznam. Záznam sa skúsi poslať päťkrát, s každou iteráciou je navýšený čas potrebný pre poslanie. V prípade, že záznam nebolo možné poslať, je ukončené posielanie záznamov, aplikácia je korektne ukončená a užívateľ je informovaný o výskyte problému.

Detekcia ľudí ako zhluky bodov a ich spracovanie

Pre naprogramovanie senzora som sa rozhodol použiť demo aplikáciu pre detekciu ľudí od výrobcu radaru. Daná aplikácia využíva senzor pre detekciu ľudí v odrazených vlnách. Taktiež počíta ich rýchlosť, uhol a vzdialenosť. Z týchto údajov vytvára dáta v nasledujúcom tvare[17]:

- Hlavička obsahujúca základné údaje o zázname
- Zoznam TLV objektov, ktorých štruktúra je:
  - hlavička TLV
  - zhluk bodov reprezentujúci objekty
  - zoznam stôp objektov
  - zoznam indexov reprezentujúcich stopy objektov z predchádzajúceho záznamu

Následne poslané dáta začne systém spracovávať. Pri spracovávaní hlavičky som si uložil aktuálny čas, kedy systém začal dané dáta spracovávať. V pravidelných intervaloch, ktorých hodnota je uvedená v globálnom konfiguračnom súbore, sa premazávajú dáta čakajúce na porte a pripravené k spracovaniu. Týmto som rieši synchronizáciu medzi zariadením, na ktorom beží systém, a samotným senzorom. Je to potrebné kvôli tomu, pretože senzor môže zbierať a vysielať dáta rýchlejšie ako ich systém stihne spracovať a pri začatí spracovávaní sa môže odchýliť čas začatia spracovávania od skutočného času zaznamenania daných objektov.

Následne kontrolujem validitu dát, či nedošlo k ich korupcii. Ak áno, v daných záznamoch sa začne hľadať synchronizačný vzor nachádzajúci sa v hlavičke daného záznamu. Po jeho nájdení je možné záznamy ďalej spracovávať. Pri spracovávaní dát získaných zo senzora vytváram objekty reprezentujúce ľudí. Tieto objekty obsahujú informácie o polohe detekovaného objektu v priestore z pohľadu senzora, ako aj jeho rýchlosť. V tomto stave je možné záznamy uložiť, čo je vysvetlené v ďalšej časti, alebo ďalej spracovávať. Takto pripravené objekty sú pripravené pre fúziu s kamerou.

Ukladanie radarových dát

Po overení validity dát získaných zo senzora je ich možné uložiť. Tieto záznamy som sa rozhodol ukladať vo formáte JSON v nasledujúcom tvare:

- Čas začatia spracovávania záznamu / čas detekcie objektov
- Zoznam objektov, kde objekt pozostáva z:
  - pozícia v priestore podľa osi X
  - vzdialenosť od senzora
  - rýchlosť objektu

Časť takto uložených záznamov zo senzoru si môžte pozrieť na obrázku 4.2 . Takto uložené dáta sa dajú načítať pred spracovávaním dát zo senzora.


```
[{"time": 1557692940, "objects": [{"distance": 0.996059238910675, "x": -0.342144250869751, "velocity": 0.8332075907768831},
{"distance": 1.8074289560317993, "x": 1.3368961811065674, "velocity": 0.0}]}]
[{"time": 1557692940, "objects": [{"distance": 0.840190589427948, "x": -0.19959601759910583, "velocity": 0.6581801445499554},
{"distance": 1.5785267353057861, "x": 0.6496613025665283, "velocity": 0.4534294914803166}]}]
[{"time": 1557692941064.2007, "objects": [{"distance": 0.8410111665725708, "x": -0.2787301540374756, "velocity":
1.1126558764163632}, {"distance": 1.587446928024292, "x": 0.6848730444908142, "velocity": 0.20505178922739}]}]
[{"time": 1557692941219.0835, "objects": [{"distance": 0.0, "x": 0.0, "velocity": 0.0}]}]
[{"time": 1557692941, "objects": [{"distance": 1.7310988903045654, "x": 0.574648380279541, "velocity": 0.358107373295916}]}]
[{"time": 1557692941524.4734, "objects": [{"distance": 1.7032867670059204, "x": 0.6136782765388489, "velocity":
0.17192484855234536}]}]
[{"time": 1557692941673.3242, "objects": [{"distance": 1.694977879524231, "x": 0.6302357316017151, "velocity":
0.3136799842243537}]}]
[{"time": 1557692941831.237, "objects": [{"distance": 1.6592786312103271, "x": 0.6811605095863342, "velocity":
```


Obr. 4.2: Ilustrácia uloženia radarových dát

Kamera

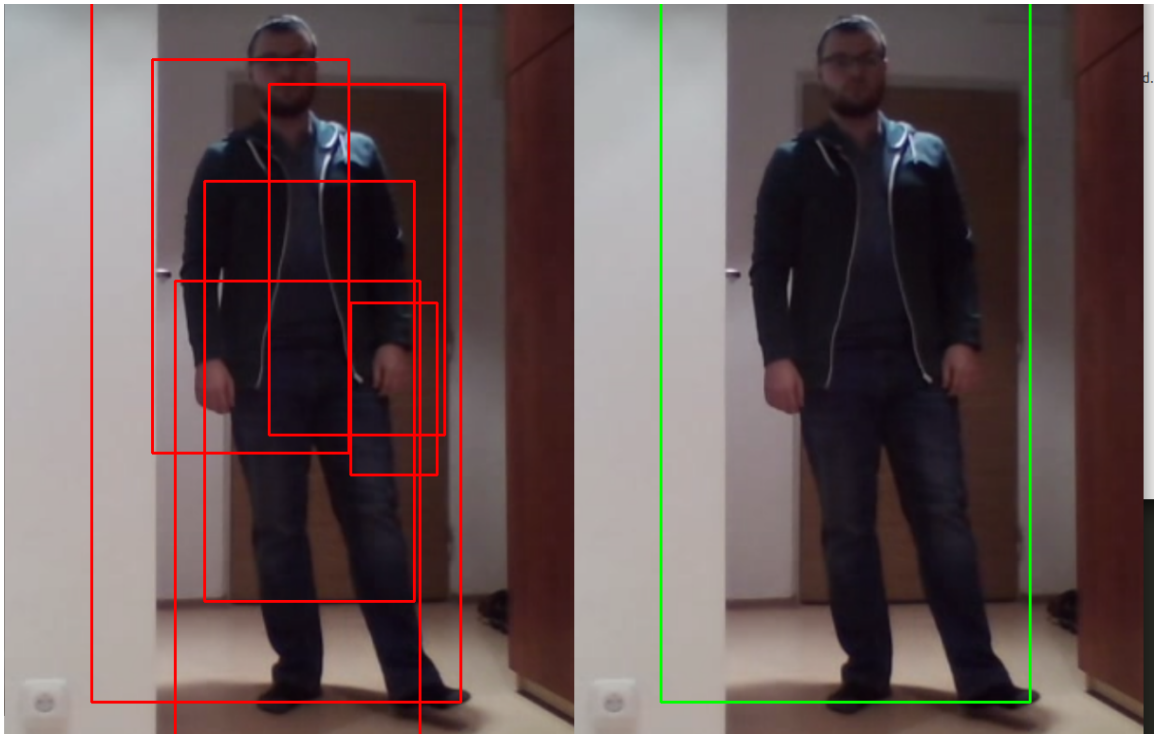
V tejto sekcii sa nachádza popis implementácie detekcie objektov v obraze. V úvode som popísal konfiguráciu senzora a systému s potrebnými údajmi o kamere, ďalej detekciu ľudí v obraze, ich spracovanie a tvar v akom sa výsledky spájajú s dátami z radaru. V závere vysvetľujem spôsob ukladania videozáznamu.

Konfigurácia kamery

Pri spustení systému sa načíta konfigurácia kamery nachádzajúca sa v globálnej konfigurácii. Táto konfigurácia obsahuje informácie o kamere, jej polohu v priestore, ako aj potrebné dáta k úspešnej detekcii.

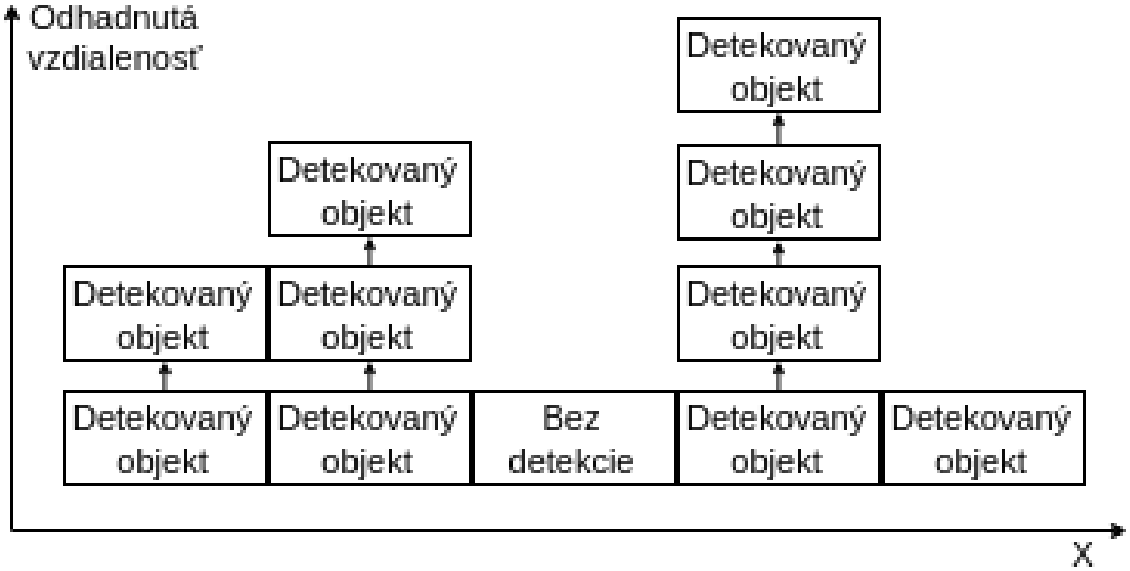
Detekcia ľudí

Pre prácu s kamerou a detekciu ľudí som sa rozhodol použiť voľnú knižnicu OpenCV. V tomto module sa z kamery načítavajú snímky v určitej perióde. Snímka je zmenšená pre urýchlenie detekovania a nad takto upravenou snímkou je pomocou knižnice OpenCV vykonaná detekcia ľudí. Detekcia využíva HOG deskriptor, ktorý je uvedený v časti 2.2 . Výsledkom detekčnej funkcie sú rámce značiace pozíciu osoby v obraze. Rámce predstavujúce tú istú osobu sú spojené do jedného rámca použitím algoritmu Non-maxima suppression (voľne preložené ako maximálne potlačenie, jedná sa o spôsob ako odstrániť prekrývajúce sa rámce). Na obrázku 4.3 môžte vidieť snímok bez použitia a s použitím Non-maxima suppression.



Obr. 4.3: Non-maxima suppresion - na ľavej strane obrázka nebolo použité Non-maxima suppression, kým na pravej strane áno. Môžme vidieť, že prekrývajúce sa rámce sa spojili do jedného rámca.

Z týchto rámcov reprezentujúcich detekovaných ľudí vytváram mapu, ktorej koncept je znázornený na obrázku 4.4. V tejto mape uvádzam detekované objekty, ich pozíciu v obraze, x-ovú súradnicu v priestore, odhadnutú vzdialenosť a pixely detekcie v obraze. Po fúzii sa tu tiež doplnia informácie z radaru, ktorými je rýchlosť a vzdialenosť, ako aj stav, či sa podarilo daný objekt spojiť.



Obr. 4.4: Ilustrácia detekčnej mapy

Pre určenie vzdialenosti som vytvoril riešenie platiace pri použití samostatnej kamery. Keďže kamera je umiestnená vo výške s určitým náklonom, osoby nachádzajúce sa bližšie ku kamere sú v obraze umiestnené nižšie. Týmto spôsobom sa určí odhadnutá vzdialenosť. Napríklad osoba na určitom X indexe najbližšie ku kamere bude v mape prvá, ďalšia s tým istým indexom bude druhá atď. Presné vzdialenosti sú doplnené až po vykonaní fúzie.

Uloženie videozáznamu

V prípade, že je potrebné uložiť záznam, budú pomocou knižnice OpenCV uložené jednotlivé snímky s určitou periódou. Snímky sa ukladajú pred samotnou detekciou a tak pri použití takto nahratého video-záznamu sa musí detekčná časť vykonať nanovo.

Fúzia

V tejto časti sa zaoberám implementáciou fúzie dát zo senzorov typu radar a kamera. Na začiatku som popísal implementáciu synchronizácie týchto dát a následne aj spôsob ich spojenia.

Synchronizácia dát zo senzorov

Dáta získavané zo senzorov nie sú zaznamenané v rovnaký čas a preto je nutná synchronizácia. Pri spracovávaní radaru, ako je spomenuté vyššie, sa vykonáva synchronizácia senzoru typu radar. Tým sa nejedná o synchronizáciu medzi samostatnými senzormi. V globálnom konfiguračnom súbore je uvedená maximálna odchýlka, pri ktorej sa dáta zo senzorov ešte

pokladajú za synchronizované. V prípade, že odchýlka bude vyššia ako je maximálna hranica, k fúzii nedôjde a modul pracujúci s radarovým senzorom dostane rovnaký signál ako pri jeho prvotnej synchronizácii, a to je resetovanie čakajúcich dát na porte, aby sa zaznamenali a okamžite začali spracovávať aktuálne dáta. Vo vizualizácii fúzie je tak zobrazený snímok len s informáciami z kamery, teda osoby, ktoré boli detekované pomocou nej, bez aktuálnych doplňujúcich informácii z radarového senzoru. V prípade, že nebolo možné dáta sfúzovať, budú k nim poskytnuté informácie staré. V globálnom konfiguračnom súbore je možné nastaviť, akú dobu (počet snímok) majú byť zobrazované tieto staré informácie. V prípade, že k synchronizácii dôjde, dáta sa aktualizujú.

Spojenie dát

Po úspešnej synchronizácii som dáta spojil. Z radaru je k dispozícii zoznam objektov obsahujúci ich pozíciu v priestore, hlavne vzdialenosť od radaru a x-ová súradnicu priestoru z pohľadu radaru. Z kamery je k dispozícii aktuálny snímok, rámce reprezentujúce pozíciu osoby v 2D obrazovej rovine snímku a ich predpokladaná vzdialenosť. Keďže kamera a radar sa nachádzajú na odlišných miestach, je potrebná ich korelácia. Pozícia objektov zaznamenaná radarom je prepočítaná, aby zodpovedala pozíciám, ktoré by boli zaznamenané, ak by sa radar nachádzal na tom istom mieste ako kamera. Je to potrebné kvôli tomu, aby mali čo najpodobnejšie zorné pole.

Aby som mohol vykonať fúziu dát, je potrebné nájsť spoločné prvky medzi zaznamenanými dátami zo senzorov. Jedným zo spoločných prvkov v dátach je x-ová súradnica priestora z pohľadu kamery. Ďalším spoločným prvkom je ich vzdialenosť, keďže viem ich presnú vzdialenosť k radaru, prípadne prepočítanú vzdialenosť smerom ku kamere a približnú odhadnutú vzdialenosť získanú z detekcie v obraze. Postupne algoritmus prechádza cez detekované objekty v obraze, na ktoré sa systém snaží napárovať detekované objekty z radaru.

V prípade úspechu sa k danému objektu doplnia informácie získané z radarového senzoru a objekt zo zoznamu objektov detekovaných radarom vylúči. V prípade neúspechu, daný objekt nie je detekovaný ako človek, poskytne sa informácia o detekcii daného objektu, ktorá je v stave *UNKNOWN* (z en. preložiteľné ako neznámy). Takýto objekt nebol započítaný v celkovom počte ľudí, ale započítal sa medzi ciele neurčité. To sú ciele, ktoré zachytil jeden zo senzorov, avšak nebol potvrdený druhým senzorom. Táto situácia môže nastať, keď je osoba v zákryte, čiže detekcia v obraze danú osobu nezachytí, avšak radar áno. Taktiež môže nastať v prostredí, ktoré nemá ideálne podmienky pre detekciu radarom. Jedná sa o prostredie so zvýšeným počtom odrazových plôch, čím radar poskytuje nepresné merania.

Po prejdení objektov detekovaných kamerou sa overí, či nezostali nenapárované objekty zo zoznamu objektov z radaru. Ak áno, dané objekty sa započítajú ako neurčité so stavom *UNKNOWN*. Tiež sa vykoná kontrola, či boli objekty správne napárované. Ak nie, k objektom sa pridá stav *UNKNOWN*. Následne dochádza k vizualizácii fúzie.

V globálnom konfiguračnom súbore je parameter pre periódu vykonávania detekcie objektov v obraze a ich fúzie. Na zrýchlenie systému nie je potrebné vykonávať fúziu s každou snímkou, čím sa celý proces spomalí, ale je možné pre plynulejší chod systému ako aj výslednú vizualizáciu určiť, pre koľkú snímku sa má fúzia vykonať.

Vizualizácia

Po skončení fúzie sa výsledok zobrazí vo videozázname. V prípade, že sa fúzia nevykoná, zobrazí sa snímok s informáciami získanými z kamery. Po vykonaní fúzie sa okrem dete-

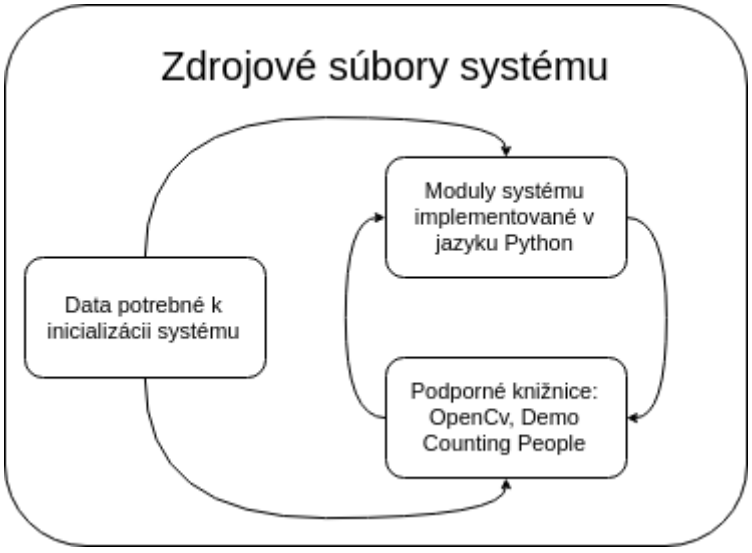
kovaných osôb zobrazia aj informácie o ich vzdialenosti ku kamere a taktiež ich rýchlosť. V snímku je tiež uvedený počet osôb nachádzajúcich sa v zornom poli. Tento počet osôb som uviedol v dvoch tvaroch:

1. tvar - počet osôb detekovaných fúziou
2. tvar - celkový počet detekovaných osôb

Tak je možné vidieť počet osôb, pri ktorých bolo vykonaná fúzia, ako aj možný počet osôb nachádzajúcich sa v miestnosti. Vzniká tak rozptyl skutočného počtu osôb nachádzajúcich sa v sledovanej oblasti.

4.2 Rozvrhnutie modulov systému

V tejto časti uvádzam celkové rozvrhnutie modulov so stručným popisom ich funkcií. Rozvrhnutie modulov je založené na logickom rozvrhnutí systému.



Obr. 4.5: Implementácia - rozvrhnutie modulov a ostatných častí systému

Časť systému starajúca sa o správu konfiguračných súborov a udržiavajúca potrebné dáta pre chod systému spravuje trieda v súbore `Config.py`. Táto trieda je inicializovaná v hlavnom súbore `main.py`. V tomto súbore je taktiež inicializovaná trieda starajúca sa o chod ostatných častí systému nachádzajúca sa v súbore `Manager.py`, ako aj trieda, ktorá rieši vstupné argumenty programu. Táto trieda sa nachádza v súbore `arguments.py`. Časť systému slúžiaca pre komunikáciu s radarovým senzorom, spracovanie radarových dát, ukladanie a načítanie objektov je implementovaná v module **Radar** v súbore `RadarHandler.py`. V tomto module sa taktiež nachádza súbor `RadarStructures.py` obsahujúci štruktúry potrebné pre prácu s radarovými dátami. Funkcie potrebné pre prácu s kamerou sa nachádzajú v `CameraHandler.py`, ktorý je súčasťou modulu **Camera**. V tomto module sa vyskytuje aj súbor `Detect.py` riešiaci detekciu ľudí v obraze. Fúziu dát získaných zo senzorov radar a kamera rieši modul **Fusion**. Samotná fúzia sa rieši v súbore `Fusion.py`. Objekt vytváraný už pri detekcii ľudí v obraze a udržiavajúci dáta z oboch senzorov sa nachádza v `FusionObject.py`. Okrem týchto hlavných častí systému sa tu vyskytujú aj podporné moduly. Tými sú:

- `Logger.py` - modul pre správu výpisov
- `errors.py` - súbor obsahujúci chybové kódy
- `constants.py` - súbor obsahujú konštanty používajúce sa v systéme

4.3 Vyhodnotenie funkčnosti systému

Vyhodnotenie funkčnosti systému ako je tento je možné spraviť viacerými spôsobmi. Buď testujem rýchlosť systému, čiže rýchlosť detekcie a plynulosť videozáznamu, alebo presnosť jednotlivých detekcií a samotnej fúzie. Pre testovanie rýchlosti je dôležité správne zvoliť parametre v globálnom konfiguračnom súbore, hlavne parameter riešiaci periódu vykonávania fúzie, ako aj parametre týkajúce sa detekcie objektov v obraze. Pre testovanie presnosti fúzie je tiež potrebné upraviť parametre v konfiguračnom súbore týkajúceho sa radaru, hlavne hodnoty týkajúce sa detekcie osôb.

4.4 Testovanie a experimenty

Pre testovanie tohto systému boli zvolené spôsoby testovania ohľadom validity výsledkov a rýchlosti systému. Vyhodnotenie funkčnosti prebiehalo v dvoch fázach. Prvá fáza sa týkala presnosti výsledkov oproti reálnym dátam. Druhá fáza sa zameriavala na rýchlosť systému v závislosti od presnosti výsledkov. Obidve fázy sú najviac ovplyvnené detekciou objektov v obraze, hlavne nasledovnými parametrami:

- winStride - krok posuvného okna
- Mierka - udáva počet vrstiev snímkovej pyramídy
- Veľkosť snímku

V oboch fázach bolo použitých 22 uložených záznamov s radarovými dátami, aj dátami z kamery. Tieto záznamy boli zozbierané z troch odlišných prostredí. Prvé prostredie v počte 7 záznamov, druhé pozostávalo z 5 záznamov a posledné z 10 záznamov.

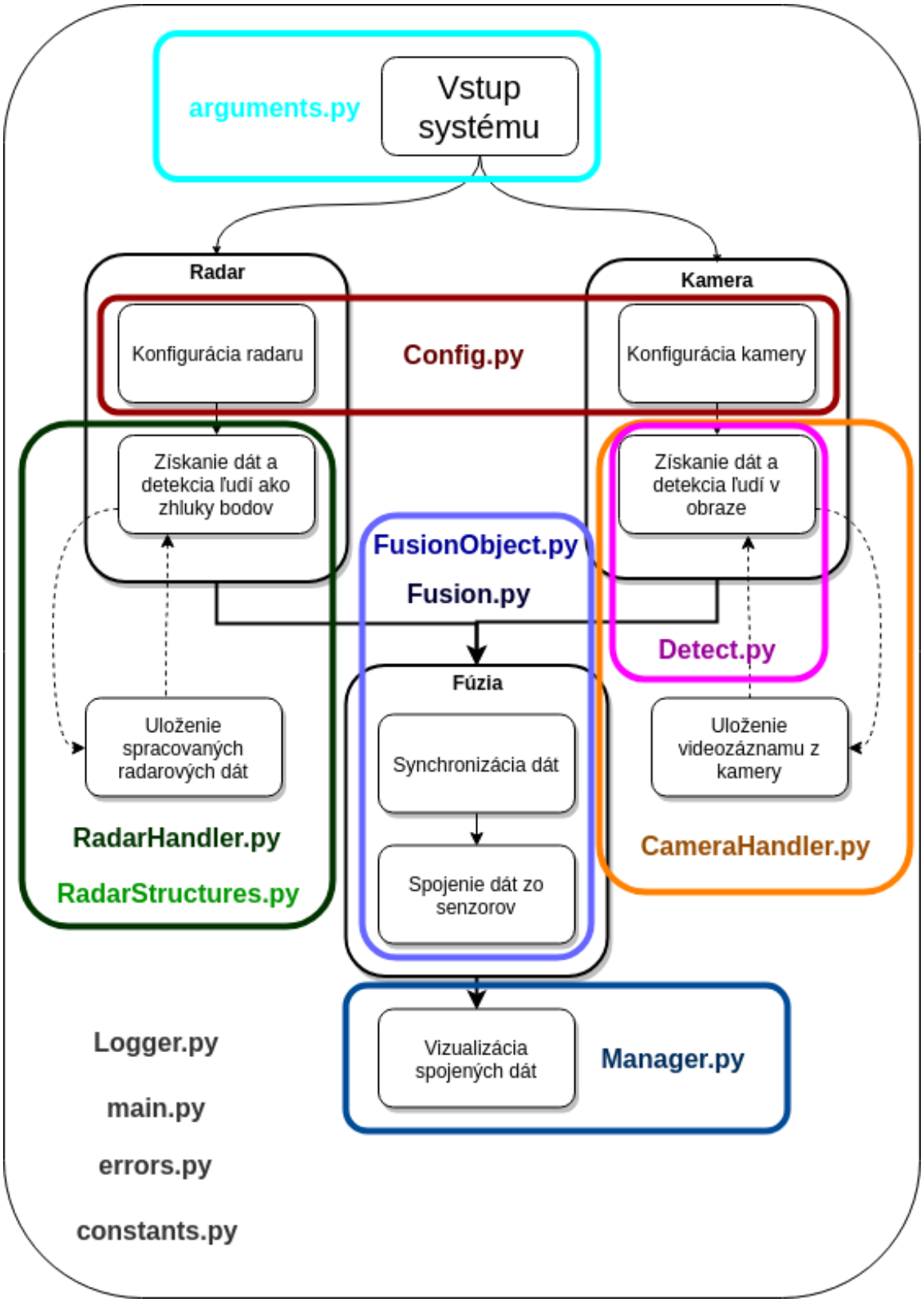
V prvej fázy sa menili jednotlivé parametre pre detekciu, ako aj umiestnenie senzorov. Výsledné dáta sa porovnávali navzájom ako aj s reálnymi dátami, čiže počet ľudí nachádzajúcich sa v zóne záujmu v aktuálnom čase snímania. Druhá fáza pozostávala použitia čo najlepšej kombinácie parametrov pre dosiahnutie optimálnej rýchlosti, pri ktorej by detekcia bola možná.

Výsledné dáta môžte nájsť v tabuľke 4.1.

		Parametre pre detekciu			Počet detekcií				
Prostredie	**Sklonenie senzorov**	**WinStride**	**Mierka**	**Veľkosť**	**Skutočnosť**	**Fúzia**	**Kamera**	**Radar**	**Čas v [ms]**
1	0	(4, 4)	1.05	600	20	13	14	27	765
1	0	(8, 8)	1.05	800	20	8	8	24	390
1	15	(4, 4)	1.4	600	20	13	16	21	260
1	15	(8, 8)	1.2	800	20	7	12	26	144
1	15	(4, 4)	1.5	600	20	15	23	18	303
1	15	(4, 4)	1.05	800	20	18	19	31	978
2	0	(4, 4)	1.05	600	17	8	9	19	780
2	0	(8, 8)	1.05	800	17	3	5	16	380
2	15	(4, 4)	1.4	600	17	9	10	18	258
2	15	(8, 8)	1.2	800	17	3	3	17	160
2	15	(4, 4)	1.5	600	17	7	9	20	300
2	15	(4, 4)	1.05	800	17	12	13	15	950
3	0	(4, 4)	1.05	600	32	22	25	40	850
3	0	(8, 8)	1.05	800	32	19	19	41	470
3	15	(4, 4)	1.4	600	32	23	26	38	273
3	15	(8, 8)	1.2	800	32	18	19	27	162
3	15	(4, 4)	1.5	600	32	31	36	34	363
3	15	(4, 4)	1.05	800	32	28	33	29	1462

Tabuľka 4.1: Porovnanie záznamov po testovaní

Ako je možné vidieť v tabuľke, parametre pre detekovanie objektu v obraze výsledky dosť ovplyvňujú. Stačí drobná zmena a buď detekuje objektov menej, alebo detekuje objekty aj na tých miestach, kde neexistujú. Ďalším prvkom, ktorý dané záznamy ovplyvňuje je sklon senzorov. Toto najviac robí problém kamere, keďže postavy sú za sebou a nedokáže ich detekovať. Aj samotné prostredie má vplyv na výsledky, veľkosť miestnosti, počtu ľudí v nej, ako aj či sa jedná o interiér alebo exteriér. Pri dosiahnutí presnejších výsledkov však stúpa čas potrebný pre detekciu, ako je možné vidieť v tabuľke a teda vo výsledku sa musí nájsť správny pomer medzi schopnosťou detekovať objekty presne a detekovať objekty rýchlo. Pri spustení systému v reálnom čase sa musí obetovať schopnosť presného detekovania, keďže v takom systéme je dôležitá najmä rýchlosť.



Obr. 4.1: Implementačná schéma systému

Kapitola 5

Záver

Cieľom tejto práce bolo vytvoriť systém schopný fúzie z dvoch senzorov typu radar a kamera, ktorý by užívateľovi pomohol zistiť zaplnenosť miestnosti. Tento cieľ bol splnený.

Najprv bolo potrebné preštudovať literatúru týkajúcu sa detekcie ľudí pre oba senzory, ako aj spôsoby fúzie dát získaných z rôznych senzorov. Potom som skúmal už existujúce riešenia. Na základe získaných informácii som sa rozhodol riešiť konkrétny problém, ktorým je detekcia ľudí v miestnosti a zistenie ich počtu, ako aj použitie vhodných knižníc, ktoré dopomôžu k dosiahnutiu riešenia daného problému. Jedným z problémov pri riešení bola synchronizácia senzorov, keďže pracujú nezávisle od seba a dáta z nich získané nie sú zaznamenané v rovnakom čase. Riešením bolo premazávať port v pravidelných intervaloch, cez ktorý bol pripojený radar. V druhej časti synchronizácie som sa zameral na určenie maximálnej časovej odchýlky, ktorá mohla nastať medzi jednotlivými dátami a v prípade jej prekročeniu prerušiť fúziu a znova resetovať port komunikujúci s radarom. Druhým problémom bola samotná fúzia. Výstupné dáta z radaru a kamery sú v odlišných dimenziách a tak hľadal spôsob, ako nájsť dostatok spoločných prvkov pre vykonanie fúzie. Prvým spoločným prvkom bola x-ová súradnica priestoru, druhým približná vzdialenosť získaná z kamery pomocou odhadu blízkosti k nej a presná vzdialenosť detekovaná radarom.

Výsledkom je konzolová aplikácia schopná pripojiť sa k senzorom typu radar a kamera. Je schopná detekcie ľudí pomocou týchto senzorov a fúzie z nich spracovaných výstupných dát. Taktiež umožňuje uloženie dát získaných zo senzorov.

Vyhodnotenie funkčnosti som spravil na uložených záznamoch z kamery a radaru, kde som skúšal presnosť detekcie a aj jej rýchlosť. Vo výsledku záleží na správnom nakonfigurovaní systémov, keďže každé prostredie je iné a senzory je potrebné prispôsobiť. Taktiež je dobré si premyslieť, či systém má byť za behu schopný detekcie v reálnom čase, alebo môže vykonávať detekciu a fúziu len raz začas. Toto rozhodnutie je potrebné spraviť preto, aby sa senzory dobre nakonfigurovali. Presné detekovanie je časovo náročné a teda nie je vhodné pre systém bežiaci v reálnom čase. Dobré je pri spustení systému si dať záležať na konfigurácii a nájsť optimálne parametre pre danú situáciu.

V priebehu práce som sa dozvedel nové poznatky o senzoroch, hlavne typu radar a kamera, ako aj o spracovaní radarového signálu a detekcie objektov v ňom. Taktiež som si rozšíril vedomosti o spracovaní obrazu a detekcii objektov v ňom. Zaujímavosťou pre mňa bola samotná fúzia, keďže ju môžme prirovnať k tomu, ako svet vnímajú živé organizmy. Pri implementácii som sa zlepšil v programovacom jazyku Python, a preto som si ho aj vybral pri tvorbe tohto systému.

V budúcnosti by som sa zameral na rozšírenie aplikácie o grafické užívateľské rozhranie pre uľahčenie práce so systémom. Ďalej by som navrhol pripraviť systém pre pluginy

(drobný program rozširujúci funkcionalitu hlavného systému), pomocou ktorých by bolo možné rozširovať systém o schopnosť detekcie ďalších objektov.

Literatúra

[1] ADVANCED DRIVER ASSISTANCE SYSTEMS. [Online, visited 2019-5-6].
URL https://www.nvidia.com/en-us/self-driving-cars/adas/

[2] Harris Corner Detection. 2018, [Online, visited 2019-4-4].
URL https://docs.opencv.org/3.4.2/dc/d0d/tutorial_py_features_harris.html

[3] IWR1642 single-chip 76-GHz to 81-GHz mmWave sensor integrating DSP and MCU evaluation module. [Online, visited 2019-5-12].
URL http://www.ti.com/tool/IWR1642BOOST

[4] OpenCV: Template matching with java. [Online, visited 2019-3-31].
URL https://riptutorial.com/opencv/example/22915/template-matching-with-java

[5] OpenSIFT: An Open-Soure library. [Online, visited 2019-3-29].
URL https://robwhess.github.io/opensift/

[6] Radar. 2007, [Online, visited 2019-3-14].
URL https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/r/Radar.htm

[7] RADAR-Basics, Types & Applications. [Online, visited 2019-3-20].
URL https://www.elprocus.com/radar-basics-types-and-applications/

[8] ZÁKLADNÍ POJMY V OBLASTI DOPRAVNÍCH INFORMAČNÍCH SYSTÉMŮ. [Online, visited 2019-5-6].
URL http://www.adas.upol.cz/o-adas.html

[9] Assirati, L.; Silva, N. R.; Berton, L.; aj.: Performing edge detection by Difference of Gaussians using q-Gaussian kernels. *Journal of Physics: Conference Series*, ročník 490, mar 2014: str. 012020, doi:10.1088/1742-6596/490/1/012020.
URL https://iopscience.iop.org/article/10.1088/1742-6596/490/1/012020/pdf

[10] Bandarupalli, V.: *Evaluation of video based pedestrian and vehicle detection algorithms*. Bachelor of technology in electronics and instrumentation engineering, Jawaharlal Nehru Technological University, India, 2007.

[11] Chavez-Garcia, R. O.: *Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments*. Robotics [cs.ro], Université de Grenoble, 2014.
URL https://hal.archives-ouvertes.fr/tel-01082021/document

[12] Ciftcioglu, O.; Bittermann, M.; Sariyildiz, S.: Sensor Data Fusion in Autonomous Robotics. 10 2007, ISBN 0-7695-2882-1, s. 26–26, doi:10.1109/ICICIC.2007.516.

[13] D.M. Gavrila, V. P.: REAL-TIME OBJECT DETECTION FOR "SMART"VEHICLES. [Online, visited 2019-4-4].
URL https://people.eecs.berkeley.edu/~malik/cs294/gavrila99.pdf

[14] Dybedal, J.: *Doppler Radar Speed Measurement Based On A 24 GHz Radar Sensor*. Dizertačná práca, 06 2013, doi:10.13140/RG.2.2.31980.77443.

[15] Ender, J.: 98 Years of the Radar Principle. *The inventor Christian Hülsmeier*, 2002.
URL http://www.design-technology.info/resourcedocuments/Huelsmeyer_EUSAR2002_english.pdf

[16] Hasanova, A.: *Pedestrian detection and Tracking in Urban Context using a Mono-Camera*. Master's thesis, University of Tartu, Institute of Computer Science, 2017, supervisor: Amnir Hadachi, PhD.

[17] Instruments, T.: People Tracking and Counting Reference Design Using mmWave Radar Sensor. [Online, visited 2019-5-6].
URL http://www.ti.com/lit/ug/tidue71c/tidue71c.pdf

[18] Johansson, M. L. S.: *Image processing for pedestrian detection using a high mounted wide-angle camera*. Master of science, Chalmers University of Technology, Department of Computer Science and Engineering, 2014.

[19] Krasňanský, M.: *Radar Signal Processing for Radio Altimeter*. Master's thesis, Brno University of Technology, Faculty of Information Technology, 2017, supervisor prof.Dr. Ing. Pavel Zemčík.

[20] Krzstian Mikolajcyzk, C. S.: Scale & Affine Invariant Interest Point Detectors. *Computer Vision*, 2004.
URL https://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_ijcv2004.pdf

[21] Lienhart, R.; Maydt, J.: An extended set of Haar-like features for rapid object detection. In *Proceedings. International Conference on Image Processing*, ročník 1, Sep. 2002, ISSN 1522-4880, s. I–I, doi:10.1109/ICIP.2002.1038171.

[22] Maršík, L.: *Algoritmy zpracování signálu v FPGA*. Diplomová práce, Brno University of Technology, Faculty of Information Technology, 2010, vedoucí práce doc. Dr. Ing. Pavel Zemčík.

[23] Nagla, K.; Uddin, M.; Singh, D.: Multisensor Data Fusion and Integration for Mobile Robots: A Review. *IAES International Journal of Robotics and Automation (IJRA)*, ročník 3, 09 2014, doi:10.11591/ijra.v3i2.4075.

[24] Naik, A.: The Evolution of ADAS: Testing Systems That Include Cameras, Radar, and Sensor Fusion. [Online, visited 2019-5-6].
URL http://2pe5rtjld2w41m0dy17n5an1-wpengine.netdna-ssl.com/wp-content/uploads/2017/12/10_NI-AESIN-ADAS-Ready_The-Evolution-of-ADAS-AN.pdf

[25] Navneet Dalal, B. T.: Histograms of Oriented Gradients for Human Detection. [Online, visited 2019-4-4].
URL https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

[26] OpenCV: OpenCV library. [Online, visited 2019-5-6].
URL https://opencv.org/

[27] Östman, A.: Detection and Tracking of Human Targets using Ultra-Wideband Radar. 2016.

[28] Premebida, C.: *Pedestrian detection using laser and vision*. Doctor of philosophy, Electrical and computer engineering of Coimbra University, 2014, supervisor: Professor Urbano J. Carreira Nunes.

[29] Scholten, U.: Radar Principle. SkyRadar, 2008-2019, [Online, visited 2019-3-14].
URL http://radarportal.skyradar.com/radar-basics/radar-principle/

[30] Sinha, U.: Fundamentals of Features and Corners, A lot of computer vision algorithms use features as their backbone. But what exactly is a feature? [Online, visited 2019-4-4].
URL http://www.design-technology.info/resourcedocuments/Huelsmeyer_EUSAR2002_english.pdf

[31] Skolnik, M. I.: *Introduction to radar systems*. McGraw-Hill, Inc, druhé vydanie, 1981, ISBN 0-07-057909-1.

[32] Skolnik, M. I.: *Radar Handbook*. McGraw-Hill, Inc, tretie vydanie, 2008, ISBN 978-0-07-148547-0.

[33] Skolnik, M. I.: Radar. 2018, [Online, visited 2019-3-10].
URL https://www.britannica.com/technology/radar/History-of-radar

[34] Slavík, B. M.: *Automatized Analysis of Advanced Driver Assistance Systems Behavior*. Diploma thesis, Czech Technical University in Prague, 2016, supervisor: Ing. Václav Jirovský, Ph.D.

[35] Wolff, C.: Doppler Effect. [Online, visited 2019-3-10].
URL http://www.radartutorial.eu/11.coherent/co06.en.html

[36] Zhong, Z.; Liu, S.; Mathew, M.; aj.: Camera Radar Fusion for Increased Reliability in ADAS Applications. *Electronic Imaging*, ročník 2018, č. 17, 2018: s. 258–1–258–4, ISSN 2470-1173, doi:doi:10.2352/ISSN.2470-1173.2018.17.AVM-258.
URL https://www.ingentaconnect.com/content/ist/ei/2018/00002018/00000017/art00011

[37] Zhou, D.: *Vision-based moving pedestrian recognition from imprecise and uncertain data*. Doctor of philosophy, Université de Technologie de Compiègne, 2014, english, Supervisors: Vincent Frémont, Benjamin Quost.

[38] Özer Çiftçioğlu; Sariyildiz, S.: Data Sensor Fusion for Autonomous Robotics. In *Serial and Parallel Robot Manipulators*, editácia S. Kucuk, kapitola 19, Rijeka: IntechOpen, 2012, doi:10.5772/33139.
URL https://doi.org/10.5772/33139