VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SLEDOVÁNÍ VOJENSKÝCH CÍLŮ VE VIDEOSIGNÁLU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE ONDŘEJ HAMADA
AUTHOR

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SLEDOVÁNÍ VOJENSKÝCH CÍLŮ VE VIDEOSIGNÁLU

TRACKING OF MILITARY OBJECTS IN A VIDEOSIGNAL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ONDŘEJ HAMADA
AUTHOR

VEDOUCÍ PRÁCE — Ing. FILIP ORSÁG, Ph.D.
SUPERVISOR

BRNO 2010

Abstrakt

Práce shrnuje a popisuje nejrozšířenější metody pro detekci cílů ve videosignálu. Speciálně se zaměřuje na využití kernel-based metody s použitím mean-shift algoritmu a metody pro porovnávání vzorů. Obě tyto metody jsou doplněny Kalmanovým filtrem. Výsledek je pak aplikován ke sledování cílů jak v normálním videozáznamu, tak i v záznamu z termokamery. Postup je implementován v C++ s použitím knihnovny OpenCV.

Abstract

First part of this thesis briefly describes the main problems of object tracking in video and methods to deal with them. Main attention is paid towards usage of these methods for target tracking in videos taken by both normal and thermal camera. It propose solution to this problem based on the kernel-based method which uses mean-shift algorhitm and on the template matching. Both tracking methods results are supervised and corrected by Kalman filter. Proposed solution is implemented using C++ with OpenCV library

Klíčová slova

sledování cílů, kalmanův filtr, kernel-based metody, odečítání pozadí, mean-shift algoritmus, porovnávání vzorů, matematická morfologie, segmentace, počítačové vidění, OpenCV

Keywords

object tracking, kernel-based methods, Kalman filter, template matching, mathematical morphology, segmentation, computer vision, OpenCV

Citace

Ondřej Hamada: Sledování vojenských cílů ve videosignálu, bakalářská práce, Brno, FIT VUT v Brně, 2010

Sledování vojenských cílů ve videosignálu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Filipa Orsága, PhD.

. . . . . .
Ondřej Hamada
19.9.2010

Poděkování

Chtěl bych poděkovat Ing. Filipu Orságovi, PhD. za pomoc a rady při řešení této bakalářské práce. Dále bych rád poděkoval Ing. Teodoru Balážovi, Csc. za pořízení a poskytnutí videozáznamů pro testování aplikace.

© Ondřej Hamada, 2010.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod **3**

2 Reprezentace a získávání videa **4**
2.1 Barevné modely . . . 5
2.1.1 RGB . . . 5
2.1.2 HSV/HSL . . . 5
2.2 Kamera a přístroje pro podporu vidění . . . 6
2.2.1 Videokamera . . . 6
2.2.2 Noktovizor a termokamera . . . 7

3 Práce s obrazem **9**
3.1 Transformace jasu a geometrických vlastností . . . 9
3.2 Filtry . . . 10
3.2.1 Lineární filtry . . . 10
3.2.2 Nelineární filtry . . . 11
3.3 Rekonstrukce obrazu . . . 11
3.4 Matematická morfologie . . . 12
3.5 Segmentace . . . 13
3.5.1 Mean-shift segmentace . . . 14

4 Sledování cílů **16**
4.1 Model cíle . . . 16
4.2 Detekce cíle . . . 18
4.2.1 Detektory bodů . . . 18
4.2.2 Klasifikátory . . . 18
4.2.3 Porovnávání vzoru . . . 19
4.2.4 Odečítání pozadí . . . 19
4.3 Sledování cíle . . . 20
4.3.1 Kernel-based metody . . . 20
4.3.2 Kalmanův filtr . . . 22
4.3.3 Částicový filtr . . . 23

5 Implementace a testování **24**
5.1 Inicializace . . . 24
5.2 Sledování . . . 25
5.3 Predikce pozice cíle . . . 26
5.4 Testování . . . 26
5.5 Další vývoj . . . 28

6 Závěr **30**

A Obsah DVD **33**

Kapitola 1

Úvod

Rozvoj výpočetní techniky umožnil prakticky se zaobírat umělou inteligencí. Jednou z jejích disciplín je i počítačové vidění, které se pomocí technických prostředků snaží alespoň částečně napodobit lidské vidění. Počítačové vidění pak nachází uplatnění v široké škále oblastní od medicíny, kde pomáhá zejména při zpracování obrazových dat z rentgenů, mikroskopů nebo angiografů, přes průmysl, v němž pomáhá například při kontrole kvality a odhalování závad ve finálních produktech, až po vojenské technologie.

Právě ve vojenství nachází patrně největší uplatnění. Jedná se hlavně o systémy pro detekci a sledování cílů nebo například navádění raket. V poslední době se také dost uplatňují v řídících systémech vozidel/letadel bez lidské posádky, kde slouží buďto k usnadnění řízení operátorovi nebo jsou přímo součástí autopilota.

Tato bakalářská práce se zabývá metodami pro sledování vojenských cílů v reálném čase. Jedná se o dosti diskutovanou problematiku s mnoha existujícími metodami, bohužel použitelnost každé z nich je silně limitována a lidské oko a mozek stále zůstávají nejdokonalejším a nejspolehlivějším sledovacím systémem. V případě vojenské techniky situaci dále ztěžuje snaha navrhovat vojenskou techniku a výstrojní součástky tak, aby co nejvíce ztížily práci detekčních a sledovacích systémů.

Druhá kapitola krátce pojednává o reprezentaci zachyceného obrazu v počítači a s tím souvisejících problémech. Dále nás seznamuje se zařízeními pro zachycení videa a zmiňuje se i o zařízeních pro podporu vidění jako jsou termokamery a noktovizory.

Ve třetí kapitole jsou shrnuty metody užívané při předzpracování obrazu jejichž vhodné nasazení usnadňuje práci pokročilejších algoritmů. Čtvrtá kapitola pak uvádí nejrozšířenější metody pro detekci, sledování a predikci polohy cíle.

V poslední páté kapitole je popsána zvolená implementace, jsou zde shrnuty a popsány výsledky testování a na závěr uvedeny plány pro budoucí vylepšení.

Kapitola 2

Reprezentace a získávání videa

Video z hlediska výpočetní techniky je chápáno jako sekvence po sobě jdoucích obrazů. Každý z těchto obrazů je obsažen v dvourozměrném signálu, jehož každý jeden vzorek odpovídá jednomu bodu obrazu. Jeho amplituda pak odpovídá hodnotě udávající barvu resp. jas. Matematicky je obraz možné vyjádřit pomocí funkce dvou proměnných:

$$I(x, y): \ \langle 0, X) \ \times \langle 0, Y) \rightarrow \Re, \tag{2.1}$$

kde X a Y definují velikost obrazu.

To by ovšem znamenalo, že obraz je reprezentován funkcí se spojitým definičním oborem, což je pro počítačové zpracování nepraktické. Proto se předpokládá dodržení Nyquist-Shannonova vzorkovacího teorému[17], podle něhož lze funkci reprezentovat pomocí diskrétních vzorků, pokud je maximální prostorová frekvence v obrazu menší než polovina vzorkovací frekvence. Výsledkem je pak následující diskrétní funkce:

$$I_d(x, y): \ \aleph \times \aleph \rightarrow \Re, \tag{2.2}$$

Definiční obor této funkce se dá lehce ohraničit a je pak možné snadno zpracovávat pouze vybranou část obrazu. Obor funkčních hodnot, který udává výslednou barvu resp. jas bodu, se aproximuje pomocí reálných čísel. V mnoha případech(včetně této práce) je dostatečnou reprezentace pomocí celých čísel, pro kterou stačí použití 8 bitových hodnot bez znaménka, což představuje rozsah hodnot v intervalu $\langle 0, 255\rangle$. Pro zobrazení většího počtu barev a jejich odstínů byly vyvinuty vícevrstvé barevné modely(2.1), které jsou praktičtější z hlediska nasazení v technických zařízeních.

Snímací zařízení zpravidla dodávají jednotlivé po sobě jdoucí snímky už digitalizované, v případě barevného záznamu nejčastěji s využitím barevného modelu RGB(2.1.1). Na takto získané snímky jsou následně aplikovány metody zpracování obrazu. Zpracování obrazu je podmnožinou zpracování signálu, pro níž je vstupem obraz jako například fotografie nebo jeden ze snímků videa. Výstupem může být buďto opět obraz nebo data určující některou z vlastností vstupního obrazu.

Problematika barevných modelů i zpracování obrazu je velmi rozsáhlá, proto jsou v práci zmíněny pouze stručně a podrobnější informace se nalézají v použitých zdrojích: [7, 3, 8, 18, 20].

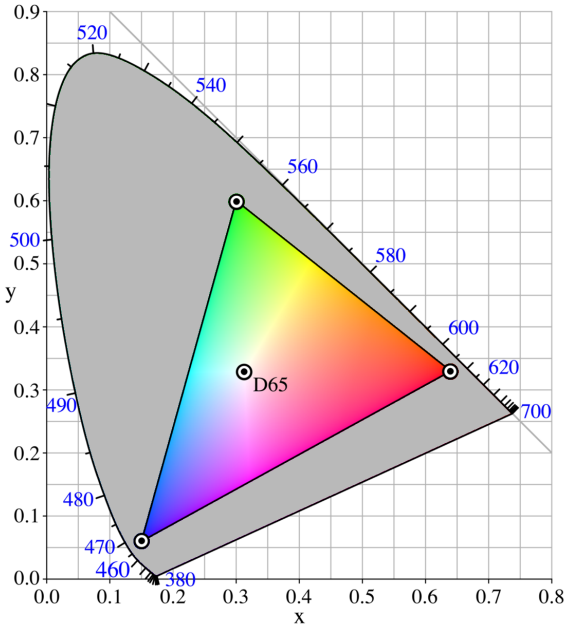
2.1 Barevné modely

Barevné modely jsou abstraktními matematickými modely, které definují způsob, jak pomocí číselných n-tic vyjádřit barvu. Barva je pouze vjemem, který je způsoben světlem určité vlnové délky dopadajícím na sítnici lidského oka. Sítnice je k vnímání barev vybavena čípky, kterých jsou tři druhy. Každý druh vnímá pouze barvu určité vlnové délky a tyto barvy zhruba odpovídají modré, zelené a červené.

V dnešní výpočetní technice se můžeme setkat s několika různými modely. Ve většině z nich je k vyjádření barvy zapotřebí trojice nebo čtveřice hodnot. Jednotlivé modely jsou mezi sebou konvertibilní, liší se však dosažitelnou oblastí barev – takzvaným gamutem. Proto při převodu krajních hodnot může dojít ke ztrátě informací.

2.1.1 RGB

RGB model je jedním z nejrozšířenějších. Jeho vznik je spojen s barevnou televizí používající katodové trubice. Základními barvami je červená, zelená a modrá. Výsledná barva a jas je v RGB modelu tvořena pomocí aditivního míchání barev. Díky jeho snadnému použití ve výpočetní technice je tento model používán jako standardní pro vstupy a výstupy počítačů, videokamer, digitálních fotoaparátů, všech druhů monitorů a televizních obrazovek a mnoha dalších zařízení s výjimkou tiskáren, kde nachází uplatnění substraktivní míchání barev uplatňované v CMYK barevném modelu.



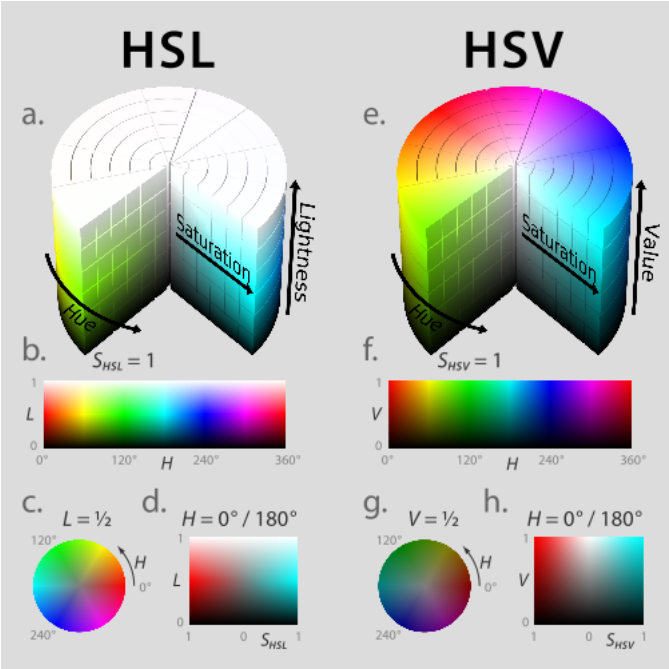
Obrázek 2.1: Gamut RGB [18]

2.1.2 HSV/HSL

Na odlišném principu funguje HSV/HSL[1] model. Barva je zde reprezentována pomocí prvních dvou složek, zatímco třetí určuje její intenzitu. Díky oddělené intenzitě jsou na

[1] zkratka pro Hue, Saturation, Value/Lightness, což znamená odstín, nasycení a světlost

tento model snadněji aplikovatelné algoritmy pro zpracování obrazu, které u RGB modelu selhávají. Rozdíl mezi HSV a HSL spočívá v jasu čisté barvy. U HSL tento jas odpovídá bílé zatímco u HSV středně šedé.



Obrázek 2.2: 3D modely HSL a HSV [20]

2.2 Kamera a přístroje pro podporu vidění

K získávání videa slouží nejčastěji videokamera, jejíž princip bude vysvětlen níže. Standardní vybavení videokamery však nedovoluje uspokojivé zachycení videa za ztížených světelných podmínek. Na druhou stranu vojenské systémy kladou požadavek na schopnost operovat za jakýchkoliv podmínek. K tomu slouží přístroje pro podporu vidění, jmenovitě se jedná o noktovizor nebo termovizi. Každý z těchto přístrojů má svá specifika ovlivňující jeho výstupy. Informace k těmto zařízením pochází z [11, 8, 13, 16, 15, 14]

2.2.1 Videokamera

Jedná se o zařízení využívající fotoelektrického nebo fotovoltaického jevu. S fotoelektrickým jevem pracují hlavně fotonásobiče a vakuové tuby TV kamer. S vynálezem polovodičů se začalo prosazovat použití fotovoltaického jevu. Při něm způsobují přicházející fotony uvolňování elektronů z valenční vrstvy a tím dochází ke změně vodivosti. Uvolněné elektrony se stávají zdrojem elektrického napětí, které je úměrné množství přicházejících fotonů. Konkrétní součástkou v kamerách, převádějící zachycenou energii na obraz jsou CCD nebo CMOS senzory.

CCD senzor sestává z prvků tvořených fotodiodou a tranzistorem. Foton dopadající na fotodiodu uvolní elektron z křemíkové mřížky, čímž dojde ke vzniku elektrického náboje, který je hromaděn v kondenzátoru. Nahromaděný náboj pak odpovídá intenzitě světla a délce jeho záření. Jednotlivé prvku jsou pak poskládány do mřížky bodů podobné matici.

Rozvoj polovodičů umožnil výrobu senzorů s maticovým uspořádáním na s využitím technologie CMOS. Výhodou kamer s CMOS čipem je větší rozsah rozlišitelných intenzit, rychlejší snímání a možnost přístupu k individuálním bodům obrazu. Naopak nevýhodou je větší šum.

Světlocitlivé senzory jsou monochromatické, proto pro získání barevného obrazu musíme použít buďto barevný filtr nebo rozdělit zachycené světlo na jednotlivá spektra. Kamery pak dělíme podle výstupu na analogové a digitální. Pro potřeby sledování cílů jsou užitečnější digitální kamery, protože netrpí rušením při přenosu ani chvěním obrazu. Navíc ukládají digitalizovaný obraz se kterým pak můžeme přímo pracovat bez nutnosti A-D převodu.

2.2.2 Noktovizor a termokamera

Noktovizory umožňují vidění za snížených světelných podmínek, jak název napovídá, používají se hlavně v noci. Jejich počátky se datují do druhé poloviny 30.let 20.století. Další zlomové období následovalo během vietnamské války, masově se však začaly rozšiřovat až v 80. a 90.letech. Existuje několik hlavních metod.

První využívá zintenzivňovač světla. Jeho čočky koncentrují přicházející fotony na fotokatodu. Energie fotonů způsobuje uvolňování elektronů, které jsou dále urychlovány elektrickým polem, narážejí do stěn a tím uvolňují další elektrony. Nakonec elektrony dopadají na fosforové stínítko, přičemž jejich energie způsobí, že se fosfor rozsvítí. Ve většině noktovizorů se používá zelený fosfor, protože lidské oko nejlépe vnímá právě odstíny zelené. Nevýhodou tohoto typu noktovizoru je nutnost alespoň minimálního osvětlení. Navíc kvůli používání fosforového stínítka hrozí poškození přístroje při sledování příliš světlých ploch.

Druhá metoda používá již zmíněné CCD senzory. Jejich čipy jsou však doplněny o zesilovač, který násobí energii dopadajících fotonů. Při práci se ale zahřívají, což snižuje jejich schopnosti. Je tedy nutné je chladit což sebou nese zvýšené energetické nároky. Naopak mezi jejich výhody patří velmi dobré schopnosti s vypořádáním se s minimálním osvětlením(i když přítomnost jisté dávky okolního světla je stále nutností), menší náchylnost k poškození sledováním příliš světlých objektů a rychlost zpracovávání obrazu.

Výstupem noktovizorů s výše zmíněnými metodami bývá dvojrozměrný monochromatický obraz – buďto černobílý nebo v odstínech zelené. Světlost ploch cíle je úměrná jejich světlosti ve skutečnosti, proto při použití kvalitního maskování nebo při pozorování na větší vzdálenosti dochází ke snižování kontrastu cíle s okolím.

Obrázek 2.3: Pohled přes noktovizor [23]

Poněkud jinou cestou se vydávají termokamery. Vychází z myšlenky, že každý objekt vyzařuje infračervenou energii úměrnou své teplotě(cíle mají zpravidla větší teplotu než okolí). Termokamery toto infračervené záření zachycují a převádí na obraz. Díky tomu jsou absolutně nezávislé na okolním světle a umožňují tedy vidět za ztížených podmínek,

dokonce i přes kouř. Nevýhodou těchto zařízení je, že samotné vyzařují nějaké teplo, které může zkreslovat výsledky. Existují verze s chlazením, které jsou pochopitelně přesnější, i bez. V praxi se používají obě. Chlazení je však velmi nákladné, energeticky náročné a rozměrné, což omezuje jeho mobilitu. Vyskytuje se ještě verze s infračerveným ozařovačem, který se používá v nejlevnějších modelech. Pro vojenské využití ale nepřipadá úvahu, protože při ozařování cíle by došlo k prozrazení polohy pozorovatele.

Výstupem termokamer je zpravidla dvojrozměrný černobílý obraz, kde světlejší barva reprezentuje vetší teplotu. U některých zařízení však může být užito takzvaných "falešných barev"pro snazší odlišení teplot objektů. Při pozorování na větší vzdálenosti bohužel také klesá kontrast mezi cílem a pozadím.

Obrázek 2.4: Pohled termokamery, bíla barva ukazuje místa s největší teplotou [10]

Obrázek 2.5: Pohled termokamery s využitím falešných barev [10]

Kapitola 3

Práce s obrazem

Práce s obrazem sestává ze dvou hlavních fází. V první fázi nasazujeme metody, které pracují na nejnižší úrovni abstrakce. Jedná se o metody náležející do souboru metod zpracování obrazu vycházejících z metod po zpracování signálů. Vstupní obraz totiž bývá mnohdy zatížen chybou vzniklou nedokonalostí snímacího zařízení nebo přenosové soustavy a právě k potlačení těchto jevů se metody zpracování obrazu nejčastěji užívají. Druhým velmi častým účelem je snaha o marginalizaci nebo naopak redukci určitých informací v obraze obsažených. Nejčastěji užívanými metodami jsou filtry, různé detektory hran, metody pro transformaci jasu a geometrických vlastností nebo metody pro redukci šumu a rekonstrukci obrazu.

Ve druhé fázi nastupují pokročilejší metody spadající do oboru počítačového vidění. Jedná se o obor, jehož snahou je naučit počítač vidět. Lidský mozek při zpracování vjemu v podobě obrazu se snaží v tomto obraze nacházet modely, které má uloženy v paměti, a dále s nimi pracovat. Bohužel vytvoření dostatečně univerzálních modelů, jejich následná správa a nakládáni s nimi stále přesahuje možnosti dnešních počítačů. Proto jsme nuceni uchylovat se k zjednodušování modelů, kdy objekt je reprezentován pouze některou z jeho vlastností, zpravidla tou, která je v dané situaci nejmarkantnější. Sem spadají metody jejichž cílem je segmentace, detekce objektů v obraze, celková analýza obrazu a také metody pro sledování cílů. Samotnému sledování cílů coby hlavní náplni této práce je věnována celá příští kapitola.

3.1 Transformace jasu a geometrických vlastností

V ideálním případě nezávisí intenzita zachyceného obrazu na jeho pozici v obraze. Ve skutečnosti však intenzita zachyceného světla snižuje s rostoucí vzdáleností od osy čočky. Stejně tak působí problémy i nerovnoměrné osvětlení snímaného objektu. K minimalizaci tohoto problému slouží metody pro transformaci jasu. Buďto můžeme hodnotu jasu pixelu upravovat na základě jeho původního jasu a pozice v obrazu nebo provést celkovou transformaci jasu na základě vlastností společných pro celý obraz. Příkladem užití druhé zmíněné metody je ekvalizace pomocí histogramu jejímž cílem je vytvoření obrazu s jasem rovnoměrně rozloženým po celé stupnici.

Geometrické transformace slouží k potlačování dalšího problému plynoucího z využívání videokamer. Zachycený snímek je dvourozměrný zatímco všechny objekty v reálném světě jsou trojrozměrné. Geometrická transformace tak přichází ke slovu zejména v případech, kdy chceme srovnat dva různé obrazy téhož objektu. Sestává ze dvou kroků:

- Transformace souřadnic

- Interpolace jasu

Transformace souřadnic slouží k namapování souřadnic pixelů vstupního obrazu na odpovídající pixely výstupního obrazu. Často užívanými jsou metody afinní a bilineární transformace. Výstupní obraz se zpravidla liší od vstupního, proto je nutné určit nový jas pixelů. K tomu slouží metody lineární a bikubické interpolace. Všechny tyto transformační metody jsou blíže popsány v [7, kap. 5.1],[3, kap. 6] a [9].

3.2 Filtry

Filtry jsou jedněmi z nejrošířenějších metod pro zpracování obrazu. Uplatnění nacházejí především při korekci obrazu. Výstup z videokamer a jiných záznamových zařízení může obsahovat šum nebo být rozostřený. Stejně tak může být zapotřebí mezi jednotlivými kroky náročnějších algoritmů například zvýraznit hrany ve snímku. Filtry se dělí do dvou skupin – na lineární a nelineární.

3.2.1 Lineární filtry

Podle definice v [8, s.28] je lineární filtr lineární funkci definovanou nad obrazem, jejímž výstupem je také obraz ve smyslu definice linearity:

$$f(o_1) + f(o_2) = f(o_1 + o_2) \tag{3.1}$$

$$\forall x, y : f(o_1, x, y) + af(o_2, x, y) = f(o_1 + o_2, x, y) \tag{3.2}$$

Typickým příkladem je lineární konvoluční filtr. Podle [8, s.28] je konvolučním filtrem filtr popsaný konvolučním vzorem. Výpočet pak probíhá pixel po pixelu, přičemž každý výsledný pixel se vypočte jako konvoluce okolí původního obrazu odpovídající poloze pixelu ve výstupním obraze s konvolučním vzorem(např. 3.1). Konvolučním vzorem bývá obvykle čtvercová nebo obdélníková plocha, nejčastěji s rozměry 3×3, 5×5 nebo 7×7. Protože pracujeme s diskrétními obrazy, můžeme konvoluci nahradit sumou. Výpočet se provádí podle následujícího vzorce:

$$y_{(i,j)} = \sum_{ii\in I}\sum_{jj\in J} x_{(i+ii,j+jj)} \cdot k_{ii,jj},\ \ I = \{-1,0,1\},\ \ J = \{-1,0,1\} \tag{3.3}$$

kde $x_{(i,j)}$ jsou pixely vstupního obrazu
$k_{i,j}$ je konvoluční vzor

0	-1	0
-1	5	-1
0	-1	0

(a) Zaostření

0	1	0
1	1	1
0	1	0

(b) Rozmazání

-1	0	1
-1	0	1
-1	0	1

(c) Zvýraznění svislých hran

Tabulka 3.1: Příklady konvolučních vzorů

3.2.2 Nelineární filtry

Možnost provádění operací nad obrazem pomocí lineárních filtrů je však omezena. Tyto filtry se dají například použít pro detekci vertikálních nebo horizontálních hran, pokud však chceme detekovat hranu vedoucí libovolným směrem, nebudou fungovat. V takových případech nasazujeme metody využívající nelineárních filtrů, což jsou obecně filtry, které nesplňují výše zmíněné podmínky linearity.

Příkladem nelineárního filtru je například Sobelův filtr. V podstatě se jedná o současnou aplikaci dvou lineárních filtrů na jeden obraz podle rovnice[8, s.36]:

$$S = \sqrt{{f_1}^2 + {f_2}^2}, \tag{3.4}$$

kde f_1 a f_2 jsou lineárními filtry, přičemž f_1 je popsaný konvolučním vzorem 3.1c a f_2 je popsán transponovaným vzorem filtru f_2.Cílem Sobelova filtru je detekce resp. zvýraznění hran.

Vylepšenou verzí je pak například Scharrův filtr pracující podle stejného vzorce, ale s rozdílnými konvolučními vzory obou jeho lineárních filtrů f_1:3.2a a f_2:3.2b. Na obdobném pricnipu funguje i množství dalších filtrů – například Robertsův, Laplaceův, Prewittův a Kirschův nebo Cannyho metoda, které jsou blíže popsány v [7, kap.5],[3, kap.6].

-3	0	3
-10	0	10
-3	0	3

(a)

-3	-10	-3
0	0	0
3	10	3

(b)

Tabulka 3.2: Konvoluční vzory Scharrova filtru

Další z hojně užívaných metod spadajících do kategorie nelineárních filtrů jsou takzvané prahovací filtry. Ty slouží k binarizaci obrazu, neboli převádí všechny pixely na jednu ze dvou hodnot – zpravidla se jedná o černou a bílou nebo jiné krajní hodnoty v závislosti na použitém barevném modelu. Výpočet se provádí pro každý bod zvlášť a určuje se porovnáním hodnoty bodu ku předem stanovené hodnotě nazývané práh(angl. threshold). Prahovací filtr lze popsat následující rovnicí[8, kap. 7.4]:

$$y_{i,j} = \begin{cases} 0, & x_{i,j} < t \\ 1, & x_{i,j} \geq t \end{cases} \tag{3.5}$$

kde t udává hodnotu prahu.

3.3 Rekonstrukce obrazu

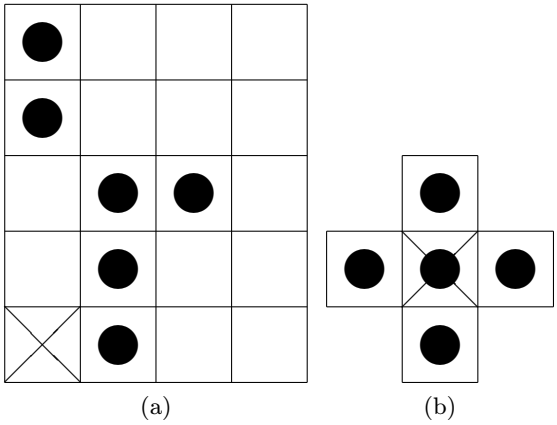
Kromě šumu a špatného osvětlení objektu se můžou vyskytnout i další jevy zhoršující snímaný obraz jako jsou defekty na čočkách, zrnitost filmu(v případě že zpracováváme video ze záznamu) nebo relativní pohyb mezi kamerou a sledovaným objektem. Metody pro rekonstrukci obrazu se snaží obraz rekonstruovat do jeho správné podoby. Vstupem těchto metod je tedy poškozený obraz a výstupem obraz zrekonstruovaný. Tyto metody dělíme na dvě skupiny: deterministické a stochastické. Deterministické je používají v obrazech s nízkým šumem s dopředu známou funkcí popisující jejich poškození. Stochastické metody se snaží přijít na nejlepší způsob rekonstrukce na základě daného stochastického kritéria.

K rekonstrukci obrazu se používají například inverzní nebo Wienerova filtrace. Inverzní filtrace vychází z předpokladu, že poškození obrazu je vyjádřitelné pomocí lineární funkce. Wienerova filtrace vrací předpokládaný původní nepoškozený obraz s minimální průměrnou kvadratickou chybou. Tento optimální předpoklad je vyjádřen pomocí nelineární funkce. Další informace k problematice jsou v [7] a [4].

3.4 Matematická morfologie

Hojně užívanými metodami při předzpracování obrazu jsou i metody patřící do takzvané matematické morfologie. Na rozdíl třeba od filtrů3.2 nepracuje při zpracování obrazu s jednotlivými body, ale s celými množinami přičemž bere v potaz například i jejich propojenost nebo tvar. Použití těchto operací vede zpravidla ke zjednodušení obrazu(ztrátě informací) a zároveň k zesílení a uchování hlavních tvarových vlastností objektů.

Morfologické operace pracují nad bodovými množinami. Příkladem takové množiny je množina $X = \{(1,0)(1,1),(1,2),(2,2),(0,3),(0,4)\}$ na obrázku 3.1a. Relace bodové množiny s jinou, menší bodovou množinou(strukturním elementem) se nazývá morfologickou transformací. Transformace spočívá v tom, že strukturní element je systematicky posouván nad celým X. Pro každý jeden bod je pak vyhodnocena relace mezi X a strukturním elementem. Výsledek je uložen do výsledného obrazu.



Obrázek 3.1: Bodová množina a strukturní element[7]

Mezi základní morfologické transformace patří dilatace a eroze. Dilatace spojuje dvě množiny pomocí vektorového sčítání. Jedná se komutativní a asociativní operaci popsatelnou vzorcem:

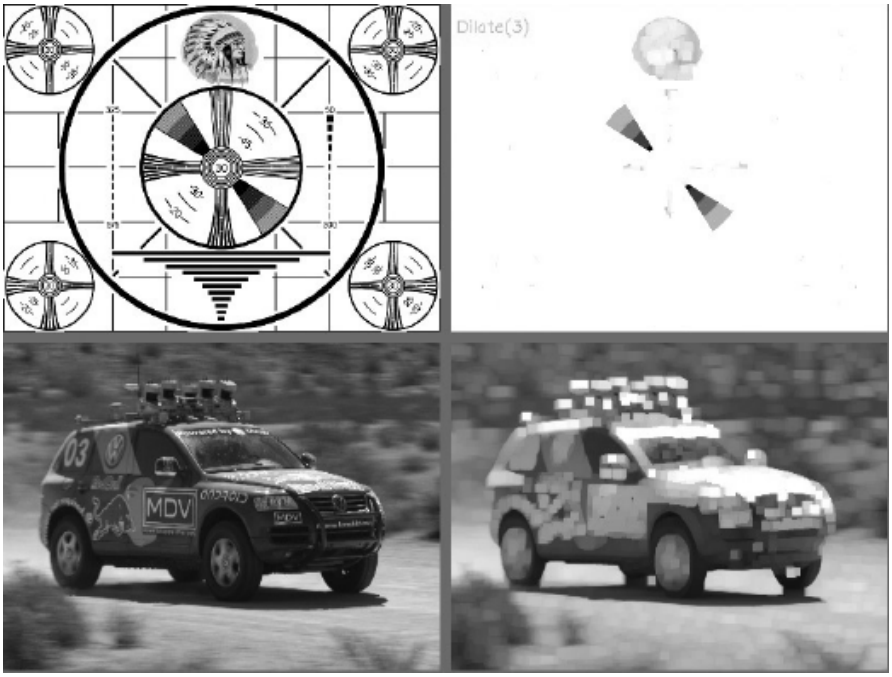
$$X \oplus B = \left\{p \in \varepsilon^2 : p = x + b, x \in X \text{ a } b \in B\right\} \tag{3.6}$$

Výsledkem použití dilatace je například změna pixelů sousedících s objektem na pixely objektu(obr. **??**). Opakem dilatace je eroze používající vektorového odečítání:

$$X \ominus B = \left\{p \in \varepsilon^2 : p = x + b, x \in X \text{ pro každý } b \in B\right\} \tag{3.7}$$

Ani jedna z metod však není invertibilní. Pokud tedy použijeme na obraz nejdříve dilataci a vzápětí erozi, nezískáme zpět původní obraz. Kombinaci těchto dvou metod pak využívají další metody jako třeba *otevírání* a **zavírání** spočívající v aplikaci eroze a následně

dilatace resp. dilatace a následně eroze nebo metody *top hat* a *black hat*. Další informace k matematické morfologii jsou v [7, 3, 4].



Obrázek 3.2: Ukázka dilatace. Zdroj: [3]

3.5 Segmentace

Segmentace, byť bývá řazena stále do metod pro zpracování obrazu, pracuje již s vyšší mírou abstrakce. Jejím cílem je obraz rozdělit na tolik částí(segmentů), aby každá obsahovala pouze jeden objekt reálného světa. Díky tomu se pak můžeme pokusit o vytvoření globálního popisu celého obrazu, což už jsou prvky náležející do oblasti počítačového vidění. V praxi bohužel často dochází k překrývání jednotlivých objektů, případně je jejich vzájemné rozlišení značně ztíženo sníženou kvalitou obrazu.

Vzhledem k důležitosti segmentace vzniklo množství metod. Ty nejzákladnější se dělí na tři skupiny:

- prahování
- podle hran
- podle oblastí

Často používanými metodami jsou například odečítání pozadí nebo analýza shluku s využitím *mean-shift* algoritmu. Odečítání pozadí je metodou, která nachází největší uplatnění při detekci pohybu, proto bude podrobněji zmíněna v příští kapitole.

Nejjednodušší a nejrychlejší metodou segmentace je nasazení prahovacího filtru vyjádřitelného vzorcem 3.5. Prahovaní celého obrazu s použitím jediné prahovací konstanty však může fungovat jen za velmi výjimečných okolností. Pro reálné nasazení je nutné sáhnout po sofistikovanějších metodách jako je lokální prahování, prahování každé vrstvy barevného

modelu zvlášť nebo užití více prahovacích konstant. Další z možností je průběžně odvozovat prahovací konstantu pomocí analýzy histogramu.

Metody segmentace podle hran se snaží detekovat hrany v obraze s pomocí filtrů. Na takto detekovaných hranách se pak pak pomocí algoritmů pro prohledávání grafů snaží vyhledávat uzavřené oblasti, které by reprezentovaly skutečné objekty. V případě, že předem víme, jaký tvar mají hledané objekty a pokud ho můžeme popsat pomocí analytických rovnic, můžeme k nalezení objektu v obraze použít například Houghovy transformace.

Poslední z trojice skupin segmentačních metod jsou segmentace podle oblastí. Jejich cílem je rozdělit obraz na co nejhomogennější oblasti podle vzorců:

$$R = \bigcup_{i=1}^{S} R_i, \quad R_i \cap R_j = \emptyset, \quad i \neq j, \tag{3.8}$$

$$H(R_i) = \text{TRUE}, \quad i = 1, 2, \ldots, S, \tag{3.9}$$

$$H(R_i \cup R_j) = \text{FALSE}, \quad i \neq j, \quad R_i \text{ sousedí s } R_j. \tag{3.10}$$

Hlavními způsoby, jak co nejvíce zvětšit homogení oblasti, jsou slučování oblastí, rozdělování oblastí a kombinace obou předchozích metod. Při slučování oblastí je procházen obraz rozdělený na značné množství segmentů. Jednotlivé oblasti jsou pak spojovány tak, aby naplnili podmínky v 3.9 a 3.10. Naopak při rozdělování je procházen obraz, který nesplňuje podmínku 3.8, a proto je rozdělen podle všech tří výše zmíněných podmínek.

3.5.1 Mean-shift segmentace

Dle našeho rozdělení by tato segmentační metoda patřila mezi metody pro segmentaci podle oblastí. Jejím základem je *mean-shift* algoritmus, který byl poprvé prezentován v sedmdesátých letech, ale při zpracování obrazu se začal prosazovat až po svém znovuobjevení v článku [1]. K obrazovým datům přistupuje jako ke shlukům. Podle [19] jsou shluky skupiny jednotek, do nichž jsou setříděny jednotky(v případě zpracování obrazu pixely) tak, aby si jednotky náležející do stejné skupiny byly podobnější než jednotky z jiných skupin.

V případě *mean-shift* algoritmu vycházíme z předpokladu, že hustota shluku roste směrem k jeho středu. Následující popis metody je převzat z [7, kap. 7.1]Hustotu shluku můžeme vyjádřit pomocí rovnice:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K \left(\frac{\| x - x_i \|}{h} \right), \tag{3.11}$$

kde n udává počet bodů x_i, h znamená šířku jádra, x je středem jádra a K je funkcí pro výpočet hodnotu jednotky x_i v rámci jádra. *Mean-shift* algoritmus používá takzvané Epanechnikovo jádro. Pro potřeby algoritmu je dosahováno nejlepších výsledků, pokud je aplikováno na oblast obrazu ve tvaru kružnice nebo elipsy. Funkce jádra je vyjádřitelná následující rovnicí:

$$K(x) = \begin{cases} c\left(1 - \| x^2 \|\right) & \text{když } \| x \| \leq 1, \\ 0 & \text{jinak}, \end{cases} \tag{3.12}$$

Tvar jádra pak vyjadřuje rovnice:

$$k(x) = \begin{cases} 1 - x & \text{pro} 0 \geq x \geq 1, \\ 0 & \text{pro} x\ 1, \end{cases} \tag{3.13}$$

Vztah mezi $K(x)$ a $k(x)$ je následující:

$$K\left(\frac{x-x_i}{h}\right) = c_k k\left(\left\|\frac{x-x_i}{h}\right\|\right), \tag{3.14}$$

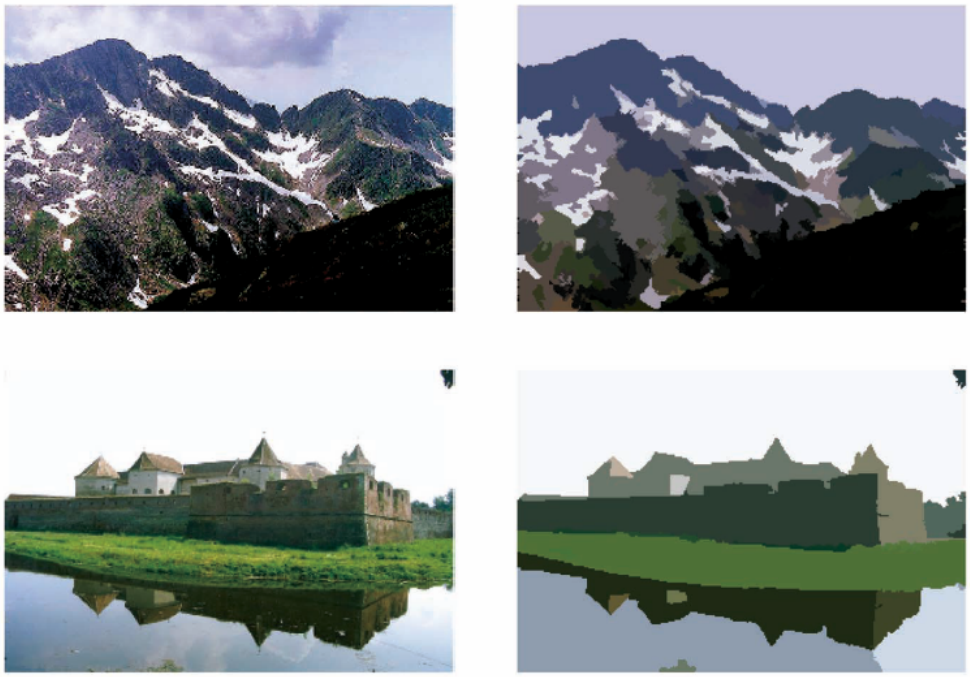
kde c_k je normalizační konstanta.

Cílem *mean-shift* algoritmu je najít nový středový bod jádra, pro který bude hustota největší. To znamená, že hledáme takové x, pro které platí $\bigtriangledown f(x) = 0$. *Mean-shift* algoritmus nám umožňuje zjistit novou pozici bez toho, abychom museli počítat hustotu pravděpodobnosti. Stačí pouze vypočítat gradient hustoty. Po úpravách popsaných v [1, 7] dostaneme výsledný vzorec pro výpočet nové pozice jádra:

$$y_{j+1} = \sum_{i=1}^{n} x_i g\left(\left\|\frac{y_j - x_i}{h}\right\|^2\right) / \sum_{i=1}^{n} g\left(\left\|\frac{y_j - x_i}{h}\right\|^2\right), \tag{3.15}$$

kde $g(x) = k'(x)$. Segmentace následně probíhá tak, že pro každý pixel x_i inicializujeme krok $j = 1$ a $y_{i,1} = x_i$. Podle vzorce 3.15 určujeme $y_{i,j+1}$ dokud není dosaženo konvergence $y_{i,con}$. Výsledkem této operace je dvojice hodnot x_i^s a $y_{i,con}^r$ pro každý pixel. Z takto ohodnocených bodů jsou pak tvořeny shluky $\{C_p\}_{p=1,\ldots,m}$. Podmínkou pro zařazení bodu do shluku je vzdálenost obou jeho hodnot menší než h_s respektive h_r. Každému pixelu $i = 1, \ldots, n$ je přiřazeno $L_i = \{p | z_i \in C_p\}$. Eventuálně je ještě vhodné eliminovat příliš malé oblasti.

Metoda poskytuje velmi dobré výsledky, je však citlivá na správné nastavení svého jediného parametru h – velikosti jádra. Při příliš vysoké hodnoty způsobí ztrátu velkého množství informací nemluvě o vysoké výpočetní náročnosti takové operace. Na druhou stranu příliš nízká hodnota vede ke vzniku příliš mnoha segmentů a segmentace ztrácí smysl.



Obrázek 3.3: Ukázka *mean-shift* segmentace. Zdroj: [1]

Kapitola 4

Sledování cílů

Sledování cílů je velmi důležitou činností spadající do oboru počítačového vidění, jejímž cílem je udržovat přehled o pozici cíle. Tato činnost sestává ze tří základních kroků. Prvním je detekce cíle. Po detekování je možné pokusit se spárovat shodné cíle v po sobě jdoucích snímcích. Závěrečným krokem je pak analýza údajů vedoucí k vytvoření modelu pohybu cíle, která nám do jisté míry umožní předpovídat jeho pohyb.

Metod po sledování cílů existuje několik. Žádná z nich však není dostatečně komplexní a jsou schopné detekovat/sledovat pouze určité vlastnosti cíle. Při volbě metody musíme brát v potaz o jaký typ cíle se jedná, jaké jsou jeho vlastnosti, v jakém prostředí se bude pohybovat a za jakých podmínek ho budeme sledovat. Dále hrají roli také parametry snímací techniky. V případě vojenských cílů můžeme předpokládat, že budeme sledovat cíle spíše na větší vzdálenosti. Sledování musíme být schopni za každých podmínek, proto můžeme předpokládat i použití prostředků pro podporu vidění. Vojenské cíle bývají často kamuflovány, nelze se tedy spolehnout, že cíl bude dostatečně barevně odlišitelný od svého okolí. Také je velmi pravděpodobné, že bude nutné cíl sledovat v reálném čase a s nestacionárními snímači. Po nadefinování podmínek můžeme přistoupit k výběru vhodného modelu pro popis cíle a vlastnosti, na kterou se zaměříme při sledování.

V první části této kapitoly si nejprve stručně představíme jednotlivé způsoby modelování cíle. Poté bude následovat přehled stěžejních metod používaných pro detekci cílů ve snímcích, z nichž některé jsou přímo schopny spárovat odpovídající cíle v po sobě jdoucích snímcích. V poslední části pak budou uvedeny dvě významné metody pro predikci pohybu cílů a jedna pro sledování cílů.

4.1 Model cíle

Model cíle slouží k zjednodušenému matematickému popisu cíle. Problém vhodného modelování je skutečnost, že všechny cíle jsou v reálném světě popsatelné v trojrozměrné soustavě. V počítačovém vidění však máme k dispozici pouze dvourozměrný obraz, proto i malý pohyb cíle jako například zatočení automobilu na křižovatce může vést k zneplatnění celého modelu. Proto jsme nuceni modelovat cíl na základě jeho vlastností, které se buďto nejméně mění, nebo je jejich změna snadno zjistitelná a v modelu aktualizovatelná.

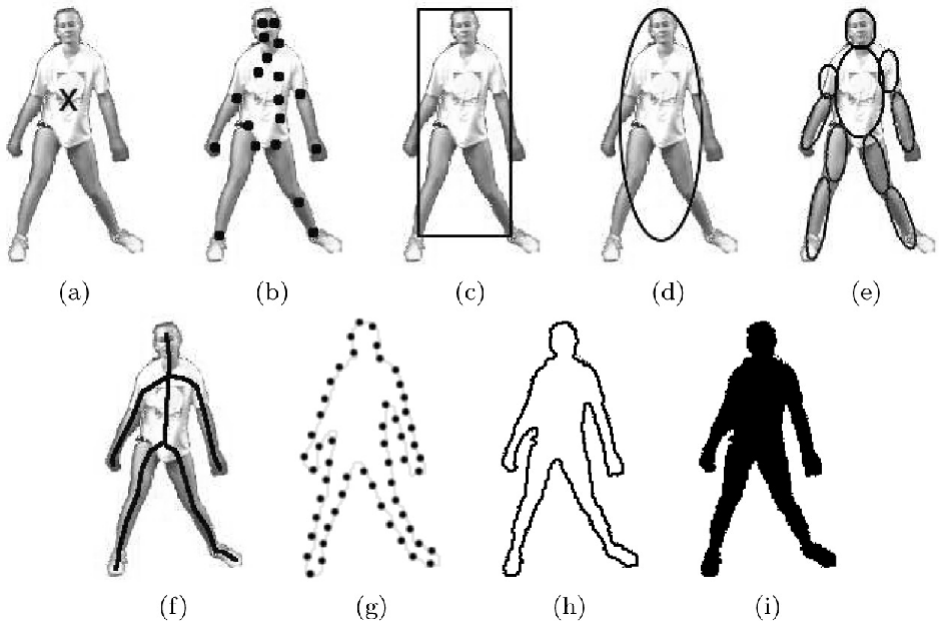
Asi nejznámější reprezentací cíle v prostoru je pomocí jednoho bodu. Zpravidla se jedná o středový bod sledovaného cíle zvaný centroid. Pro větší objekty je však tato reprezentace dost nepřesná a vedla by ke ztrátě informací, proto se používá především pro modelování velmi malých cílů. Větší a především tuhé cíle (cíle jejichž tvar se nemění) se dají vyjádřit

pomocí základních geometrických tvarů jako jsou čtverec, obdélník, kruh nebo elipsa.

Pokročilejšími metodami schopnými vypořádat se i netuhými cíli je modelování pomocí kontury cíle. Kontura vyjadřuje ohraničení cíle, přičemž oblast uvnitř kontury se nazývá silueta. Úspěšné vytvoření siluety je pak základem pro další z modelů a tím je kostra. Kostra spolu se spojenými geometrickými tvary jsou například velmi vhodnými pro aplikace zabývající se sledováním a analýzou pohybu osob.

Dalšími z metod pro modelování cíle je kombinace vymezení cíle v prostoru pomocí jednoduchého geometrického útvaru a hustoty pravděpodobnosti výskytu určitých vlastností cíle v tomto vymezeném prostoru. Příkladem takového postupu jsou například *kernel-based* metody[1]. O poznání jednodušší přístup je při reprezentaci cíle pomocí vzoru. Vzorem je myšlen obraz cíle nebo jeho části vyjádřený texturou, který je pak vyhledáván v následujícím snímku.

Z uvedených modelů se pro reprezentaci vojenských cílů jeví jako nejperspektivnější modelovaní cíle použitím jednoho bodu, což nachází uplatnění při sledování velmi vzdálených a tudíž malých cílů. Pro větší cíle se nabízí reprezentace pomocí vzoru, která má však smysl jen pokud by se model cíle příliš neměnil což je situace nastávající například při sledování kolony vozidel. Poslední možností je sledování na základě kontur, bohužel vhodnou technikou kamufláže se dají kontury cíle rozbít natolik, že je není možné rozpoznat ani lidským okem, natož technickými prostředky.



Obrázek 4.1: Jednotlivé typy reprezentace cílů: (a) středový bod, (b) více bodů, (c)(d) geometrický útvar, (e) spojené útvary, (f) kostra, (g) řídící body kontury, (h) kontura, (i) obrys. Zdroj: [25]

[1]angl. metody založené na jádře, v českých zdrojích se používá běžně anglický název, ale při popisu metod se *kernel* překládá jako jádro

4.2 Detekce cíle

Každá metoda pro sledování cílů potřebuje detekovat cíle ať již v každém snímku nebo pouze v prvním snímku sekvence. Získané pozic potenciálních cílů jsou předány sledovacím metodám. Sledovací metody pak určí, který cíl odpovídá kterému objektu.

Detekční metody se dělí na dvě základní skupiny. První skupina detekuje cíle výhradně na základě některé z jeho vlastností. Takovéto metody si vystačí pouze s informacemi obsaženými v pravě zpracovávaném snímku a řadíme mezi ně některé pokročilejší segmentační metody jako třeba shlukování s algoritmem *mean-shift* 3.5.1, dále pak detektory význačných bodů, klasifikátory nebo prosté hledaní shodné textury.

Do druhé skupiny naopak patří metody detekující cíl na základě jeho pohybu nebo přesněji řečeno na základě optického toku v obraze. Optický tok vyjadřuje změny v obraze v důsledku pohybu během časového intervalu. Tento interval musí být dostatečně krátký, aby mezi jednotlivými snímky nedocházelo k příliš velkým změnám. Také je důležité zachování konstantních světelných podmínek, protože metody pracující s datovým tokem obvykle pracují s hodnotami jasu. Kromě odečítání pozadí je tato skupina metod reprezentována také metodami Lucas-Kanade nebo Horn-Schunck, které jsou blíže popsány v [3, kap.10].

4.2.1 Detektory bodů

Při detekci bodů se snažíme najít v obraze body, které jsou ve svém okolí významné. Za významné přitom považujeme body, které jsou invariantní vůči změnám jasu nebo natočení kamery. Mezi zástupce těchto metod patří například Moravcův operátor, Harrisův detektor významných bodů nebo SIFT detektor [25].

Hledání významného bodu pomocí Moravcova operátoru probíhá tak, že nejdříve je vypočítána variace jasu obrazu pomocí masky o velikosti 4×4. Variace je počítána zvlášť ve vertikálním i horizontálním směru i po obou diagonálách. Z vypočtených hodnot vybere minimální hodnoty. Pokud jsou vybrané hodnoty zároveň i lokálním minimem v okolí bodu o velikosti 12×12, pak je bod považován za významný.

4.2.2 Klasifikátory

Klasifikátory bývají často označovány také jako metody pro učení s učitelem. Aby mohly objekty detekovat, musí se je nejdříve naučit rozpoznat. K učení používají sadu vzorků skládající se z různých pohledů na objekt. Pokud například budeme chtít rozeznávat tank, sada vzorků se bude skládat ze snímků zachycujících tank ze všech možných pozorovatelných úhlů. Pokud takový klasifikátor aplikujeme, vrátí nám detekované objekty třídy tank. Sada vzorků i rozdělení do tříd se definují manuálně – proto mluvíme o učení s učitelem.

Důležitým předpokladem pro správné použití klasifikátorů je vhodná volba klasifikovaných vlastností, která umožňuje vzájemné rozlišení jednotlivých tříd. Kromě obvyklých vlastností jako je barva, tvar nebo textura můžeme klasifikovat objekty i na základě jiných vlastností. Například pomocí plochy objektu, natočení objektu nebo hustoty pravděpodobnosti výskytu cíle. Přesnost klasifikátoru je závislá na kvalitě a množství vzorků. Čím více vzorků, tím větší pravděpodobnost detekce objektu.

Příkladem klasifikátoru je například metoda *Adaboost*. Jedná se o iterativní metodu, která vyváří velmi přesný klasifikátor kombinací základních klasifikátorů. V prvním kroku fáze učení určí váhy jednotlivých vzorků. Následně vybere takový ze základních klasifikátorů, který nad danými vzorky pracuje s nejmenší chybou, přičemž chyba je úměrná váhám

špatně klasifikovaných vzorků. Váhy špatně klasifikovaných vzorků jsou následně zvýšeny, takže algoritmus zvolí další základní klasifikátor, který si s nimi v další iteraci povede lépe. Výsledkem je potom velmi přesný klasifikátor.

4.2.3 Porovnávání vzoru

Porovnávání vzoru je velmi jednoduchou metodou. Modelem cíle je obraz celého cíle nebo alespoň jeho části, který je reprezentován texturou. Hledaná textura je postupně posouvána nad aktuálním snímkem a pro každý pixel je spočítána hodnota vyjadřující míru shody. Hledání textury je podobné klasifikátoru, jehož třída je klasifikována pouze na základě jednoho vzorku. K výpočtu se používá metod jako například zjišťování druhé mocniny odchylek 4.1, korelace 4.2 nebo korelace s použitím koeficientu 4.3.

$$R(x,y) = \sum_{x',y'} \left[T(x',y') - I(x+x',y+y')\right]^2 \tag{4.1}$$

$$R(x,y) = \sum_{x',y'} \left[T(x',y') \cdot I(x+x',y+y')\right]^2 \tag{4.2}$$

$$\begin{aligned} R(x,y) &= \sum_{x',y'} \left[T'(x',y') \cdot I'(x+x',y+y')\right]^2 \\ T'(x',y') &= T(x',y') - \frac{1}{(w \cdot h) \sum_{x'',y''} T(x'',y'')} \\ I'(x+x',y+y') &= \frac{1}{(w \cdot h) \sum_{x'',y''} I(x+x'',y+y'')} \end{aligned} \tag{4.3}$$

Nevýhodou metod založených na vyhledávání textury je náchylnost ke změně modelu a v případě prohledávání velkého obrazu i vysoká výpočetní náročnost. Nejčastěji se s nimi tedy spíše v rolích pomocných metod.

4.2.4 Odečítání pozadí

Díky své jednoduchosti a četnosti používání stacionárních kamer se odečítání pozadí dočkalo značného rozšíření zejména v bezpečnostních kamerách nebo systémech pro sledování provozu na křižovatkách apd. Základní myšlenkou této metody je vytvoření modelu pozadí. Jakýkoliv cizí objekt, který by se v obraze objevil, by potom mohl být snadno detekován prostým odečtením hodnot aktuálního snímku od modelu pozadí. V nejjednodušší variantě této metody je za model pozadí považován pouze předchozí snímek. Ač je tento postup velmi rychlý, vyžaduje pro správnou funkčnost splnění několik základních podmínek.

První je konstantní osvětlení snímané scény. Projevy jakékoliv změny osvětlení mohou sahat od detekce neexistujících objektů až po úplné oslepení obrazu. Dalším předpokladem pro správnou činnost je statičnost scény, protože jakýkoliv pohyb s kamerou by vedl ke změně snímané scény a tedy i ke kompromitaci modelu pozadí. Poslední z důležitých podmínek je rychlost pohybu objektu ve scéně. U příliš pomalu se pohybujícího objektu hrozí, že se stane součástí modelu pozadí a tudíž nebude nadále pozorovatelný. Ve snaze zvýšit odolnost proti těmto negativním jevům bylo vytvořeno množství metod. Dále budou zmíněna pro ukázku pouze metoda modelování pozadí pomocí mediánu. Další metody jsou zmíněny a popsány v [5, 7, 3].

Při modelování pozadí pomocí mediánové filtrace postupujeme podle následujícího postupu [7]:

1. V rámci inicializace získáme K snímků. Pro každý jeden bod spočítáme medián z jeho hodnot v K snímcích. Při vícevrstvém barevném modelu provádíme tento výpočet pro každou vrstvu zvlášť. Výsledkem je počáteční model pozadí.

2. Získáme následující snímek $(K + 1)$ a vypočítáme rozdíl mezi tímto snímkem a modelem pozadí.

3. Pro odstranění šumu aplikujeme na tento rozdíl prahovací filtr.

4. Pomocí dalších morfologických operací(viz. 3.2) odstraníme velmi malé oblasti. Zbylé oblasti reprezentují pohybující se cíle.

5. Ze sady snímků pro vytváření modelu pozadí odstraníme nejstarší snímek a zároveň vložíme snímek $K + 1$. Provedeme přepočet modelu pozadí.

6. Pokračujeme krokem č.2

Vzhledem k náročnosti kroku č.5 a potřebě uchovávat v paměti všech K snímků existuje zjednodušení, kdy je uložen pouze model pozadí a hodnoty jednotlivých pixelů jsou inkrementovány nebo dekrementovány podle toho, zda jsou jasnější nebo tmavší než pixely v aktuálním snímku.

4.3 Sledování cíle

Metody pro sledování cílů slouží k udržování přehledu o pohybu cíle – tedy modelování jeho trajektorie. Vstup těchto metod jsou polohy potenciálních neboli kandidátních cílů. V závislosti na zvolené metodě dochází k rozhodnutí, který z kandidátů v aktuálním snímku je totožný se sledovaným cílem na předchozím snímku. Výstupem je v první řadě aktuální pozice cíle, případně i vygenerovaná trajektorie.

Postup získání poloh kandidátních cílů lze rozdělit do dvou skupin. První způsob se plně spoléhá na detektory, které najdou všechny objekty splňující základní kritéria. Typickým příkladem takového vstupu jsou údaje opatřené pomocí odečítání pozadí. Při užití druhého způsobu je potenciální poloha cíle odhadována přímo sledovací metodou. Potenciální cíle jsou pak vyhodnoceny a výsledný model aktualizován.

V dalším textu bude zmíněna jedna z *kernel-based* metod vycházející z *mean-shift* algoritmu a dále Kalmanův a částicový filtr, které patří mezi nejčastěji používané metody pro modelování trajektorie a predikci polohy cíle.

4.3.1 Kernel-based metody

Tyto metody pracují s jádrem, které reprezentuje cíl v prostoru. Další vlastnosti cíle jsou reprezentovány hustotou pravděpodobnosti jejich výskytu v prostoru vymezeném jádrem. Dále bude popsána jedna z těchto metod pracující s *mean-shift* algoritmem 3.5.1. Vstupem je poloha cíle v čase $t - 1$, která spolu s modelem cíle je dostatečná pro určení a detekci konkrétního cíle v čase t. V případě této metody je cíl modelován pomocí Epanechnikova jádra 3.12 a sledovanou vlastností je barevné rozložení.

V inicializační fázi je vytvořen model cíle vycházejíce z barevného histogramu. Použita je následující rovnice:

$$\hat{q}_u = C \sum_{i=1}^{n} k \left(\|x_i^*\|^2 \right) \delta \left(b(x_i^*) - u \right), \tag{4.4}$$

kde u je indexem třídy v histogramu a $u \in \{1, \ldots, m\}$, přičemž m je počet tříd histogramu. Hodnota n udává počet pixelů v jádru, x_i^* je normalizovaná souřadnice pixelu v jádru. Funkce $b(x_i^*)$ vrácí index třídy histogramu, do kterého spadá barevná hodnota pixelu x_i^*. C je normalizační konstanta a δ Kroneckerova delta.

$$\delta(x) = \begin{cases} 1 & \text{pro} x = 0, \\ 0 & \text{jinak}, \end{cases} \tag{4.5}$$

$$C = \frac{1}{\sum_{i=1}^{n} k \left(\|x_i^*\|^2 \right)} \tag{4.6}$$

Z obdobného vzorce se vytvoří i model pro potenciální cíl:

$$\hat{p}_u = C_h \sum_{i=1}^{n_h} k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \delta \left(b(x_i) - u \right), \tag{4.7}$$

kde y udává střed jádra potenciálního cíle. Normalizační konstanta se vypočítá podle vzorce:

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)} \tag{4.8}$$

Vzájemný vztah mezi rozloženími pravděpodobnosti $\hat{p}_u$ a $\hat{q}_u$ je

$$\sum_{u=1}^{m} \hat{q}_u = \sum_{u=1}^{m} \hat{p}_u = 1$$

. V případě této metody je pro určení míry shody mezi modelem cíle a kandidáta na cíl použito Bhattacharyyovy vzdálenosti. Ta se v našem případě vypočítá následovně:

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u}, \tag{4.9}$$

kde y opět udává střed jádra kandidáta na cíl.
Sledování cíle probíhá podle postupu z [7] následovně:

1. Předpokládáme, že máme k dispozici model cíle $\{\hat{q}_u\}$ pro $u \in \{1, \ldots, m\}$ a že známe pozici cíle v předchozím snímku $\hat{y}_0$.

2. V aktuální snímku použijeme pozici $\hat{y}_0$ jako pozici kandidáta a vypočítáme $\{\hat{p}_u(\hat{y}_0)\}$ a $\rho[\hat{p}(\hat{y}_0), \hat{q}]$.

3. Vypočítáme váhy jednotlivých bodů kandidáta podle vzorce:

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta \left(b(x_i) - u \right) \tag{4.10}$$

4. Určíme pozice nového kandidáta podle rovnice:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left|\left|\frac{\hat{y}_0 - x_i}{h}\right|\right|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left|\left|\frac{\hat{y}_0 - x_i}{h}\right|\right|^2\right)} \tag{4.11}$$

5. Vypočítáme $\{\hat{p}_u(\hat{y}_1)\}$ a $\rho[\hat{p}(\hat{y}_1), \hat{q}]$.

6. Pokud je $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$, přehodnotíme pozici kandidáta na $\hat{y}_1 = \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$, opět vypočítáme $\rho[\hat{p}(\hat{y}_1), \hat{q}]$ a vrátíme se na začátek kroku č.6.

7. Pokud je vzdálenost mezi body $\hat{y}_1$ a $\hat{y}_0$ menší než ε nebo pokud sme dosáhli limitu pro počet iterací, měli bychom mít výslednou polohu cíle v aktuálním snímku, jinak se vrátíme do kroku č.3.

Detaily ke vzorcům a jejich odvození jsou v [2, 7]. Tato metoda se osvědčila jako spolehlivá a použitelná i v reálném čase. Je však poměrně citlivá na správně nastavené parametry. Podobně jako u *mean-shiftové* segmentace je i zde důležitý výběr vhodně velkého jádra. Další důležitou podmínkou je vhodné nastavení podmínek pro ukončení výpočtu. Za dostatečné se ukázalo nastavení počtu iterací v řádu jednotek viz.[3]. Z hlediska sledování vojenských cílů má však tato metoda vážný nedostatek v podobě modelování cíle pomocí barevného histogramu. V případě vojenských cílů se dá předpokládat sledování objektů, jejichž zbarvení je téměř nebo úplně nekontrastní vůči okolí. V takových podmínkách je nutné metody modifikovat například přidání dalšího jádra [26].

4.3.2 Kalmanův filtr

Kalmanův filtr je matematická metoda pro práci s naměřenými hodnotami, které trpí šumem nebo jinými nepřesnostmi. Na základě výsledků předchozích měření se snaží najít hodnoty blíže odpovídající skutečnému stavu. Od svého uvedení počátkem 60.let se dočkal značného rozšíření především v měřících systémech, ale může se pochlubit také například nasazením v navigačních systémech nukleárních balistických raket odpalovaných z ponorek, střel Tomahawk nebo raketoplánů NASA. Své využití nachází však i v oblasti počítačového vidění.

Systém modelovaný Kalmanovým filtrem musí být lineární. Odhad nového polohy cíle se provádí na základě pravděpodobnosti výskytu objektu v obraze. Za předpokladu, že trajektorie, po níž se cíl pohybuje, je spojitá, můžeme předpokládanou polohu modelovat jako normální rozložení. Hustotu rozložení pravděpodobnosti vyjádříme jako funkci pozice cíle:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right), \tag{4.12}$$

kde σ je rozptyl a $\bar{x}$ je střední hodnota. Pravděpodobnost v normálním rozložení dosahuje nejvyšších hodnot právě kolem střední hodnoty, což je tím pádem nejpravděpodobnější pozice cíle. Celý výpočet sestává ze dvou hlavních kroků – predikce a korekce. Predikci vyjadřují rovnice:

$$x_{k+1} = A\ x_k + Bu_k + w_{k+1}, \tag{4.13}$$

kde x je stavem systému v čase k, A bývá označována jako tranzitivní matice, která určuje vztahy jednotlivých prvků stavu systému. w_k představují Gaussův šum. Vektor u a matice B se označují jako ovládací vstupy – umožňují započítat do odhadu námi ovlivnitelné zásahy. Pokud je například kamera umístěná na pohyblivé, námi ovládané platformě, můžeme prostřednictvím matice B zapracovat do systému změny polohy kamery. V predikčním kroku je tedy odhadnut stav systému a také kovariance chyby P^- systému podle rovnice:

$$P_k^- = A\ P_{k-1}A^T + Q, \tag{4.14}$$

kde Q je kovarianční maticí.

Následuje korekční krok. Který nejprve spočítá Kalmanův zisk K:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \tag{4.15}$$

kde H je matice určující vztah mezi naměřenými hodnotami a modelem systému a R je matice o rozměru 1×1 nesoucí hodnotu chyby měření. Kalmanův zisk nám tedy udává jakou váhu přikládat novým údajům. S pomocí těchto údajů je vypočítán aktualizovaný stav systému:

$$x_k = x_k^- + K_k(z_k^- - H_k x_k^-), \tag{4.16}$$

kde z_k^- označuje změřené hodnoty. Taktéž je aktualizována hodnota kovariance chyby:

$$P_k = (I\ - K_k H_k) P_k^- \tag{4.17}$$

Informace o Kalmanovu filtru byly čerpány z [7, 3, 6, 21]. Značným omezením Kalmanova filtru je nutná linearita modelovaného systému. Obzvláště při sledování cílů můžou a nastávají situace, které nelze popsat lineárním modelem. Tyto situace se snaží řešit například rozšířený Kalmanův filtr. Daleko lepších výsledků však dosahuje jiný typ metod – částicové filtry.

4.3.3 Částicový filtr

Částicové filtry jsou z hlediska predikce pozice cíle obecnější metodou než Kalmanův filtr. Systém reprezentující pohyb cíle je zde modelován pomocí sady vzorků, které na základě pravděpodobnosti vycházející z empirických zjištění určují, co by se mohlo a co naopak nemohlo stát. Rozšířenou metodou založenou na částicovém filtru je například algoritmus CONDENSATION, který je do češtiny přeložitelný jako podmíněné předávání hustoty [6].

Jeho základní myšlenkou je pravděpodobnostní závislost aktuálního stavu systému na předcházejícím stavu. Každý ze vzorků je tvořen dvojicí stavu systému a odpovídající váhou. Výpočet se podobně jako u Kalmanova filtru skládá ze dvou hlavních částí – predikce a korekce. Predikce i korekce se provádí pro každý vzorek zvlášť. Při predikci dochází nejdříve k převzorkování tak, aby se u vzorků – částic s nejvyšší váhou snížila váha a zároveň se v jejich okolí vygenerovaly částice nové.

Ve fázi korekce se mění váha jednotlivých vzorků na základě výsledků měření. Dobře reprezentujícím částicím je váha zvyšována, naopak hůře reprezentujícím je snižována a částice s nulovou váhou jsou eliminovány. Díky převzorkování však nikdy nedojde k odstranění všech částic.

Kapitola 5

Implementace a testování

Tato kapitola si klade za cíl popsání návrhu a implementace metod a jejich správného vzájemného propojení tak, aby byly schopné sledovat zvolené objekty ve videu. Následuje popis prováděných testů a zhodnocení dosažených výsledků. S přihlédnutím k výsledkům testů jsou pak diskutovány další možnosti vývoje a vylepšení aplikace.

Navržený postup se skládá ze tří základních bloků. Prvním je detekce cíle ve chvíli, kdy se objeví v záběru nebo kdy se ho rozhodneme sledovat. Tyto metody se mohou lišit od metod, které budou dále používány pro detekci a rozpoznání konkrétního objektu. Prvotní detekce je následována vytvořením modelu cíle. Po tomto inicializačním bloku nastává fáze sledování, kdy se snažíme rozpoznat konkrétní objekt ve snímku neboli provést spárování dvou totožných objektů ve dvojici po sobě jdoucích snímků. Třetí blok se snaží výsledky dosažené při sledování analyzovat a na jejich základě predikovat další pohyb cíle.

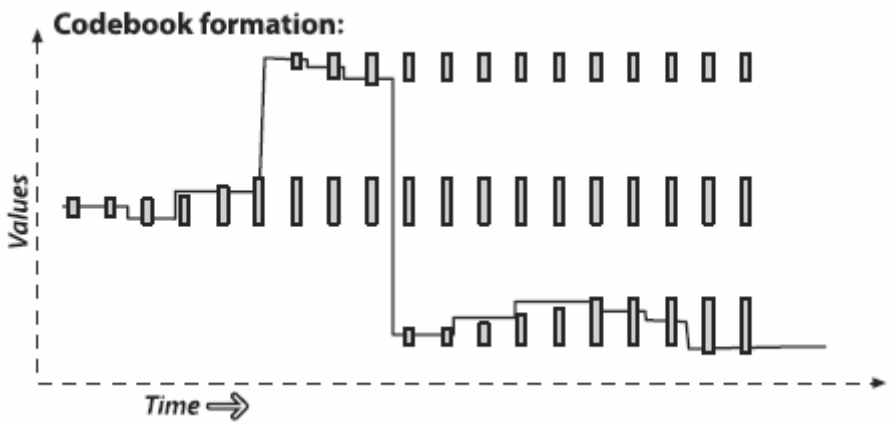
Pro samotnou implementaci byla zvolena *open-source* knihovna *OpenCV*[24]. Jedná se o multiplatformní knihovnu obsahující přes pět set funkcí pro zpracování obrazu a počítačové vidění. Původně byla vyvíjena firmou *Intel*, v současnosti je však volně dostupná pod BSD licencí[22]. Ze tří primárně podporovaných programovacích jazyků byla zvolena verze pro C++. Kromě množství funkcí z oboru počítačového vidění obsahuje *OpenCV* i vlastní knihovny pro jednoduché uživatelské rozhraní a pro přímý přístup k datům uložených v souborech s videem nebo obrázky, odpadá tedy nutnost použití dalších knihoven třetích stran.

5.1 Inicializace

Správná inicializace je nezbytným předpokladem pro další úspěšný průběh sledování. K detekování pohybujícího se cíle ve scéně se nejčastěji používají metody odečítání pozadí (kap. 4.2.4). Výstupem těchto metod je binární obraz, neboli obraz v němž jednotlivé body nesou hodnotu 0 – černá nebo 1 – bílá. Problémy při tvorbě binárního obrazu působí zejména šum, proto je před samotným odečtením pozadí vhodné aplikovat některou z metod filtrace 3.2. Velmi oblíbeným je pro tento účel mediánový filtr.

Značná část metod pro odečítání pozadí je již v *OpenCV* implementována. Jmenovitě se jedná o například *Mixture of Gaussian*(dále jen MoG), která modeluje každý bod pozadí pomocí několik normálních rozložení pravděpodobnosti. Což přináší dobré výsledky, ale s vyššími výpočetními nároky. Z dalších metod stojí za zmínku metoda *Codebook*, která modeluje intenzitu jednotlivých bodů ve všech třech vrstvách barevného modelu pomocí struktur složených z horního a dolního prahu. Pokud je rozsah mezi oběma prahy příliš velký, dochází k rozdělení struktury na dvě menší. Přesnost modelu pozadí je závislá na

délce inicializace, tedy čím více snímků zpracuje, tím lepší podá výsledky.



Obrázek 5.1: Graf změny hodnot jasu pixelu v čase demonstrující použití metody *Codebook*. Pokud se hodnota aktuálního snímku vejde do rozsahu reprezentovaného některou ze struktur, je považována za pozadí. Zdroj: [3]

Vzniklý binární obraz často i přes redukci šumu obsahuje větší množství velmi malých objektů. Naopak větší objekty mohou být narušeny "dírami"nebo přímo fragmentovány na menší, velmi blízko sebe ležící objekty. K zaplnění těchto děr a vytvoření větších, kompaktních objektů stejně jako k redukci malých osamocených objektů je vhodné nasazení metod matematické morfologie. V našem případě je obraz transformován nejprve pomocí metody *open* a vzápětí pomocí *close* (viz. 3.4). Výsledný obraz by měl obsahovat už jen spojité oblasti. Tyto oblasti jsou v anglické literatuře označovány jako *bloby*. Tohoto označení se budeme držet i v této práci, protože český překlad jako kapka nebo skvrna by mohl být zavádějící. Bloby ještě můžeme roztřídit podle velikosti a ignorovat ty, jejichž velikost je pod stanovenou hranicí.

Přestože obě výše zmíněné metody poskytovaly vcelku uspokojivé výsledky, jejich výpočetní nároky se podepisovaly na plynulosti běhu aplikace. Ve výsledné implementaci proto byly nahrazeny manuální detekcí. Cíle volí uživatel pouze s pomocí myši, kdy cíl je označován obdélníkem.

5.2 Sledování

Po úspěšné inicializaci můžeme přistoupit ke sledování cíle. S přihlédnutím k dostupným testovacím videím byly zvoleny dvě metody. První z nich je takzvaná *kernel-based* metoda využívající algoritmus *mean-shift*, která byla podrobně popsána v kapitole 4.3.1). Implementace této metody, která je zmiňována i v [3, kap. 10] se jevila jako krajně nespolehlivá a zdaleka nedosahovala parametrů popsaných v [2]. Proto byla přejata implementace z knihovny zmiňované zde [12]. Důvodem pro volbu této metody bylo, že dokáže sledovat cíle na základě jejich barvy, což je poměrně obecný popis umožnující sledovat libovolně se pohybující cíle. Je však otázkou, na kolik bude schopná sledovat i menší cíle a jak obstojí při nasazení na záběry z termokamery.

Druhá metoda je založena na porovnávání vzoru (viz. 4.2.3). Aby byla snížena výpočetní náročnost, prohledávaná oblast je vymezena obdélníkem o rozměrech $3w \times 3h$, kde w a h jsou šířka resp. výška modelu, přičemž střed tohoto obdélníku odpovídá poslední známé pozici

cíle. Výpočet je prováděn korelační metodou podle vzorce 4.2. Tento způsob je přesnější než metoda SSD a téměř stejně přesná jako metoda korelace s koeficientem, byť má menší výpočetní nároky. K volbě této metody vedla především skutečnost, že testovací videa byla pořízena statickou kamerou snímající plynulý silniční provoz. Díky tomu odpadly problémy spojené s aktualizací modelu. Dá se tedy předpokládat, že tato metoda bude podávat velmi dobré výsledky při rozpoznání cíle. Další motivací je zjištění, nakolik je taková metoda použitelná pro sledování cílů na záběrech termokamery.

5.3 Predikce pozice cíle

Nad oběma metodami je aplikován Kalmanův filtr. Ten slouží k předpovídání polohy cíle v následujícím snímku. Pro inicializaci filtru je nutné detekovat cíl ve dvou po sobě jdoucích snímcích. Teprve poté je schopen predikce. Předpovězené poloha je pak použita jako výchozí bod pro hledání cíle pomocí výše zmíněných metod. V tomto bodě je implementováno vylepšení.

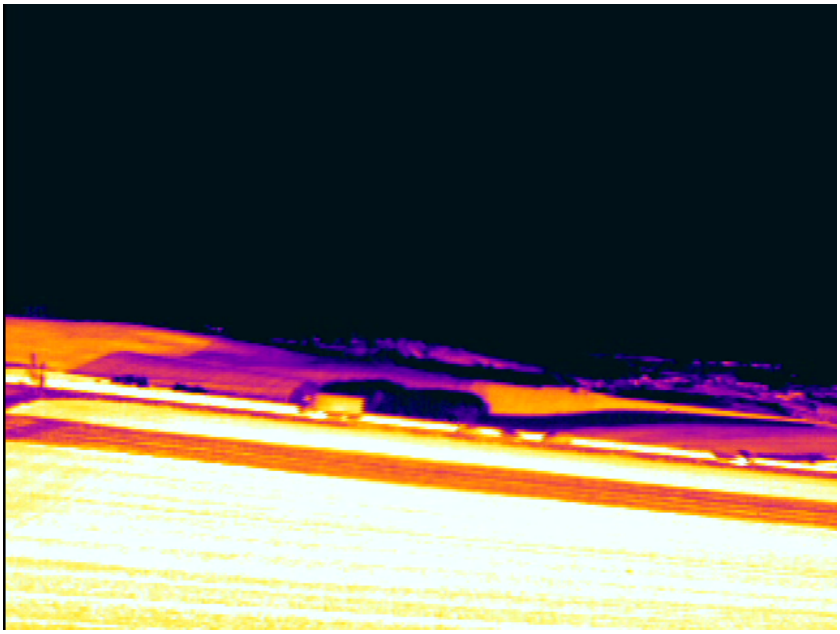
Metoda pro porovnávání vzorů vrací hodnotu vyjadřující míru shody. Míra shody je normalizovaná do intervalu $\langle 0, 1 \rangle$, přičemž hodnota 1 udává největší shodu a naopak 0 absolutní rozdílnost. Při míře shody nižší než nastavený práh jsou výsledky vlastního hledání ignorovány a místo nich jsou použity hodnoty predikované Kalmanovým filtrem. Obdobný postup je aplikován i v případě vysokého rozdílu mezi naměřenou a dosavadní pozicí. Tento krok má za cíl snížení počtu falešných detekcí. Dalším z jeho přínosů je schopnost detekovat cíl i po velmi dlouhém úplném překrytí, podmínkou však je neměnná trajektorie a rychlost cíle.

5.4 Testování

Aby byl popis celého postupu sledování cílů kompletní, musíme jej otestovat a vyhodnotit úspěšnost sledování. Vzhledem k absenci spolehlivé metody, která by byla schopná obdobné činnosti a s níž bychom mohli porovnat výsledky, jsme nuceni provést testování manuálně. K dispozici máme dva druhy videozáznamů. Oba byly pořízeny na stejném místě pomocí stacionárních kamer a snímaly stejný úsek silnice, vzdálený asi 350 metrů. První kamera byla klasická digitální. Při snímání scény měla nastaveno dvojnásobné přiblížení. Druhá byla termokamera bez přiblížení. Záběry byly pořizovány ve dne což na jednu stranu umožnilo pořízení záznamu z normální kamery s téměř konstantním jasem, na druhou stranu teplota ovzduší i země se velmi přiblížila teplotě vozidel v záběru. Výsledkem jsou záběry termokamery s velmi nízkým kontrastem mezi pohybujícími se vozidly a jejich okolím.

Pro samotné testování byly zvoleno několik scénářů, které by ověřily předpokládané silné i slabé stránky obou implementovaných metod. Všechny scénáře byly následně testovány na obou dostupných typech videí. Jediným posuzovaným kritériem je správné určení polohy cíle. Kromě základní metody *mean-shift*(dále jen MS) a porovnávání vzoru (dále jen TM) byly obě metody testovány i v kombinaci s Kalmanovým filtrem (dále jen MS+K resp. TM+K).

První scénář testuje činnost obou metod za optimálních podmínek. Ty spočívají v jednoznačném a nezaměnitelném cíli. Protože obě metody pracují s hodnotami barev všech vrstev, je pro ně nejlepším cílem objekt kontrastní vůči svému okolí. Metoda pro porovnávání vzoru pak podává tím přesnější výsledky, čím je textura(obraz cíle) členitější a tedy hůře zaměnitelná. V obou typech videozáznamů splňují tyto podmínky především těžká nákladní



Obrázek 5.2: Ukázka testovacího videozáznamu z termokamery. Teplotní rozsah kamery byl nastaven na $17 - 23°$

vozidla. Další z podmínek je nerušené sledování, neboli že nesmí dojít k žádnému překrytí sledovaného cíle.

Tento scénář byl testován na vzorku 15 objektů. Výsledky jsou uvedeny v tabulce (5.1). Jak je zřejmé, obě metody si poradily s cíli velmi dobře. Mírné zhoršení u MS metod jsou důsledkem celé koncepce MS algoritmu, který se v každé iteraci snaží najít největší hustotu rozložení barev odpovídající vzoru. V případě záběrů z termokamery, obzvláště pak v našem případě, mají sledované objekty velmi podobnou hustotu rozložení barev jako jejich okolí. Navíc obě metody uspěly při sledování v normálním videozáznamu.

Typ videozáznamu	MS	MS+K	TM	TM+K
Normální	93,3	93,3	93,3	100,0
Termokamera	80,0	86,7	93,3	100,0

Tabulka 5.1: Úspěšnost sledování za optimálních podmínek [%]

Druhý scénář testoval sledování cílů o velikosti osobního auta, tedy cíle nejčastější. Pro testování bylo použito 30 vzorků. Výsledky těchto testů odhalily potíže především u samostatného MS algoritmu, který zcela propadl při sledování cílů z termokamery. Značné zhoršení je také patrné u porovnávání podle vzoru, které z takto malých cílů nebylo schopné vytvořit dostatečně přesný model. V porovnání s předchozím testem zde také vyniká nasazení Kalmanova filtru, který v kombinaci s implementovanými podmínkami umožňuje ignorovat absolutně nepřesná měření.

Třetí scénář se zaměřil na testování kritických situací ke kterým dochází, pokud se dva objekty míjejí. Obzvláště u barevně podobných objektů může dojít k situaci, že metoda vyhodnotí druhý objekt jako cíl. Míjením označujeme situaci, kdy sledovaný cíl je blíže kameře – při míjení tedy nedojde k jeho zakrytí. Vzhledem k nižšímu počtu míjejících se objektů v dostupných testovacích videích byl počet vzorků snížen na 15.

Typ videozáznamu	MS	MS+K	TM	TM+K
Normální	70,0	83,3	53,3	87,6
Termokamera	23,3	66,7	86,7	96,7

Tabulka 5.2: Úspěšnost sledování malých cílů [%]

Typ videozáznamu	MS	MS+K	TM	TM+K
Normální	53,3	86,7	26,7	73,0
Termokamera	13,3	26,6	46,7	80,0

Tabulka 5.3: Úspěšnost sledování míjejících se objektů [%]

Čtvrtý scénář sledoval hlavně situaci, při níž je cíl částečně nebo úplně překrytý po krátký časový úsek. S takovým jevem se můžeme setkat když se auta míjí ve vysoké rychlostí nebo když vozidlo projíždí za nějakou překážkou. Vzhledem k testovacím videům, kdy jsou všechny objekty v jedné rovině je častějším jevem překrytí úplné. Ve videozáznamech z normální kamery se tento jev vyskytuje ve snížené míře, proto bylo k dispozici pouze 10 vzorků. Tento počet jsme proto dodrželi i při testování na záběrech z termokamery.

Typ videozáznamu	MS	MS+K	TM	TM+K
Normální	30,0	50,0	20,0	70,0
Termokamera	10,0	20,0	10,0	90,0

Tabulka 5.4: Úspěšnost sledování při překrytí cíle [%]

Výsledky zjištěné při vyhodnocování třetího a čtvrtého scénáře jenom potvrzují výsledky známé už z druhého testovacího scénáře. Jasně tak dokazují, že MS metoda dosahuje velmi dobrých výsledků v normálním videozáznamu, obzvlášť v případě kdy jsou cíle kontrastní vzhledem ke svému okolí. Naopak metoda porovnávání vzoru dosahuje v normálním obraze horších výsledků než MS, což může být zaviněno defekty a šumem v obrazu, ale hlavně nedokonalostí vzoru. Pro obě metody je citelným vylepšením aplikace Kalmanova filtru, který díky korekci výsledků redukuje chybovost a falešné detekce.

5.5 Další vývoj

Testovaní navrženého postupu odhalilo nedostatky především ve způsobu reprezentace a detekce cíle. Velikost sledovaných cílů je obzvláště u videí z termokamery dost malá. Vhodným řešením tohoto problému by mohlo být provádění detekce cílů pomocí klasifikátoru. Vzhledem k záměru práce sledovat vojenské cíle by učení klasifikátorů bylo snazší, neboť typů vojenské techniky je omezené množství a její vlastnosti jako základní tvary a rozměry jsou známé. Navíc by použití klasifikátorů umožnilo autodetekci cílů. Současný stav s manuální detekcí je příliš závislý na zručnosti uživatele a použití detektoru blobů trpí taktéž chybovostí.

Dalším potenciálním vylepšením by byla aplikace pokročilejšího filtru pro predikci a korekci polohy. Pro tuto funkci se nabízí částicový filtr, který by dokázal modelovat i nelineární pohyb cíle. Toho by se dalo využívat zejména při úplném zakrytí cíle. V současném modelu je tento problém řešen pomocí předpokladu, že rychlost i trajektorie cíle se nemění. Právě využití částicového filtru by umožnilo dynamičtější predikci pozice díky které bychom

mohli modelovat i složitější pohyby cílů.

V neposlední řadě by bylo zajímavým rozšířením aplikace takzvaného *ego-motion*, který slouží k odhadování pozice kamery vzhledem k cíli. Jeho hlavní výhodou je, že na základě získaných dat bychom mohli model cíle vhodně transformovat. To by nám umožnilo sledovat cíle prostřednictvím pohyblivé kamery s použitím složitějších modelů cílů než je schopna například metoda MS. Tím by se použitelnost rozšířila například i do oblasti robotiky.

Kapitola 6

Závěr

Cílem této bakalářské práce bylo nastudování problematicky sledování cílů, navržení řešení problému a jeho implementace. Při plnění těchto bodů se mi podařilo implementovat navržené řešení a dokázat jeho funkčnost na testovacích videozáznamech pocházejících jak z normální videokamery, tak i z termokamery. Pravě nutnost vypořádat se s odlišným chováním termokamery celou práci ozvláštnilo. Velký přínos práce spatřuji také v bližším seznámení se s principy zpracování obrazu a počítačového vidění. Dalším přínosem bylo obeznámení s knihovnou *OpenCV*, která nabízí množství do budoucna použitelných funkcí i mimo obory týkající se zpracování obrazu.

Veškerá nastudovaná teorie je zmíněna postupně ve druhé až čtvrté kapitole. V páté kapitole je pak popsána navržená implementace a shrnuty výsledky testů. Právě výsledky testů dokazují správnost a použitelnost navrženého postupu, který se skládal ze dvou vzájemně zaměnitelných metod pro detekci cíle posílených o Kalmanův filtr pro korekci výsledků měření a odhadnutí polohy cíle v dalším snímku. Závěr páté kapitoly pak patří nastínění a prezentaci návrhů pro další vývoj aplikace. Především zmiňuje potřebu lepšího detektoru objektů a obecnější metody pro predikci pohybu sledovaného cíle.

Literatura

[1] Comaniciu, D.; Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, volume 24, nr. 5, 2002: s. 603–619, ISSN 0162-8828.

[2] Comaniciu, D.; Ramesh, V.; Meer, P.: Kernel-Based Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, volume 25, nr. 5, 2003: s. 564–575, ISSN 0162-8828.

[3] G.Bradsky; A.Kaehler: *Learning OpenCV*. O'Reilly Media, Inc., 2008, ISBN 978-0-596-51613-0.

[4] J.R.Parker: *Algorithms for Image Processing and Computer Vision*. Wiley Computer Publishing, 1996, ISBN 0-471-14056-2.

[5] McIvor, A.: Background Subtraction Techniques. In *Image & Vision Computing New Zealand 2000 IVCNZ2000*, Auckland, New Zeland: Reveal Ltd., 2000.

[6] Mlích, J.: *Sledování objektů ve videosekvencích*. Vysoké učení technické v Brně, 2008, diplomová práce.

[7] M.Sonka; V.Hlavac; R.Boyle: *Image Processing, Analysis and Machine Vision*. Thomson Learning, third edition, 2008, ISBN 978-0-495-08252-1.

[8] P.Zemčík; M.Španěl: *Zpracování obrazu*. Vysoké učení technické v Brně, 2006, [Studijní opora].

[9] WWW stránky: Image Pre-processing: Geometric transformations. http://www.engineering.uiowa.edu/~dip/LECTURE/PreProcessing2.html, 2003-09-08 [cit. 2010-03-20].

[10] WWW stránky: Infrared Cameras adn Night Vision Cameras [online]. http://www.imaging1.com/, 2003 [cit. 2010-01-27].

[11] WWW stránky: How night vision works [online]. http://www.hownightvisionworks.com/, 2009 [cit. 2010-01-28].

[12] WWW stránky: OpenCV Wiki: Video Surveillance [online]. http://opencv.willowgarage.com/wiki/VideoSurveillance, 2009 [cit. 2010-05-10].

[13] WWW stránky: Photovoltaic effect [online]. http://en.wikipedia.org/wiki/Photovoltaic_effect, 2010-01-14 [cit. 2010-01-27].

[14] WWW stránky: Charge-coupled device [online]. http://en.wikipedia.org/wiki/Cmos, 2010-01-22 [cit. 2010-01-27].

[15] WWW stránky: Charge-coupled device [online]. http://en.wikipedia.org/wiki/Charge-coupled_device, 2010-01-23 [cit. 2010-01-27].

[16] WWW stránky: Photoelectric effect [online]. http://en.wikipedia.org/wiki/Photoelectric_effect, 2010-01-26 [cit. 2010-01-27].

[17] WWW stránky: Shannonův teorém [online]. http://cs.wikipedia.org/wiki/Shannon%C5%AFv_teor%C3%A9m, 2010-03-18 [cit. 2010-04-12].

[18] WWW stránky: RGB Color Model [online]. http://en.wikipedia.org/wiki/RGB_color_model, 2010-04-04 [cit. 2010-04-12].

[19] WWW stránky: Shluková analýza [online]. http://cs.wikipedia.org/wiki/Shlukov%C3%A1_anal%C3%BDza, 2010-04-06 [cit. 2010-04-30].

[20] WWW stránky: HSL and HSV [online]. http://en.wikipedia.org/wiki/HSV_color_space, 2010-04-08 [cit. 2010-04-12].

[21] WWW stránky: The Kalman Filter. http://www.cs.unc.edu/~welch/kalman/index.html#Anchor-Rudolph-6296, 2010-04-29 [cit. 2010-04-30].

[22] WWW stránky: BSD licenses [online]. http://en.wikipedia.org/wiki/BSD_license, 2010-05-10 [cit. 2010-05-8].

[23] WWW stránky: USMC, Operations, Night Vision [online]. http://www.visualintel.net/Marines/Operations/USMC-Operations-Night-Vision, 2010 [cit. 2010-01-27].

[24] WWW stránky: OpenCV Wiki [online]. http://opencv.willowgarage.com/wiki/, 2010 [cit. 2010-05-10].

[25] Yilmaz, A.; Javed, O.; Shah, M.: Object tracking: A survey. *ACM Comput. Surv.*, volume 38, nr. 4, 2006: str. 13, ISSN 0360-0300.

[26] Yilmaz, A.; Shafique, K.; Shah, M.: Target tracking in airborne forward looking infrared imagery. *Image and Vision Computing*, volume 21, nr. 7, 2003: s. 623 – 635, ISSN 0262-8856.

Příloha A

Obsah DVD

`\bin`	binární soubory
`\examples`	testovací videa
`\src`	zdrojový kód
`\text`	textová část práce