



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ROBOTICKÁ ZBRAŇOVÁ VĚŽ PRO HLÍDÁNÍ PERIMETRU –
DETEKCE CÍLE

ROBOTICS GUN TURRET FOR PERIMETER GUARDING – TARGET DETECTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Mykola Kaplenko

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Luděk Žalud, Ph.D.

BRNO 2016



Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Mykola Kaplenko

ID: 178038

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Robotická zbraňová věž pro hlídání perimetru - detekce cíle

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s konceptem automatické robotické zbraňové věže pro hlídání perimetru. Cílem je společně se studentem Dovalou Lubošem vytvořit plně funkční model.
2. Seznamte se s možnostmi detekce osob pomocí optoelektronických systémů. Zaměřte se především na barevné kamery a termovizní techniku.
3. Navrhněte systém označování cíle světelnou značkou (laserem) s možností detekce běžnou kamerou nebo kamerou upravenou z běžné kamery.
4. Vyberte vhodný algoritmus detekce značky a tento realizujte na Raspberry Pi a otestujte.
5. Navrhněte a vytvořte řídicí algoritmus, který na základě znalosti pozice značky namíří zbraň správným směrem a pošle signál k odbavení zbraně.
6. Společně se studentem Dovalou Lubošem zařízení otestujte, zaměřte se především na rychlost a přesnost reakce.

DOPORUČENÁ LITERATURA:

Yoshihiko Takahashi, Service Robot Applications, ISBN 978-953-7619-00-8, August 2008

Termín zadání: 6.2.2017

Termín odevzdání: 29.5.2017

Vedoucí práce: prof. Ing. Luděk Žalud, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Základem efektivní činnosti libovolného autonomního zařízení jsou řídicí algoritmy. Proto daná bakalářská práce je věnovaná problému řízení jednoho z nejperspektivnějších druhů zbraně – automatické zbraňové věže. V rámci dané práce je uveden přehled metod detekce objektu na snímku a jsou porovnány výhody a nevýhody použití jednotlivých metod při různých okolních podmínkách. Na základě provedené analýzy je navržen algoritmus detekce světelné značky (červeného bodu od laserového ukazovátka). Algoritmus řízení krokových motorů automatické zbraňové věže je navržen na základě provedených teoretických výpočtů. Pro uvedené algoritmy se nabízejí možnosti vylepšení a také se analyzují jejich silné a slabé stránky.

KLÍČOVÁ SLOVA

Počítačové vidění, detekce objektu, sledování objektu, Raspberry Pi, bod od laserového ukazovátka, openCV, C++

ABSTRACT

The basis for the efficient operation of any autonomous device are control algorithms. Therefore, the bachelor thesis is dedicated to the management problems one of the most promising types of weapon - automatic weapon towers. In the framework of the given work there is an overview of the object detection methods in the picture and comparing the advantages and disadvantages of using different methods in different surrounding conditions. On the basis of the analysis, the light mark detecting algorithm (red dot from the laser pointer) is designed. The stepper motor control algorithm of the automatic weapon tower is designed on the basis on the performed theoretical calculations. For the declared algorithms are offering possible improvements and analyzing their strengths and weaknesses.

KEYWORDS

Computer vision, object detection, object tracking, Raspberry Pi, point of the laser pointer, openCV, C++

KAPLENKO, Mykola. *Robotická zbraňová věž pro hlídání perimetru – detekce cíle*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2016. 32 s. Vedoucí práce: prof. Ing. Luděk Žalud, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou prací na téma „Robotická zbraňová věž pro hlídání perimetru – detekce cíle“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu své bakalářské práce panu prof. Luďku Žaludovi za odborné vedení a cenné rady při zpracování této bakalářské práce.

OBSAH

Úvod	1
1 Současný stav problematiky	2
1.1 Detekce osob pomocí optoelektronických systémů	3
1.1.1 Barevné kamery.....	3
1.1.2 Termovizní technika.....	3
1.2 Základní druhy zbraňových věží	3
1.2.1 Zbraňové věže bez algoritmů počítačového vidění	4
1.2.2 Zbraňové věže s automatickým rozeznáním cíle.....	4
1.2.3 Zbraňové věže s detekcí světelné značky	5
2 Popis modelu automatické věže pro hledání perimetru	6
2.1 Raspberry Pi 2B.....	6
2.2 Pi Camera	6
2.3 Knihovna OpenCV	6
2.3.1 Představení obrazů v knihovně OpenCV	7
2.3.2 Souřadnicový systém pro práci se zobrazeními	7
2.4 Barevné prostory	8
2.4.1 Prostor RGB	8
2.4.2 Prostor HSV	8
2.5 Wiring Pi	9
2.5.1 Základní knihovní funkce.....	9
3 Algoritmus namíření zbraně na základě znalosti pozice světelné značky	10
3.1 Teoretické výpočty	10
3.2 Implementace algoritmu namíření zbraně	11
3.3 Možnosti zlepšení algoritmu namíření zbraně	12
4 Metody stanovení polohy objektu na snímcích videa	13
4.1 Metody sledování objektů	13
4.1.1 Metody sledování zvláštních bodů (point trackich)	13
4.1.2 Metody sledování komponent (kernel tracking).....	14
4.2 Metody detekce objektu na každém snímku	14

4.2.1	Metody stanovení polohy objektu podle šablony	14
4.2.2	Detekce objektu metodami stanovení oblasti pohybu	15
4.2.2A	Stanovení oblasti pohybu pomocí odčítání pozadí	15
4.2.2B	Stanovení oblasti pohybu pomocí odčítání předchozího snímku od aktuálního	17
4.2.3	Hledání pixelu s určitou hodnotou parametrů	18
4.2.3A	Detekce objektu v prostoru RGB	18
4.2.3B	Detekce objektu v prostoru HSV	19
5	Návrh algoritmu detekce světelné značky	21
5.1	Popis jednotlivých kroků algoritmu	21
5.2	Výsledky testování navrženého algoritmu detekce světelné značky	24
5.2.1	Srovnání výsledků činnosti navrženého algoritmu a algoritmů zkoumaných v rámci dané práce	24
5.2.2	Úspěšnost detekce světelné značky	28
5.2.3	Rychlost zpracování snímků	28
5.2.4	Srovnání algoritmů použitých v semestrální a bakalářské práci	28
5.3	Možnosti zlepšení algoritmu detekce světelné značky	29
6	Závěr	30
	Literatura	31

SEZNAM OBRÁZKŮ

Obrázek 1	Redwolf airsoft gun.....	3
Obrázek 2	Pult dálkového ovládaní Redwolf airsoft gun.....	4
Obrázek 3	Military sentry gun “Adunok”.....	4
Obrázek 4	Detekce cíle ve vzdálenosti 500 m systémem “Adunok” pomocí kamery (levý obrázek) a teplovizoru (pravý obrázek).....	5
Obrázek 5	Použitý souřadnicový systém.....	7
Obrázek 6	Barevný prostor HSV.....	8
Obrázek 7	Namíření zbraně automatické věže.....	10
Obrázek 8	Snímek číslo 75 z testovacího videa.....	16
Obrázek 9	Zpracování snímku číslo 75 z testovacího videa metodou cv::BackgroundSubtractorMOG2.....	16
Obrázek 10	Použití metody odčítání dvou následujících snímků při stálém osvětlení okolí.....	17
Obrázek 11	Použití metody odčítání dvou následujících snímků při měnícím se osvětlení okolí.....	17
Obrázek 12	Výsledky použití metody hledání objektu v prostoru RGB za normálního (denního) osvětlení.....	19
Obrázek 13	Výsledky použití metody hledání objektu v prostoru RGB za slabého osvětlení.....	19
Obrázek 14	Výsledky použití metody hledání objektu v prostoru HSV na tmavém pozadí.....	20
Obrázek 15	Zpracování první sady snímků: zachycení dvou následujících snímků z testovacího videa.....	25
Obrázek 16	Zpracování první sady snímků: výsledek odčítání snímků.....	25
Obrázek 17	Zpracování první sady snímků: (a) Detekce světelné značky (b) Výsledek použití operace dilate.....	26
Obrázek 18	Zpracování druhé sady snímků: zachycení dvou následujících snímků z testovacího videa.....	26
Obrázek 19	Zpracování druhé sady snímků: výsledek odčítání snímků.....	27
Obrázek 20	Zpracování druhé sady snímků: (a) Detekce světelné značky (b) Výsledek použití operace dilate.....	27
Obrázek 21	Úspěšnost detekce světelné značky na bílém pozadí.....	28
Obrázek 22	Měření rychlosti zpracování snímků minipočítačem Raspberry Pi 2B...28	

ÚVOD

Každý den lidstvo objevuje a vyvíjí nové technologie, vítězí nad nemocemi, které se považovaly za nevléčitelné, zkoumá dno oceánu a brázdí vesmír, ale stále se nemůže zbavit nejtemnějších stran své duše. Ve 21. století touha jednotlivých lidí po vládě a bohatství stále nutí bojovat celé národy, politické a religi3zní názory se stávají příčinami teroristických činů a důvodem smrti nevinných lidí. Pokud se nezmění duchovní úroveň celého lidstva, ve světě bude existovat násilí bez ohledu na to, jaká zbraň je k dispozici, protože příčinou násilí je člověk, nikoliv zbraň. Kdyby ve dnešní době část lidstva odmítla použití zbraně, byla by zničena těmi, kteří považují násilí za jediný možný způsob dosažení cíle. A proto se vojenský průmysl musí vyvíjet, ale výhradně v cílech bezpečí a ochrany veřejného klidu.

V každé válce jsou nejdůležitější lidské životy, a proto se vojenský průmysl už dlouho zabývá problémem vytvoření autonomních bojových jednotek, jejichž zničení nezpůsobí smrt vojáků. Kromě toho, stroje mohou plnit určité úkoly mnohem lépe než lidi. K takovým strojům patří automatické zbraňové věže pro hledání perimetru, jejichž efektivnost a bezpečnost prací je závislá na použitých řídicích algoritmech.

Hlavním cílem práce je návrh algoritmů pro řízení automatické zbraňové věže. K určení souřadnic X a Y cíle je navržen algoritmus detekce světelné značky na snímcích, získaných pomocí staticky umístěné barevné kamery. K zpracování snímků je použita knihovna OpenCV v. 3.1. Pro řízení krokových motorů automatické zbraňové věže je navržen algoritmus namíření zbraně na základě znalosti pozice světelné značky. Jako výpočetní a řídicí jednotka je použit minipočítač Raspberry Pi 2B.

Daná bakalářská práce obsahuje veškeré teoretické podklady pro vytvoření plně funkčního modelu automatické zbraňové věže. S velkou pravděpodobností model bude realizován příští rok v rámci bakalářské práce kolegy, který se rozhodl rozvolnit studium, což neumožnilo prezentaci funkčního modelu tento rok.

1 SOUČASNÝ STAV PROBLEMATIKY

Automatická zbraňová věž – stacionární zařízení, schopné samostatně nebo pomocí dálkového ovládání plnit přidělené bojové úkoly. V úplně autonomním režimu (bez operátora) mohou v dnešní době efektivně pracovat jenom věže likvidující letící střely. Další druhy věží pro správnou a efektivní práci vyžadují komunikaci s operátorem. [1]

Výhodou použití zbraňových věží je to, že jsou jednoduše maskovatelné. Operátor se může nacházet v bezpečné vzdálenosti od zařízení, což umožňuje střídání operátorů bez demaskování palebné pozice a ničení bojových jednotek nepřítele bez ohrožení života operátora.

Největší nevýhodou zbraňových věží je nebezpečnost spojená s jejich použitím. Nedokonalý řídicí algoritmus věže, nedbalost nebo unavenost operátora mohou být příčinami nežádáných a nevratných následků.

Obecně zbraňová věž se skládá z bloku pozorování, bloku řízení a bloku zbraňového zabezpečení. [1]

Řízení může být realizováno pomocí drátového nebo bezdrátového spojení věže a zařízení operátora. Výhodou drátového spojení je rychlost komunikace. Nevýhodou pak omezená možnost pohybu operátora. Výhodou bezdrátového spojení je možnost řízení z jakéhokoliv bodu ležícího uvnitř zóny dosahu, ale nevýhodou je to, že nepřítel může teoreticky získat kontrolu nad zbraňovou věží.

Blok zbraňového zabezpečení se skládá ze zbraně, jejíž typ zaleží na tom, jaké úkoly má věž plnit, a z mechanismu nabíjení zbraně.

Blok pozorování je vybaven speciálním optoelektronickým zařízením: barevnými kamerami, termovizní technikou atd.

1.1 Detekce osob pomocí optoelektronických systémů

Problém detekce osob optoelektronickým zařízením zbraňové věže spočívá v tom, že nepřítel se bude snažit proniknout do chráněného perimetru nepozorovaně a s velkou pravděpodobností bude používat maskovací uniformu. V takovém případě strojové algoritmy nemohou zaručit detekci narušitele. Mnohem efektivnější možností hledání perimetru je společná práce algoritmů počítačového vidění a člověka, realizovaná například následujícím způsobem.

Blok pozorování zbraňové věže poskytuje operátorovi informaci o okolí ve formě vhodně upraveného zobrazení. Operátor na základě získané informace označuje cíl speciální světelnou značkou. Blok pozorování určuje souřadnice světelné značky. Na základě získaných souřadnic je namířena zbraň. Střelba se začíná okamžitě po namíření nebo po povolení od operátora.

1.1.1 Barevné kamery

K získání informace o okolí se úspěšně používají barevné kamery pracující ve viditelném spektru světla. Důležitým parametrem kamery je rozlišení (počet pixelů v zobrazení) poskytovaných snímků. Rozlišení snímků poskytovaných kamerou má být co největší, aby byla možnost korektně detekovat světelnou značku na zobrazení, ale zároveň platí, že zpracování snímků s větším rozlišením zabírá více času. Proto se volí kompromis mezi kvalitou snímku a rychlostí jeho zpracování. Nevýhodou použití barevných kamer je, že při zvláštních okolních podmínkách (děšť, sníh, mlha, špatné osvětlení atd.) není viditelné spektrum světla dostačující pro úspěšné detekování sil nepřítele.

1.1.2 Termovizní technika

Při zvláštních okolních podmínkách je vhodné použít termovizní techniky, která zachycuje záření ve středním (tepelném) infračerveném diapazonu ($0,7 - 1,5 \mu\text{m}$) a poskytuje informaci o rozložení teploty zkoumaného povrchu ve formě barevného zobrazení. Zdrojem infračerveného záření ve středním diapazonu jsou všechny objekty s teplotou větší než 0 K, ale obvykle termovizní technika je schopná detekovat záření objektů s teplotou větší než $-50 \text{ }^\circ\text{C}$.

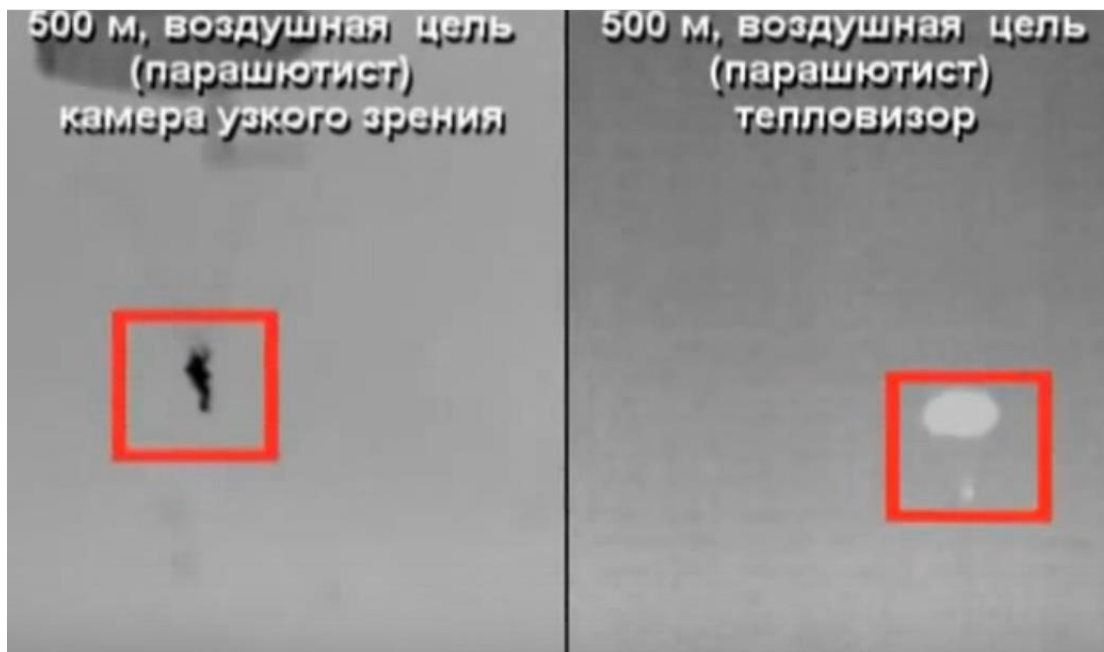
1.2 Základní druhy zbraňových věží

1.2.1 Zbraňové věže bez algoritmů počítačového vidění

Komunikace se uskutečňuje pomocí pultu dálkového ovládání. Zbraňová věž poskytuje operátorovi obraz z pozorovacího zařízení, umístěného na věži, a přijímá příkazy o tom, kam má mířit, kdy a jakými náboji střílet.



Obrázek 1 Redwolf airsoft gun



Obrázek 4 Detekce cíle ve vzdálenosti 500 m systémem “Adunok” pomocí kamery (levý obrázek) a teplovizoru (pravý obrázek)

1.2.3 Zbraňové věže s detekcí světelné značky

Tyto věže jsou druhem zbraňových věží s automatickým rozeznáním cíle. Princip je založen ne na detekci samotných sil nepřítele, ale na detekci světelné značky. Typ značky závisí na tom, pro jaké úkoly se používá a pomocí jakého zařízení má být detekována. Nejrozšířenějším typem světelných značek ve vojenském průmyslu jsou laserové značky.

2 POPIS MODELU AUTOMATICKÉ VĚŽE PRO HLEDÁNÍ PERIMETRU

Úkolem automatické zbraňové věže je hledání určité části perimetru. Proto je vhodné realizovat blok pozorování staticky umístěnou kamerou (pro realizaci úkolů modelu zbraňové věže není nutné použití termovizní techniky).

Na snímcích perimetru se detekuje poloha světelné značky. Nejlepší dostupnou variantou značky je bod od laserového ukazovátka. Má optimální rozměry a je velice jasný.

Na snímcích z infračervené kamery laserový bod pořád mění svou barvu, což neumožňuje detekci podle barvy. Proto pro plnění úkolů zbraňové věže byla zvolena barevná kamera (bez infračerveného spektru).

Jako zbraň automatické věže může být použita airsoftová pistole, která je řízena krokovými motory na základě znalosti pozice světelné značky.

2.1 Raspberry Pi 2B

Řídicí jednotka automatické zbraňové věže musí mít co nejmenší rozměry a hmotnost a při tom dostačující výpočetní výkonnost. Právě proto byl zvolen minipočítač Raspberry Pi 2B. Čtyřjádrový procesor Broadcom BCM2836 s frekvencí 900 MHz každého jádra, 1GB RAM paměť s rychlostí 450 MHz a grafický procesor VideoCore IV umožňují zpracování videa v reálném čase.

2.2 Pi Camera

Pi Camera je modul speciálně vyvinutý pro minipočítač Raspberry Pi. Umožňuje natočení videa s nastavením rozlišení jednotlivých snímků, počtu snímků za sekundu a nastavení výstupního formátu. Defaultní nastavení je: rozlišení 1920x1080 pixelů, 30 snímků za sekundu.

2.3 Knihovna OpenCV

OpenCV je knihovna realizující počítačové vidění. Je napsaná v jazyce C/C++, podporuje Windows, Linux, Mac OS, iOS a Android a má Python, Java, C a C++ interface. Knihovna podporuje multi-core processing. [2]

Pro realizaci úkolů dané práce byla zvolena knihovna OpenCV v. 3.1, protože obsahuje široké spektrum nástrojů umožňující zpracování snímků videa při použití výpočetní výkonnosti minipočítače Raspberry Pi 2B.

2.3.1 Představení obrazů v knihovně OpenCV

V digitální představě je obraz maticí číselných hodnot. Každý prvek matice popisuje jednotlivý pixel obrazu. V knihovně OpenCV je zobrazení popsáno třídou Mat s dvěma datovými částmi: záhlaví matice (obsahuje informaci o velikosti matice, o metodě používané pro uložení adresy matice atd.) a ukazatel na pole obsahující hodnoty pixelů. Velikost záhlaví matice je konstantní, avšak celková velikost matice se může lišit.

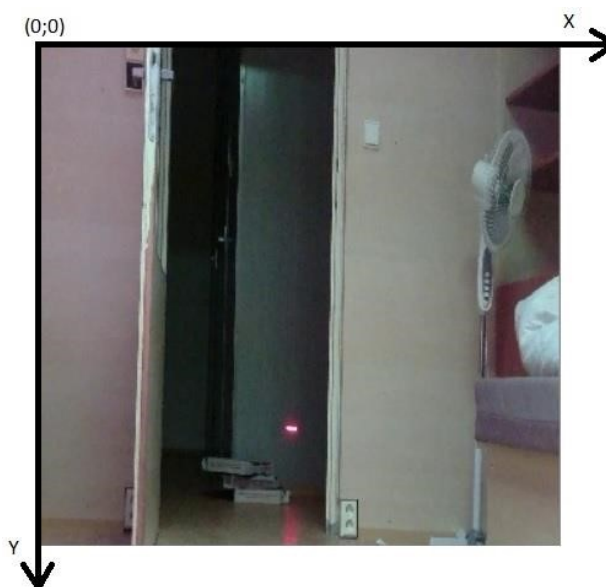
V dané práci je použita třída Mat pro uložení binárních a také osmibitových tříkanálových a jednonálových obrazů.

Každý pixel tříkanálového zobrazení je popsán třemi parametry, a proto matice reprezentující toto zobrazení je tvořena submaticemi [1x3]. Prvky matice, která odpovídá jednonálovému zobrazení, jsou velikosti [1x1], protože obsahují informaci jenom o jednom parametru – jasu (jednonálové zobrazení obsahuje jenom gradaci šedé barvy).

Každý z parametrů prvku matice reprezentující osmibitové zobrazení může nabývat hodnot 0 až 255 ($2^8 = 256$ různých hodnot). Prvek matice binárního zobrazení může nabývat jenom dvou hodnot (0 nebo 1), a proto binární obraz obsahuje jenom černé nebo bílé pixely.

2.3.2 Souřadnicový systém pro práci se zobrazeními

V dané práci poloha světelné značky je popsána souřadnicemi X a Y těžiště značky v pravoúhlém souřadnicovém systému, začátkem kterého je nejhornější a nejlevější pixel snímku. Rozlišení snímku je 1920x1080 pixelů. Bod, jehož souřadnice jsou (960,540) se označuje jako střed snímku.



Obrázek 5 Použitý souřadnicový systém

2.4 Barevné prostory

Barevný prostor je matematický popis množiny barev, který umožňuje představení barvy pomocí číselných hodnot. Barva každého pixelu zobrazení je popsána souřadnicemi v určitém barevném prostoru.

2.4.1 Prostor RGB

V barevném prostoru RGB každá barva je představena pomocí sčítání základních barev – červené (Red), zelené (Green) a modré (Blue). V číslkové podobě každá ze základních barev je reprezentovaná hodnotou od 0 do 255. Takže souřadnice barev, představených v prostoru RGB, se nacházejí v rozmezí (0,0,0) až (255,255,255). Prostor RGB se často používá v zobrazovacích zařízeních. V knihovně OpenCV jsou obrazy představeny v prostoru BGR, který se liší od prostoru RGB jenom pořadím jednotlivých parametrů.

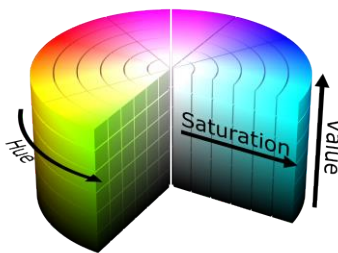
2.4.2 Prostor HSV

V prostoru HSV je každá barva představena pomocí parametrů Hue, Saturation a Value.

Hue – převládající barevný tón neboli odstín. Obecně se odstín označuje názvem barvy a měří se jako poloha na standardní barevné kružnici (0° až 360°).

Saturation – sytost barvy neboli čistota barvy. Měří se v rozmezí od 0 % (šedá barva) do 100 % (plně sytá barva).

Value – hodnota jasu. Měří se od 0 (bez ohledu na parametry Hue a Saturation barva s hodnotou parametru Value 0 je černou) do 100 (jasná barva).



Obrázek 6 Barevný prostor HSV

2.5 WiringPi

WiringPi je knihovna pro práci s porty GPIO minipočítače Raspberry Pi. Umožňuje práci s I2C, SPI a UART.

2.5.1 Základní knihovní funkce

```
void pinMode (int pin, int mode)
```

Nastavení režimu pinu. Možné hodnoty parametru mode: INPUT, OUTPUT, PWM_OUTPUT, GPIO_CLOCK. Režim PWM_OUTPUT podporuje jenom pin BCM_GPIO 18, režim GPIO_CLOCK – jenom pin BCM_GPIO 4.

```
void pullUpDnControl (int pin, int pud)
```

Zapojení vnitřních rezistorů pro pin pracující v režimu INPUT.

```
void digitalWrite (int pin, int value)
```

Nastavení vysoké (value = 1) nebo nízké (value = 0) úrovně na pinu pracujícím v režimu OUTPUT.

```
void pwmWrite (int pin, int value)
```

Parametr value určuje koeficient plnění PWM. Může nabývat hodnot v rozmezí 0 (0%) až 1024 (100%). PWM podporuje jenom pin BCM_GPIO 18.

```
int digitalRead (int pin)
```

Funkce vrací 0, pokud není napětí na pinu, nebo 1, pokud napětí na pinu je.

3 Algoritmus namíření zbraně na základě znalosti pozice světelné značky

Cílem algoritmu je stanovení počtu kroků, které musí udělat krokové motory, aby zbraň byla otočena z libovolné aktuální polohy do takové polohy, ve které je zbraň namířena do světelné značky. Přesnost nastavení polohy zbraně je daná krokem motoru. Při střelbě na vzdálenost 5 m musí být uvedený algoritmus realizován s použitím krokových motorů s uhlovým krokem nejvíc $0,12^\circ$ (alespoň 3000 kroků za otáčku) – pak jeden krok motoru způsobí posun bodu zásahu střelou o 1 cm.

Zbraň dostane signál k odpálení střely, pokud vzdálenost mezi bodem, kam zbraň míří, a bodem, který má být trefen, bude menší než vzdálenost odpovídající jednomu kroku motoru (v každé z os: X a Y).

y – je počet kroků nutných na to, aby zbraň byla otočena z aktuální pozice do pozice, ve které je namířena do světelné značky.

Dále platí:

$$\frac{H_{pož}}{S} = tg(\alpha_{akt} + y \cdot \alpha_{min}) \quad (3.4)$$

kde $H_{pož}$ [px] – je vzdálenost od středu snímku do místa, kde se aktuálně nachází světelná značka (vzdálenost C_2C_4 na obrázku – “požadovaná vzdálenost”).

$$\alpha_{akt} = n \cdot \alpha_{min} \quad (3.5)$$

kde n – je počet kroků nutných na to, aby zbraň byla otočena o úhel α_{akt} .

Po dosazení rovnice 3.5 do rovnice 3.4 a následných úpravách platí:

$$y = \frac{1}{\alpha_{min}} \arctg\left(\frac{H_{pož}}{S}\right) - n \quad (3.6)$$

kde α_{min} a S jsou konstanty.

Takže počet kroků, které musí být udělány pro namíření zbraně, je závislý na aktuální poloze světelné značky a na počtu již udělaných kroků.

Obdobným způsobem může být nalezen i počet kroků nutných pro namíření zbraně v horizontální rovině.

3.2 Implementace algoritmu namíření zbraně

Algoritmus je implementován pomocí následujících kroků:

1. Zapnutí Raspberry Pi, začátek plnění řídicího programu.
2. Kontrola, jestli zbraň je namířena do středu zóny viditelnosti kamery (jestli úhel natočení obou řídicích motorů je nulový). Pokud ne, tak zbraň bude vycentrována.

Mechanismus vycentrování zbraně bude realizován obdobně jako mechanismus nastavení aktuálního času v Raspberry Pi. Minipočítač nemá hodiny reálného času, které by běžely, když je Raspberry Pi vypnutý. A proto Raspberry ukládá aktuální čas při vypnutí; při zapnutí začíná čas běžet od poslední uložené hodnoty. Obdobně při vypnutí Raspberry bude uložen aktuální počet kroků, který udělal každý z motorů. Při zapnutí motory udělají uložený počet kroků v opačném směru. Výsledkem je vycentrování zbraně a nastavení hodnoty aktuálního počtu udělaných kroků na nulu.

3. Algoritmus detekce světelné značky vyhodnocuje polohu cíle a vrací její souřadnice, které se ukládají do proměnných s názvy X a Y .
4. Výpočet hodnot proměnných nutných pro namíření zbraně.

Souřadnice X a Y se počítají od začátku souřadnicového systému (bod (0;0)), zatímco $H_{požX}$ a $H_{požY}$ se počítají od středu snímku (bod (960;540)), a proto “požadované vzdálenosti [px]” se přepočítají podle následujících rovnic:

$$\begin{aligned}H_{požX} &= X - 960 \\H_{požY} &= Y - 540\end{aligned}$$

Proměnné $H_{požX}$ a $H_{požY}$ mohou nabývat kladných i záporných hodnot. Směr otáčení motorů je určen znaménkem proměnných.

Stanovení počtu kroků motorů a uložení do proměnných s názvy x a y :

$$\begin{aligned}x &= \frac{1}{\alpha_{min}} \arctg\left(\frac{H_{požX}}{S}\right) - nhor \\y &= \frac{1}{\alpha_{min}} \arctg\left(\frac{H_{požY}}{S}\right) - nver\end{aligned}$$

kde $nhor$ a $nver$ jsou proměnné, ve kterých je uložen aktuální počet udělaných kroků v horizontální a vertikální ose.

Výraz $\frac{1}{\alpha_{min}} \arctg\left(\frac{H_{pož}}{S}\right)$ je zaokrouhlen na nejbližší celé číslo. Pokud se x a y současně rovnají nule, pak zbraň udělá výstřel.

5. Řízení motorů na základě hodnot proměnných x a y .

Krokové motory dostanou příkaz udělat určitý počet kroků. Hodnota proměnných $nhor$ a $nver$ se obnovuje po každém kroku motorů. Pokud dojde k vypnutí Raspberry, aktuální hodnoty proměnných $nhor$ a $nver$ se uloží a při následujícím zapnutí Raspberry se zbraň normálně vycentruje.

6. Vrácení k bodu číslo 3. Body číslo 3 až 6 se opakují, dokud Raspberry nebude vypnutý.

3.3 Možnosti zlepšení algoritmu namíření zbraně

Hlavní nevýhodou algoritmu je to, že se nejdřív zpracovává snímek z kamery a pak se uskutečňuje řízení motorů, během kterého se snímky z kamery nezpracovávají. Dochází ke ztrátě času a zpomalení práce algoritmu. Řešením problému může být použití dvou minipočítačů Raspberry Pi (popř. použití Rasperry Pi a Arduino). Jeden minipočítač by zpracovával snímky z kamery a posílal data druhému minipočítači, který by se zabýval řízením motorů. Výsledkem je to, že jakmile snímek bude zpracován, motory začnou pohyb k aktuální poloze světelné značky z aktuálního úhlu natočení. Během této doby se už zpracovává další snímek. Oba procesy běží současně a nedochází ke ztrátě času.

4 METODY STANOVENÍ POLOHY OBJEKTU NA SNÍMCÍCH VIDEA

Poloha objektu na snímcích videa může být stanovena dvěma způsoby:

- Objekt se detekuje na každém snímku. Výsledkem zpracování snímku jsou souřadnice bodu odpovídajícího objektu (třeba těžiště objektu, střed obdélníku opsaného kolem objektu atd.).

- Detekce objektu na prvním snímku a jeho sledování na dalších snímcích. Výsledkem je trajektorie pohybu objektu, popsána polem souřadnic, kterými procházel objekt.

Oba způsoby jsou popsány v následujících kapitolách.

4.1 Metody sledování objektů

Algoritmus sledování zajišťuje sestavení trajektorie pohybu cílových objektů na vstupní posloupnosti snímků. Většinou se používají dva druhy metod sledování objektů:

- Metody sledování zvláštních bodů (point tracking)
- Metody sledování komponent (kernel tracking)

4.1.1 Metody sledování zvláštních bodů (point tracking)

Nutnou podmínkou použití metod této skupiny je existence označených bodů, které nejlépe popisují objekt (uhly, hrany atd.). Zpracování zvláštních bodů se uskutečňuje při použití detektorů a deskriptorů. Detektor slouží k nalezení poloh zvláštních bodů na obrázku. Pokud zvláštní bod byl nalezen, deskriptor sestaví vektor příznaků tohoto bodu.

Tracker (algoritmus sledování) zvláštních bodů funguje podle následujících kroků [3]:

1. Hledání zvláštních bodů na aktuálním snímku videa pomocí zvoleného detektoru.
2. Výpočet deskriptorů získané sady bodů.
3. Porovnání (matching) deskriptorů získaných na aktuálním a předchozím snímku.

Na základě výsledků porovnání deskriptorů je posouzen pohyb objektů, reprezentovaných sadami zvláštních bodů.

Nejčastěji používanými jsou SIFT [4] (Scale Invariant Feature Transform), SURF (Speeded up Robust Features) [5], GLOH (Gradient Location-Orientation Histogram) [6], DAISY [7], BRIEF (Binary Robust Independent Elementary Features) [8] a ORB [9] deskriptory. Příkladem algoritmů sledování zvláštních bodů mohou být Lucas – Kanade, Tomasi – Kanade a Shi – Tomasi – Kanade algoritmy [2].

Metody sledování efektivně pracují s pohybem velkých objektů, které obsahují velký počet zvláštních bodů. Sledování pohybu malých objektů, jako je bod od laserového ukazovátka, je dost problematické.

4.1.2 Metody sledování komponent (kernel tracking)

Komponenta je forma objektu nebo jeho vnější podoba. V nejjednodušším případě komponenta může být reprezentována šablonou obdélníkové nebo eliptické formy, v složitějších případech – trojrozměrným modelem objektu. Obecně metody dané skupiny se používají, když pohybem je obyčejný posun, otáčení nebo afinní transformace. Příkladem metody sledování komponent je metoda CAM Shift.

V případě hledání bodu od laserového ukazovátka metody sledování komponent nemohou být použity, protože se na snímcích vnější podoba světelné značky pořád mění a při rychlém pohybu laserový bod vypadá jako úsečka.

Obecně metody sledování jsou výpočetně náročné, a proto při hledání laserového bodu je rychlejší tento bod detekovat na každém snímku videa zvlášť, než ho sledovat na posloupnosti snímků. Proto metody sledování se nebudou brát v úvahu při návrhu algoritmu detekce světelné značky automatickou zbraňovou věží.

4.2 Metody detekce objektu na každém snímku

4.2.1 Metody stanovení polohy objektu podle šablony

Hledání objektů na základě určité šablony předpokládá, že existuje zobrazení objektu se stanovenými příznaky – šablona – a testované zobrazení, které se porovnává s touto šablonou podle zvolené metriky. V nejjednodušším případě je šablonou matice intenzit barev, nejvíc příznačných objektu. Složitější metody zkoumané skupiny jako šablonu používají sady vektorů příznaků (deskriptory), geometrické představení objektů [10] nebo pravděpodobnostní modely objektů, které obsahují informaci o rozložení intenzit pixelů [11]. Výsledkem práce metod dané skupiny je detekce objektu odpovídajícího šabloně na každém snímku videa.

Metody hledání podle šablony efektivně pracují s jednotlivými objekty, jejichž tvar se nemění (při vzniku překrytí objektů nebo při změně formy objektu zanikají některé příznaky v popisu). [12]

V případě detekce laserového bodu je problémem to, že při rychlém pohybu světelná značka na snímku nevypadá jako bod, ale jako křivka nebo úsečka. A proto metoda hledání světelné značky podle šablony nemůže být v daném případě efektivně použita.

4.2.2 Detekce objektu metodami stanovení oblasti pohybu

Úkol stanovení oblasti pohybu na videu je jedním z klasických úkolů počítačového vidění. Protože se má sledovat změna okolí s časem, je vhodné použít mechanismus odčítání snímků. Výsledkem odčítání je zobrazení, na kterém nenulovými (nečernými v prostoru RGB) jsou jenom ty pixely, jejichž hodnota se změnila.

4.2.2A Stanovení oblasti pohybu pomocí odčítání pozadí

Nejjednodušším řešením je použití mechanismu odčítání pozadí od snímku videa (background subtraction). Procedura odčítání předpokládá, že pro dané video je sestrojen model pozadí a také, že existuje mechanismus změny modelu pozadí s časem. [13] Pro jednocanálové zobrazení je možno stanovení oblasti pohybu rozdělit na dvě části:

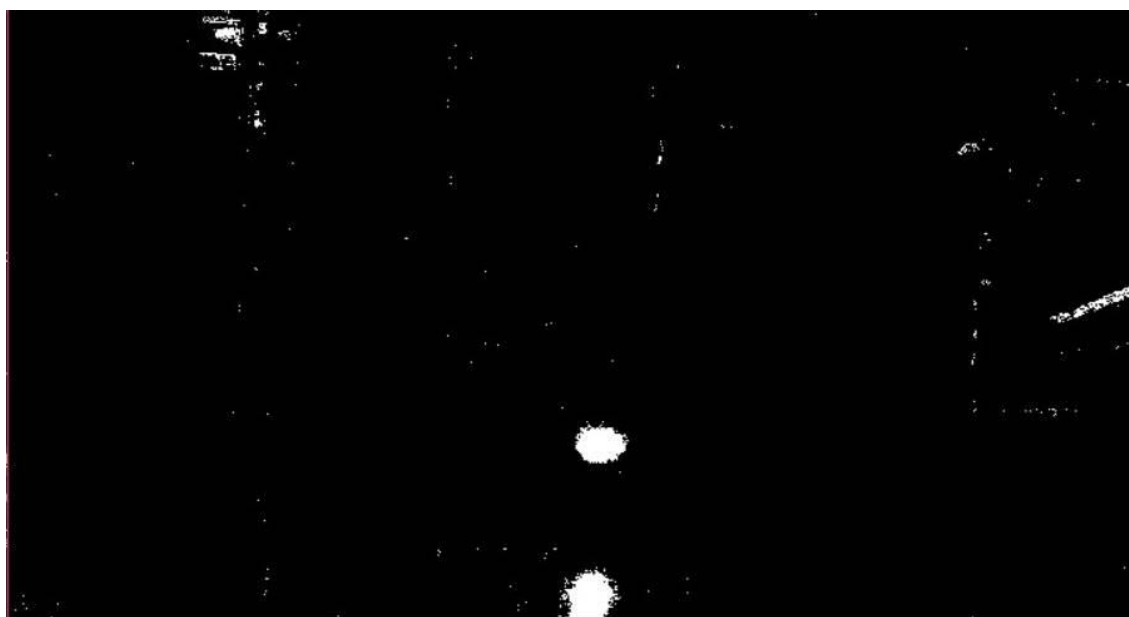
1. Odčítání pozadí od aktuálního snímku videa. Na tomto kroku se uskutečňuje odčítání intenzit snímku videa a modelu pozadí po jednotlivých pixelech.
2. Výběr pixelů patřících objektu – sestrojení binárního zobrazení (maska). Uvažuje se, že pixel patří objektu a má bílou barvu v masce, když rozdíl intenzit aktuálního snímku a pozadí pro daný pixel překračuje některou prahovou hodnotu.

Navíc k uvedeným operacím za účelem zvýšení kvality hledání může být použita třeba filtrace snímků videa nebo také mohou být použity morfologické operace k výsledku odčítání. [14]

Kvalita stanovení oblasti pohybu pomocí odčítání je v mnohém závislá na kvalitě sestrojeného modelu pozadí, a proto existují metody, které obnovují model pozadí s každým následujícím snímkem. V dané práci byla otestována metoda vytvoření binární masky popředí (pomocí odčítání pozadí) cv::BackgroundSubtractorMOG2. Daná metoda byla zvolena pro testování, protože zpracování snímků je poměrně rychlé a protože automatická zbraňová věž pro hledání perimetru pracuje pořád se stejným pozadím, a proto má dostatek času, aby pomocí obnovení byl získán kvalitní model pozadí. Výsledky zpracování snímku ze speciálně natočeného videa modulem Pi Camera jsou uvedeny na obrázcích.



Obrázek 8 Snímek číslo 75 z testovacího videa



Obrázek 9 Zpracování snímku číslo 75 z testovacího videa metodou `cv::BackgroundSubtractorMOG2`

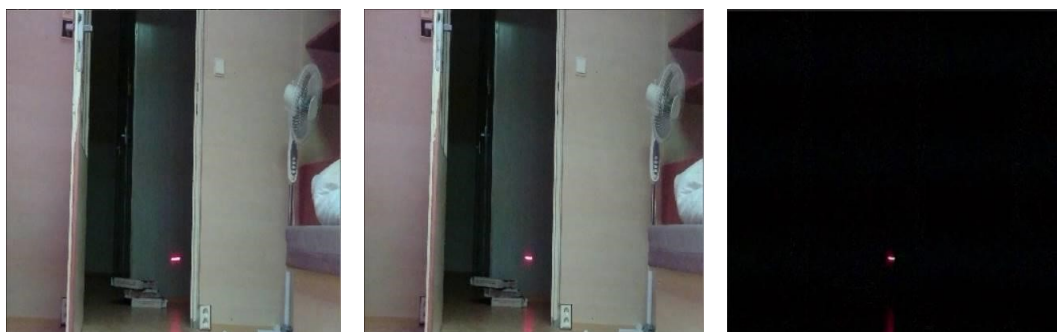
K momentu zpracování snímku byl model pozadí obnoven 74krát, ale výstupní binární obrázek je dost zašuměný a obsahuje pixely odpovídající odrazu světelné značky od lesklého povrchu (dolní část obrázku). Výsledky detekce světelné značky nejsou uspokojivé, a proto metoda odčítání pozadí nebude součástí algoritmu navrženého v dané práci.

4.2.2B Stanovení oblasti pohybu pomocí odčítání předchozího snímku od aktuálního

Dalším často používaným způsobem řešení úkolu detekce oblasti pohybu je odčítání dvou následujících snímků a binarizace výsledku odčítání s určitou prahovou hodnotou.

Nevýhodou použití tohoto způsobu je obtížná volba prahové hodnoty binarizace: pokud zvolená hodnota je příliš malá, binární zobrazení bude obsahovat šumy, a pokud zvolená hodnota prahu binarizace je příliš vysoká, pak na binárním zobrazení nebudou přítomné některé pixely odpovídající hledané oblasti pohybu.

Výhodou algoritmu odčítání dvou následujících snímků je rychlost, a proto se může používat v kombinaci s dalšími rychlými metodami pro detekci objektu v obrázku. Výsledky odčítání snímků natočeného testovacího videa jsou na následujících obrázcích.



Obrázek 10 Použití metody odčítání dvou následujících snímků při stálém osvětlení okolí



Obrázek 11 Použití metody odčítání dvou následujících snímků při měnícím se osvětlení okolí

Součástí navrženého algoritmu detekce světelné značky je metoda odčítání dvou následujících snímků, která je použita pro předzpracování zobrazení, na kterém se dál detekuje poloha bodu od laserového ukazovátka.

4.2.3 Hledání pixelů s určitou hodnotou parametrů

V metodách této skupiny se uvažuje, že pixel patří objektu, pokud parametry pixelu vyhovují určitým podmínkám. Tyto metody jsou rychlé a při vhodném předzpracování snímků dávají dobré výsledky. [15] Proto dvě metody ze zkoumané skupiny byly zvoleny pro testování a ověřování jejich efektivity.

4.2.3A Detekce objektu v prostoru RGB

Protože každý pixel je určen třemi parametry, je vhodné barevné zobrazení rozdělit do tří vrstev, ze kterých každá bude obsahovat informaci jenom o jednom z parametrů: R, G nebo B. V jednotlivých vrstvách budou pixely popsány submaticemi $[1 \times 1]$, které obsahují hodnotu odpovídajícího parametru.

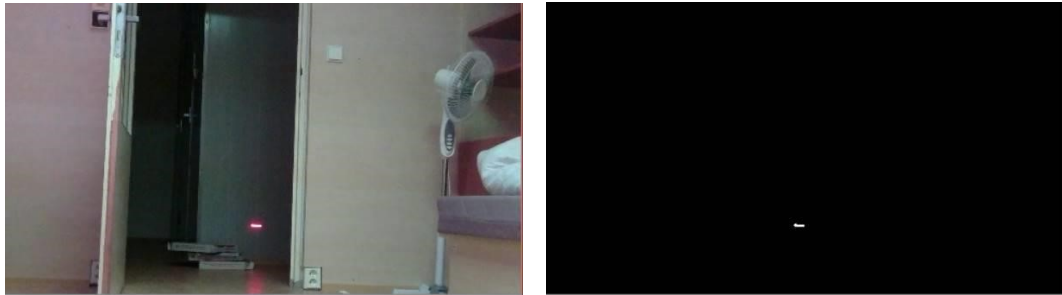
Snímky jsou osmibitové tříkanálové, a proto po rozdělení každá z jednotlivých vrstev (R, G, B) je reprezentována osmibitovým jednokanálovým zobrazením (gradace šedé barvy).

Hledání podle barvy se uskutečňuje tak, že se v jednotlivých vrstvách vybírají pixely s určitými hodnotami parametrů. K tomuto účelu se inicializují proměnné $Rmin$, $Rmax$, $Gmin$, $Gmax$, $Bmin$, $Bmax$, popisující dolní a horní hranice intervalů hodnot parametrů. Pixel jednokanálového zobrazení vyhovuje podmínkám hledání, když jeho parametr R (příp. G, B) ve vrstvě R (příp. G, B) patří do intervalu $\langle Rmin, Rmax \rangle$ (příp. $\langle Gmin, Gmax \rangle$, $\langle Bmin, Bmax \rangle$).

Dalším krokem je binarizace každé z jednotlivých vrstev: pixelům, které vyhovují podmínkám hledání, je přiřazena hodnota 1 (na binárním zobrazení jsou tyto pixely bílé), všem ostatním pixelům hodnota 0 (na zobrazení jsou pixely černé). Tím pádem jsou získány tři binární zobrazení, které se následně slučují pomocí logické operace AND (&). Výsledkem je jedno binární zobrazení, jehož bílým pixelům odpovídají hledané objekty na snímku.

V případě hledání bodu od červeného laserového ukazovátka na snímku musí být vybrány pixely s velkou hodnotou parametru R a menšími hodnotami parametrů G a B. Problém je ale v tom, že kromě bodu od laserového ukazovátka, najdeme i všechny červené objekty na snímku. Další nevýhodou algoritmu je to, že výsledky hledání jsou velice závislé na osvětlení.

Výhodou algoritmu je jeho rychlost.



Obrázek 12 Výsledky použití metody hledání objektu v prostoru RGB za normálního (denního) osvětlení



Obrázek 13 Výsledky použití metody hledání objektu v prostoru RGB za slabého osvětlení

Jak je vidět z obrázků, popsaná metoda funguje dobře při obyčejném denním osvětlení: na zpracovaném obrázku (obr.12) není odraz laserového bodu od podlahy. V případě, že se na snímku objeví další jasně červený objekt (statický nebo se pohybující), výstupní obrázek obsahuje nejenom hledaný bod od laserového ukazovátka.

4.2.3B Detekce objektu v prostoru HSV

Metoda se liší od hledání v prostoru RGB tím, že nyní je každý pixel popsán parametry H, S, V. Hledání pixelu s určitými parametry se realizuje obdobně jako v prostoru RGB. Nejdřív je nutno transformovat zobrazení z prostoru BGR do prostoru HSV a pak rozdělit do tří vrstev – H, S a V. Dále se jednotlivé vrstvy binarizují na základě toho, které z pixelů patří příslušným intervalům $\langle H_{min}, H_{max} \rangle$, $\langle S_{min}, S_{max} \rangle$ a $\langle V_{min}, V_{max} \rangle$. Následně tři získané binární zobrazení jsou sloučena pomocí logické operace AND do jednoho binárního zobrazení obsahujícího informaci o pixelech, které odpovídají hledaným objektům na snímcích.

Bod od červeného laserového ukazovátka je velice jasný, a proto na barevném snímku z kamery je reprezentován pixely s velkou hodnotou parametru V. Hodnota parametru H hledaných pixelů musí odpovídat červenému odstínu. Hodnota parametru S (sytnost) se může měnit v širokém rozmezí v závislosti na osvětlení. Bod od laserového ukazovátka je sytě červený, pokud se nachází na tmavém pozadí (velká hodnota S). Pokud se nachází na světlém pozadí, pak je na snímcích jeho střed skoro bílý (malá hodnota S).

Hledání červeného bodu od ukazovátka v prostoru HSV umožňuje detekci všech červených objektů mezi objekty, které jsou nejjasnější, na snímcích získaných pomocí kamery.

Tento algoritmus je mnohem odolnější změně osvětlení než detekce v prostoru RGB, ale při velmi světlém pozadí je dost problematické určit polohu světelné značky. Výhodou algoritmu je opět jeho rychlost.



Obrázek 14 Výsledky použití metody hledání objektu v prostoru HSV na tmavém pozadí

Z obrázků je patrné, že metoda hledání objektu v prostoru HSV poskytuje lepší výsledky než metoda hledání v prostoru RGB.

Zpracovaný obrázek obsahuje pixely odpovídající odrazu světla od dveří. Tento problém se nedá vyřešit zvětšením prahu binarizace, protože dané pixely mají stejný jas jako laserový bod. Existuje ale jiné řešení: problém může být odstraněn hledáním světelné značky na zobrazení, které je dáno odčítáním dvou následujících snímků, protože v daném případě je nežádoucí nalezený objekt (dveře) statický.

5 NÁVRH ALGORITMU DETEKCE SVĚTELNÉ ZNAČKY

Algoritmus je kombinací metody odčítání dvou následujících snímků videa a metody detekce objektu na základě výběru pixelů s vhodnou hodnotou parametrů v prostoru HSV.

5.1 Popis jednotlivých kroků algoritmu

Algoritmus detekce bodu od laserového ukazovátka je realizován pomocí následujících kroků:

1. Zapojení knihoven obsahujících C++ a OpenCV funkce.

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
```

2. Zapojení jmenných prostorů std a cv obsahujících používanou funkcionalitu.

```
using namespace cv;
using namespace std;
```

3. Deklarace proměnných, které budou použity pro uložení zobrazení.

```
Mat frame_previous;
Mat frame_actual;

Mat dst;
Mat hsv1;
Mat hsv2;

vector<Mat> hsvChannels(3);

Mat hsv_and;
Mat h_range;
Mat s_range;
Mat v_range;

Mat dilate_im;
```

4. Inicializace proměnných nutných pro výpočet těžiště světelné značky.

```
int Xc = 0;
int Yc = 0;
int counter = 0;
int X=0;
int Y=0;
```

5. Inicializace proměnných určujících dolní a horní hranice intervalů hodnot parametrů H, S a V. Proměnné mohou nabývat hodnot v rozmezí <0;255> (256 různých hodnot).

```
int Hmin = 1;
int Hmax = 20;
int Smin = 34;
int Smax = 255;
int Vmin = 50;
int Vmax = 255;
```

Pokud se na aktuálním snímku světelná značka přemístila do pixelů, které na předchozím snímku popisovaly pozadí, pak dochází ke změně hodnot parametrů těchto pixelů.

Experimentálně je stanoveno, že přemístění červeného laserového bodu na bílém pozadí při denním osvětlení způsobuje změnu parametrů pixelů v rozmezí:

H (ton): 1 až 20 bodů z 256 možných.

S (syty): 34 až 255 bodů z 256 možných.

V (jas): 50 až 255 bodů z 256 možných.

Pokud změna tónu je menší než 1 nebo větší než 20, změna sytosti je menší než 34, změna jasu je menší než 50, tak to znamená, že změna nebyla způsobena přemístěním světelné značky, ale vlivem dalších objektů na snímku.

6. Otevření implicitní kamery (Pi Camera).

```
VideoCapture capture(0);
```

7. Vytvoření strukturujícího elementu pro uskutečnění morfologických operací.

```
Mat Kern;
Kern = getStructuringElement(MORPH_RECT, Size(4, 4), Point(-1, -1));
```

8. Zachycení prvního snímku z kamery.

```
capture >> frame_previous;
```

9. Začátek nekonečného cyklu, ve kterém se budou zpracovávat snímky z kamery.

```
while (true) {
```

10. Zachycení dalšího (aktuálního) snímku z kamery.

```
capture >> frame_actual;
```

11. Pokud aktuální nebo předchozí snímek nebyl zachycen, vystoupíme z cyklu a ukončíme program.

```
if (frame_actual.empty() || frame_previous.empty())
{
break;
}
```

12. Transformace získaných snímků z prostoru BGR do prostoru HSV.

```
cvtColor(frame_previous, hsv1, CV_BGR2HSV);  
cvtColor(frame_actual, hsv2, CV_BGR2HSV);
```

13. Realizace metody odčítání dvou následujících snímků – odčítání snímků v prostoru HSV pixel po pixelu.

```
subtract(hsv2, hsv1, dst);
```

Hodnoty parametrů pixelů zobrazení *dst* jsou dány rozdílem hodnot parametrů pixelů zobrazení *hsv2* a *hsv1*.

14. Rozdělení do tří vrstev zobrazení, daného výsledkem odčítání snímků.

```
split(dst, hsvChannels);
```

15. Binarizace jednotlivých vrstev rozděleného obrázku s určitými prahovými hodnotami.

```
inRange(hsvChannels[0], Scalar(Hmin), Scalar(Hmax), h_range);  
inRange(hsvChannels[1], Scalar(Smin), Scalar(Smax), s_range);  
inRange(hsvChannels[2], Scalar(Vmin), Scalar(Vmax), v_range);
```

Na binárním zobrazení *h_range* (příp. *s_range*, *v_range*) budou bílými jenom ty pixely, ve kterých se parametr H (příp. S, V) změnil o hodnotu z intervalu $\langle Hmin, Hmax \rangle$ (příp. $\langle Smin, Smax \rangle$, $\langle Vmin, Vmax \rangle$).

16. Provedení logické operace AND pro jednotlivé pixely mezi výsledky binarizace jednotlivých vrstev (na výsledném binárním zobrazení *hsv_and* budou bílé ty pixely, které ve všech třech vrstvách byly bílé).

```
bitwise_and(h_range, s_range, hsv_and);  
bitwise_and(hsv_and, v_range, hsv_and);
```

17. Použití operace dilate – rozšíření (zvětšení) světlých oblastí na zobrazení. Na zpracovaných snímcích se občas stává, že mezi bílými pixely, které odpovídají světelné značce se vyskytují pixely černé (světelná značka je rozdělena černými pixely na několik částí). Pro sloučení sousedních bílých pixelů se používá morfologická operace dilate.

```
dilate(hsv_and, dilate_im, Kern, Point(-1, -1), 3);
```

18. Výpočet koordinát těžiště detekovaného laserového bodu. Pokud na zpracovaném snímku ani jeden pixel nebude bílý (světelná značka není nalezena), v proměnných X a Y zůstanou souřadnice těžiště světelné značky, která byla úspěšně detekována na předchozím snímku.

```
Xc = 0;  
Yc = 0;  
counter = 0;  
  
for (int y = 0; y < dilate_im.rows; y++) {
```

```

uchar* ptr = (uchar*)(dilate_im.data + y * dilate_im.step);

for (int x = 0; x<dilate_im.cols; x++)
{
    if (ptr[x]>0)
    {
        Xc += x;
        Yc += y;
        counter++;
    }
}

if (counter != 0)
{
    X = int(float(Xc) / counter);
    Y = int(float(Yc) / counter);
}

waitKey(1);

```

19. Hluboké kopírování snímku. Aktuální snímek se stává předchozím. Nový aktuální snímek bude zachycen v další iteraci cyklu.

```
frame_previous=frame_actual.clone();
```

20. Konec nekonečného cyklu while.

```
}
```

21. Ukončení programu.

```
return 0;
}
```

5.2 Výsledky testování navrženého algoritmu detekce světelné značky

5.2.1 Srovnání výsledků činnosti navrženého algoritmu a algoritmů zkoumaných v rámci dané práce

V této podkapitole jsou uvedeny výsledky jednotlivých stádií detekce laserového bodu v situacích, v nichž algoritmy popsané v kapitole 4 nedokázaly světelnou značku jednoznačně detekovat.

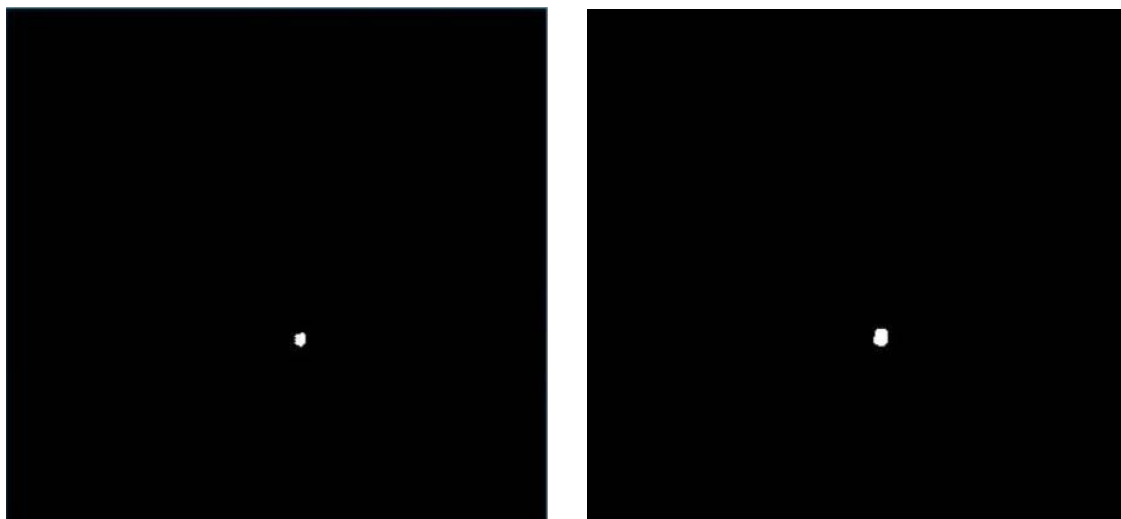
První sada snímků znázorňuje detekci laserového bodu při nepodstatné změně denního osvětlení, což se běžně může stát při použití algoritmu v praxi. V dané situaci výsledkem použití pouze metod stanovení oblasti pohybu bylo zašuměné zobrazení, zatímco výsledkem použití navrženého algoritmu je jednoznačně detekovaný objekt.



Obrázek 15 Zpracování první sady snímků:
Zachycení dvou následujících snímků z testovacího videa



Obrázek 16 Zpracování první sady snímků:
Výsledek odčítání snímků



a)

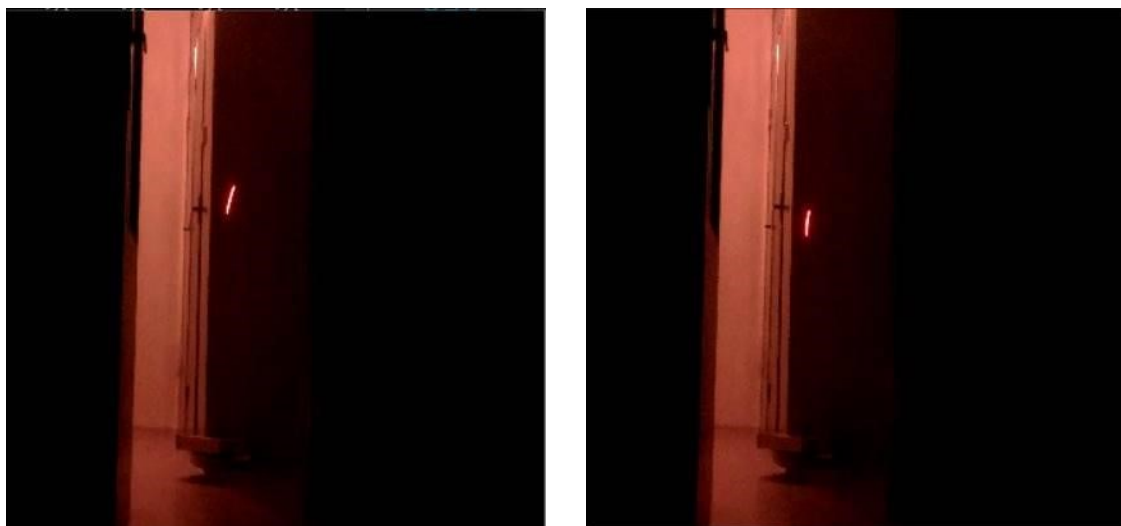
b)

Obrázek 17 Zpracování první sady snímků:

(a) Detekce světelné značky

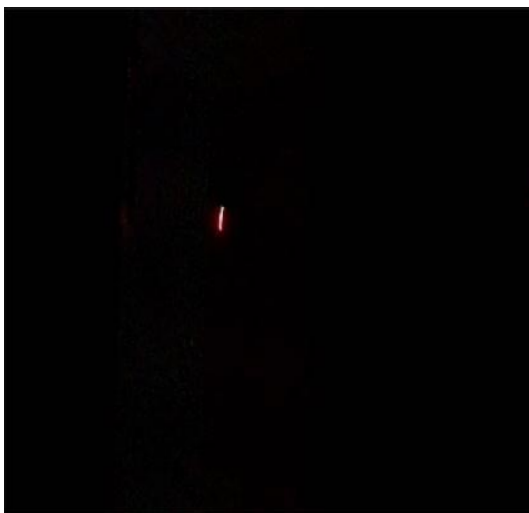
(b) Výsledek použití operace dilate

Druhá sada snímků obsahuje jasné objekty, jejichž barva je podobná barvě laserového bodu. V této situaci algoritmy hledání v prostoru RGB a HSV nedokázaly detekovat na snímku jenom laserový bod. Navržený algoritmus poskytuje jednoznačný výsledek odpovídající hledanému objektu.

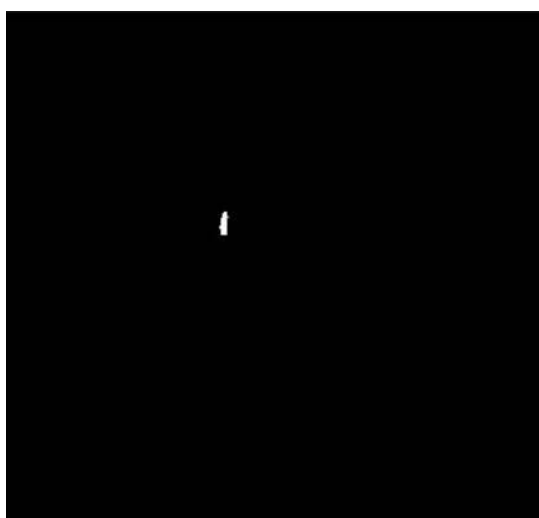


Obrázek 18 Zpracování druhé sady snímků:

Zachycení dvou následujících snímků z testovacího videa



Obrázek 19 Zpracování druhé sady snímků:
Výsledek odčítání snímků



a)



b)

Obrázek 20 Zpracování druhé sady snímků:

(a) Detekce světelné značky

(b) Výsledek použití operace dilate

Negativní vliv na úspěšnost detekce světelné značky má osvětlení s proměnnou intenzitou, které je příčinou výskytu šumů, a také osvětlení příliš intenzivní, protože pak světelná značka na snímcích není dost jasná a není proto detekována. Dalším faktorem, na kterém záleží úspěšnost, je povrch, na kterém se nachází světelná značka. Lesklý povrch hodně odráží světlo, což je příčinou vzniku šumů na snímcích. Úspěšnost detekce laserového bodu zaleží také na barvě pozadí. Nejmenší úspěšnost je v případě, že se bod nachází na jasném a lesklém červeném pozadí, naopak nejlepší úspěšnost je v případě, že pozadí je černé a matné.

5.2.2 Úspěšnost detekce světelné značky

Při detekci světelné značky na bílé stěně při denním osvětlení je úspěšnost algoritmu 93 % (ze 142 zpracovaných snímků byla světelná značka úspěšně detekována na 132 snímcích).

```
nikki@nikki:~/Документы/проекты/opencv$ ./algoritm_test*
Total number of frames      : 142
successfully processed      : 132
unsuccessfully processed    : 10
success ratio               : 92.9577%
nikki@nikki:~/Документы/проекты/opencv$
```

Obrázek 21 Úspěšnost detekce světelné značky na bílém pozadí

Úspěšnost detekce světelné značky na černém pozadí je 98–100 %. Příklady zpracování testovacích videí jsou v příloze k dané práci.

5.2.3 Rychlost zpracování snímků

Použití navrženého algoritmu umožňuje zpracování videa běžným notebookem v reálném čase. Výpočetní výkonnost minipočítače Raspberry Pi 2B umožňuje zpracování 142 snímků za 45,553 s (3,12 snímků/s). Pozn.: real je reálný čas mezi spuštěním a ukončením skriptu.

```
pi@raspberrypi: ~/Projects
pi@raspberrypi:~/Projects $ time ./algoritm_raspberry
Total number of frames      : 142
real    0m45.553s
user    1m48.120s
sys     0m0.420s
pi@raspberrypi:~/Projects $
```

Obrázek 22 Měření rychlosti zpracování snímků minipočítačem Raspberry Pi 2B

5.2.4 Srovnání algoritmů použitých v semestrální a bakalářské práci

V rámci semestrální práce se používal algoritmus detekce světelné značky v prostoru HSV (bez kombinace s metodou odčítání dvou následujících snímků a bez uskutečnění morfologických operací), který umožňoval minipočítači Raspberry Pi 2B zpracovávat 10 snímků za sekundu s úspěšností detekce 70–80 %.

V rámci bakalářské práce byl původní algoritmus detekce světelné značky v prostoru HSV zkombinován s metodou odčítání snímků a s použitím morfologických operací. Výsledkem je pokles rychlosti zpracování na 3,12 snímků za sekundu, zvětšení úspěšnosti detekce o 20 % a podstatné zvýšení odolnosti algoritmu proti rušením.

5.3 Možnosti zlepšení algoritmu detekce světelné značky

Zlepšením navrženého algoritmu pro detekci laserového bodu může být použití Double Difference metody pro předzpracování snímků, která pracuje se třemi po sobě jdoucími snímky videa. První snímek se odečítá od druhého, druhý snímek se odečítá od třetího, pak se provede logické násobení mezi výsledky prvního a druhého odčítání. Výsledkem je stanovení oblasti pohybů na snímcích s tím, že množství šumu je podstatně menší než při použití metody odčítání jenom dvou po sobě následujících snímků.

6 ZÁVĚR

Práce se zabývá problematikou automatizace činnosti zbraňové věže.

Na základě podmínek, ve kterých by měla zbraňová věž fungovat, byly provedeny teoretické výpočty, jejichž výsledky byly použity k návrhu algoritmu namíření zbraně v závislosti na pozici světelné značky.

V rámci bakalářské práce byl uveden přehled algoritmů počítačového vidění pro detekci objektů na posloupnosti snímků. Některé z uvedených algoritmů byly zvoleny pro testování a ověřování jejich funkcionality. Byly posouzeny výhody a nevýhody otestovaných algoritmů. Na základě provedené analýzy byl navržen algoritmus detekce světelné značky. Navržený algoritmus je kombinací metody odčítání dvou následujících snímků videa a metody detekce objektu na základě výběru pixelů s určitou hodnotou parametrů v prostoru HSV.

Testování algoritmů probíhalo v prostředí Microsoft Visual Studio 2015 (OS Windows 10) a Microsoft Visual Studio Code (OS Linux Ubuntu 16.04) s použitím C++ rozhraní knihovny OpenCV 3.1.

Navržený algoritmus byl přenesen do minipočítače Raspberry Pi 2B, jehož výpočetní výkonnost umožnila zpracování 3 snímků za vteřinu s úspěšností detekce světelné značky 90 až 100 %.

Testování navrženého algoritmu ukázalo možnost jeho využití pro ovládání činnosti prototypu automatické zbraňové věže.

LITERATURA

- [1] Wikipedia. *Автоматическая турель* [online]. 11/2016 [citováno 2016-11-7]. Dostupné z: https://ru.wikipedia.org/wiki/Автоматическая_турель
- [2] Portál *opencv.org*. [online]. 2016 [cit. 2016-11-11]. Dostupné z: www.opencv.org.
- [3] ЗОЛОТЫХ, Н., КУСТИКОВА, В., МЕЕРОВ, И. Обзор методов поиска и сопровождения транспортных средств на потоке видеоданных. 2012. УДК 004.932
- [4] Tuytelaars T., Mikolajczyk K. Local Invariant Feature Detectors: A Survey // *Foundation and Trends in Computer Vision*. 2007. V. 3. №3. s. 177–280.
- [5] Bay H., Ess A., Tuytelaars T., Gool L.V. SURF: speed up robust features // *Computer Vision and Image Understanding (CVIU)*. 2008. V. 110, № 3. s. 346–359.
- [6] Mikolajczyk K., Schmid C. Scale and affine invariant interest point detectors // *International Journal of Computer Vision*. 2004. №60(1). s. 63–86.
- [7] Tola E., Lepetit V., Fua P. A Fast Local Descriptor for Dense Matching // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. 2008. s. 1–8.
- [8] Calonder M., Lepetit V., Strecha C., Fua P. BRIEF: Binary Robust Independent Elementary Features // *Proceedings of the 11th European Conference on Computer Vision (ECCV'10)*. 2010.
- [9] Rublee E., Rabaud V., Konolige K., Bradski G. ORB: an efficient alternative to SIFT or SURF // *Proceedings of the International Conference on Computer Vision (CVPR'11)*. 2011. s. 2564–2571.
- [10] Hilario C., Collado J.M., Armingol J.M., Escalera A. Pyramidal Image Analysis for Vehicle Detection // *Proceedings to Intelligent Vehicles Symposium*. 2005. s. 88–93.
- [11] Amit Y. *2D Object Detection and Recognition: models, algorithms and networks*. The MIT Press, 2002. 325 s.
- [12] Фаворская М. Н., Пахирка А. И., Шилов А. С., Дамов М. В. Методы поиска движения в видеопоследовательностях // *Вестник сибирского государственного университета имени академика М. Ф. Решетнева*, 2009, стр. 71
- [13] Методы поиска движения в видеопоследовательностях [online] 2/2017 [citováno 2017-2-19] Dostupné z : sntbul.bmstu.ru/file/out/568767
- [14] Bradski G., Kaehler A. *Learning OpenCV Computer Vision with OpenCV Library* // O' Reilly Media Publishers, 2008. 571s.
- [15] Алпатов Б.А., Муравьев С.И., Муравьев В.С. Пространственный алгоритм выделения объектов на основе адаптивной пороговой обработки // *Цифровая обработка сигналов и ее применения. Мат. 8-й международный конф. Том 2, 2006, стр. 445*