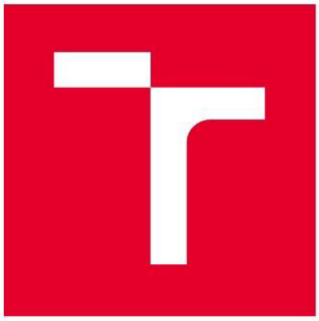


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

DETEKCE OSOB V OBRAZE POMOCÍ RASPBERRY PI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Lukáš Svoboda

AUTHOR

VEDOUCÍ PRÁCE — Ing. Michal Matějásko

SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Lukáš Svoboda
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	Ing. Michal Matějásko
Akademický rok:	2017/18

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Detekce osob v obraze pomocí Raspberry Pi

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat možnostmi realizace detekce osob v obraze s pomocí levného počítače Raspberry Pi. Možnosti budou prezentovány vytvořením úlohy v prostorách FSI VUT. K statistickému zpracování a zobrazení dat bude využita platforma ThingSpeak.

Cíle bakalářské práce:

1. Seznamte se s počítačem Raspberry Pi 3 a s platformou ThingSpeak.
2. Propojte Raspberry Pi 3 s kamerou a vytvořte software zpracovávající obraz z kamery.
3. Realizujte funkcionalitu detekce a počítání osob procházejících danou oblastí.
4. Propojte zařízení s ThingSpeak, kde zobrazujte a případně analyzujte získaná data.

Seznam doporučené literatury:

VALÁŠEK, M.: Mechatronika, Vydavatelství ČVUT, 1995

HEROUT, Pavel. Učebnice jazyka C. 5. vyd. České Budějovice: Kopp nakladatelství, 2008. ISBN 978-80-7232-351-7.

SUMMERFIELD, Mark. Python 3: výukový kurz. Brno: Computer Press, 2010. ISBN 978-80-251-2-37-7.

FEYNMAN, Richard Phillips, Robert B. LEIGHTON a Matthew SANDS. The Feynman lectures on physics. 6th print. Reading, Massachusetts: Addison-Wesley Publishing Company, 1977. ISBN 0-20--02116-1. http://www.feynmanlectures.caltech.edu/, California Institute of Technology

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato práce se zabývá detekcí osob v obraze pomocí kamerových zařízení s důrazem na využití minipočítače Raspberry Pi. Byly vytvořeny algoritmy zpracovávající obrazová data a schopné rozlišit, zda se jedná o příchozí nebo odchozí osobu. Jejich další úlohou je odesílání dat do platformy ThingSpeak, kde jsou vyobrazeny ve formě grafů a statusu na Twitteru. Detekce osob je realizována detektory pracujícími s metodami optického toku a ACF - Aggregate Channel features. Všechen software byl vytvořen ve vývojovém prostředí MATLAB/Simulink.

ABSTRACT

This thesis describes the detection of people in the image using camera devices with focus on using a Raspberry Pi minicomputer. There were created algorithms processing image data and containing algorithms, which are able to distiguish whether it is incoming or outcoming person. Another role of these softwares is to send data to the ThingSpeak platform, where charts are made from the data. The detectors of people are based on Optical flow and ACF - Aggregate Channel features methods. All software was created in the development environment MATLAB/Simulink.

KLÍČOVÁ SLOVA

Raspberry Pi, programování v MATLABu a Simulinku, detekce osob, metoda optického toku, metoda ACF - Agregatte channel feature, algoritmus počítání osob, platforma ThingSpeak

KEYWORDS

Raspberry Pi, programming in MATLAB and Simulink, person detection, Optical flow method, ACF - Aggregate Channel feature method, people counting algorithm, ThingSpeak platform

BIBLIOGRAFICKÁ CITACE: SVOBODA, L. *Detekce osob v obraze pomocí Raspberry Pi*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 45 s. Vedoucí bakalářské práce Ing. Michal Matějásko.

PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Michalu Matějáskovi za odborné vedení, ochotu a cenné rady, které mi pomohly při tvorbě této práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením pana Ing. Michala Matějáska a s použitím literatury a zdrojů uvedených v seznamu zdrojů.

V Brně dne 24. 5. 2018

.....

Lukáš Svoboda

OBSAH

1 ÚVOD 9
2 POUŽITÉ HARDWAROVÉ A SOFTWAROVÉ PLATFORMY, METODY DETEKCE 10
2.1 Raspberry Pi 3 10
2.2 Software Raspberry Pi 10
2.2.1 Operační systém Raspbian 11
2.3 PlatformaThingSpeak 11
2.4 Internet věcí 11
2.5 Použitá zobrazovací technika 11
2.5.1 Kamerový modul V2 12
2.5.2 Kamera AirLive POE-260CAM 12
2.6 MATLAB/Simulink 12
2.7 Metody detekce osob 13
3 SPOJENÍ A NASTAVENÍ HARDWARU A SOFTWARU 14
3.1 Vytvoření spojení Raspberry Pi a MATLABu/Simulinku 14
3.2 Spojení Raspberry Pi s hostitelským počítačem 15
3.2.1 Přes kabel RJ45 přímo s hostitelským počítačem 15
3.2.2 Pomocí sítě LAN 17
3.3 Vytvoření komunikace pomocí MATLAB 18
3.4 Nastavení modelu Simulinku pro Raspberry Pi 18
4 ALGORITMUS SOFTWARU V MODELU SIMULINKU 19
4.1 Pořízení obrazu 19
4.2 Detekce osoby 20
4.2.1 Úprava obrazu 20
4.2.2 Detekce pohybujících se objektů 21
4.2.3 Filtrace a vytvoření binárního obrazu 22
4.2.4 Označení objektu 23
4.3 Zjištění odchozích nebo příchozích osob 25
4.4 Aktualizace počtu osob 29
4.5 Odeslání dat do platformy ThingSpeak 30
5 ALGORITMUS SOFTWARU V MATLABU 31
5.1 Metoda ACF - Aggregate Channel features 31
5.2 Zjištění odchozích nebo příchozích osob 32
5.3 Odesílání dat do platformy ThingSpeak 34
6 VIZUALIZACE DAT V PLATFORMĚ THINGSPEAK 35
7 ZÁVĚR 35
8 POUŽITÉ ZDROJE A LITERATURA 37
9 SEZNAM OBRÁZKŮ 40
10 SEZNAM ZKRATEK 41
11 SEZNAM PŘÍLOH 42

1 Úvod

V dnešní době se stále častěji setkáváme se situacemi, kdy je nezbytně nutné detekovat pohyb jedné či více osob. Nejčastěji takové detektory odhalují narušení prostoru pachatelem či zjišťují a vyhodnocují pohyb ve sledovaném prostoru, například na letištích, nádražních halách, parkovištích, veřejných prostorách měst či v komerčních budovách.

Neustále jsou vyvíjeny nové metody pro detekci objektů v obraze. Vyzkoušet si různé metody a rozšířit tak zkušenosti s programováním umožňuje miniaturní počítač Raspberry Pi a to díky neomezeným možnostem propojení a ovládání periferních zařízení.

Počítač Raspberry Pi vznikl původně proto, aby se jak děti, tak i dospělí, mohli zábavným způsobem učit programování počítačů. Raspberry Pi se stal, navzdory své nízké ceně a velikosti platební karty, vysoce funkčním počítačem, jenž se hodí k nejrůznějším úkolům: od přehrávání videí, přes zábavu s hardwarem, až po výuku programování [1].

Tato práce se zabývá možnostmi detekce osob v obraze za využití vývojového prostředí MATLAB/Simulink a miniaturního počítače Raspberry Pi, který je propojen s kamerou ke sledování osob nacházejících se v Mechatronické laboratoři FSI VÚT (dále Mechlab). Údaj o počtu osob je odeslán do platformy ThingSpeak a následně vizualizován v podobě grafu a statusu na Twitteru.

Postupně je v této práci představen minipočítač Raspberry Pi model 3, softwarové a hardwarové vybavení, spojení Raspberry Pi se systémy MATLAB a Simulink, prezentován algoritmus počítání osob a odesílání dat do platformy ThingSpeak, vykreslen graf vývoje počtu osob a popsána ukázka na Twitteru. Poslední částí bakalářské práce je závěr, ve kterém jsou zejména shrnuty praktické poznatky o fungování vytvořeného softwaru.

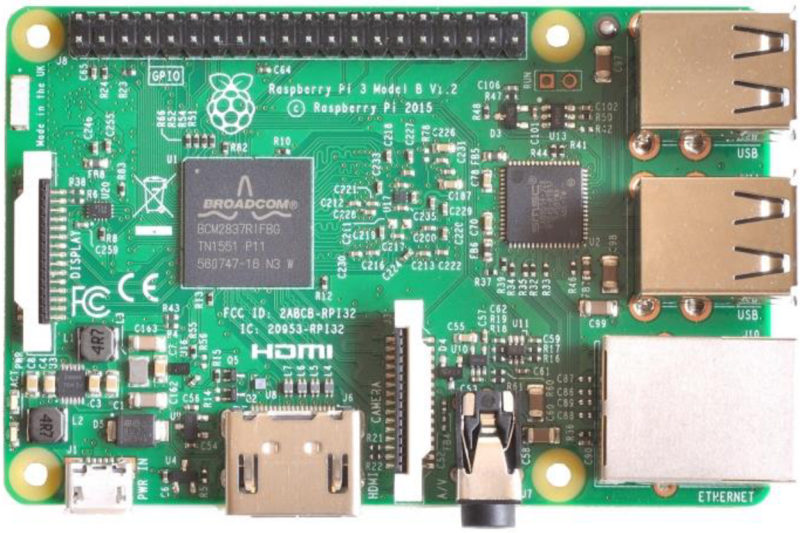
2 Použité hardwarové a softwarové platformy, metody detekce

Tato kapitola se zabývá minipočítačem Rasberry Pi, především je zde popsán model Raspberry Pi 3 a další použité hardwarové a softwarové platformy.

2.1 Raspberry Pi 3

Jedná se o třetí verzi mikropočítače s cenou 35 dolarů. Tento model je vybaven integrovanou Wi-Fi 802,11n s Bluetooth 4.1. Základem se stal 64bitový procesor se čtyřmi jádry *ARM Cortex-A53*, pracující na frekvenci 1,2 GHz, který je přibližně 10x výkonnější v porovnání s čipem v první verzi, přesto je naprosto kompatibilní se staršími modely *Raspeberry Pi 1* a *2*. Operační paměť zůstala nezměněna (1 GB). O grafické operace se stará výkonná grafická karta *Broadcom VideoCore IV 3D* (ve 3D je takt zvýšen o 50 MHz na 300 MHz). Pro instalaci operačního systému a ukládání dat slouží paměťové karty typu microSD, pro které je připraven paměťový slot (počítač je bez předinstalovaného operačního systému). Mezi konektorovou výbavu patří porty HDMI, audio výstup, GPIO pro připojení zabezpečovací techniky, čtyři klasické USB, Camera Serial Interface (CSI), Display Serial Interface (DSI). Připojení počítače k domácí nebo firemní síti je možno i prostřednictvím klasického LAN rozhraní [2],[3].

Model Raspberry Pi 3 Model B je hlavním hardwarem této bakalářské práce.



Obr. č. 1: Raspberry Pi 3 Model B

2.2 Software Raspberry Pi

Operační systém se nahraje na paměťovou microSDkartu a z té se následně přímo spouští. Jako operační systém je možno použít některou z linuxových distribucí,

upravených pro ARM procesory – například *Debian Squeeze, Arch Linux ARM, Fedora Remix nebo Raspbian* [4].

Pro samotné programování v bakalářské práci je použit operační systém *Raspbian*, který patří k nejpoužívanějším [5].

2.2.1 Operační systém Raspbian

Operační systém *Raspbian* je odvozený z *Debianu* pro Raspberry Pi i osobní počítače. Je oficiálně poskytován nadací Raspberry Pi Foundation, která jej doporučuje jako primární operační systém pro počítače z "rodiny" Raspberry Pi, neboť je optimalizovaný pro ARM procesory. Jako hlavní desktopové prostředí *Raspbian* používá PIXEL, který se skládá z modifikovaného prostředí LXDE a správce oken Openbox [1].

2.3 PlatformaThingSpeak

ThingSpeak je online analytická služba. Jedná se o projekt typu open source (volně přístupná) se zaměřením na tvorbu aplikací *Internetu věcí*. ThingSpeak umožňuje uživatelům shromažďovat v prostředí cloudu aktuální data ze senzorů v reálném čase, analyzovat data, vizualizovat data ve formě grafů a následně vytvořit na tato data odpověď. Touto odpovědí může být například upozornění pomocí webových služeb jako je Twitter. Ukládání a načítání dat je realizováno pomocí HTTP protokolu přes internet nebo pomocí lokální sítě. Výhodou této platformy je možnost využívat analytické funkce programu MATLAB. Uživatelé zde mohou psát a spouštět skripty v jazyce MATLAB a pomocí nich pracovat s daty [7].

Další výhodou je možnost navrhnout a prototypovat systémy *Internetu věcí,* aniž by si musel uživatel zřídit vlastní serverovou architekturu nebo programovat webový software.

2.4 Internet věcí

V současnosti je *Internet věcí* poměrně často skloňovaným pojmem v souvislosti s informačními a komunikačními technologiemi. *Internet věcí* je možno chápat jako síť propojených zařízení, která je založena na standardizovaných komunikačních protokolech umožňující výměnu a sdílení dat a informací. Cílem sítě je tedy umožnit spojení mezi zařízeními. Síť v případě *Internetu věcí* nemusí představovat jen celosvětový systém vzájemně propojených počítačů komunikujících na základě protokolů TCP/IP, ale také pouze lokální síť LAN. Zařízením může být elektronika obsahující software a senzory, které umožňují získávat informace o snímané veličině popřípadě veličinách. Takže se jedná o zařízení samostatně poskytující data, jenž jsou odesílána pomocí kabelu nebo bezdrátově sdílena s dalšími zařízeními [8].

2.5 Použitá zobrazovací technika

Pro tuto bakalářskou práci je zvolena průmyslová kamera Kamera AirLive POE-260CAM, protože je naistalována v Mechlabu a oficiální kamera pro Raspberry Pi -

Kamerový Modul V2, který je využíván ke sledování na vzdáleném počítači zachycených videí ve vysoké kvalitě (HD) a zachycených fotografií. Průmyslová kamera je ideální pro monitorování situací a následné nahrávání záznamů.

2.5.1 Kamerový Modul V2

Raspberry Pi Kamerový Modul V2 má integrovaný Sony IMX219 8-megapixel senzor. Podporuje video režimy 1080p30, 720p60 a VGA90 stejně jako zachycování obrázků a připojuje se přes 15cm plochý kabel do portu CSI na Raspberry Pi. Kamera funguje se všemi modely Raspberry Pi ve verzích 1, 2, a 3 a je podporována v nejnovější verzi preferovaného systému Raspbian, Raspberry Pi. Kamerový modul je možné využít v aplikacích pro domácí zabezpečení a pro fotopasti [9].

2.5.2 Kamera AirLive POE-260CAM

Jedná se o výkonnou pohyblivou kameru s otáčením v rozsahu 355° horizontálně a 125° vertikálně s maximálním rozlišením 640 x 480 pixelů. Kamera umožňuje záběry streamovat pomoví protokolu RSTP (Real Time Streaming Protocola) nebo ukládat do počítače, ke kterému je připojena jen ethernetovým kabelem, po němž jsou doručována jak data, tak i elektrická energie a to díky podpoře PoE standardu. Může být tedy umístěna i tam, kde není elektrický rozvod [10].

2.6 MATLAB/Simulink

Ke spuštění algoritmu počítání osob a následnému vygenerování výstupů z dat je třeba naprogramovat výpočetní a vývojové prostředí MATLAB a Simulink firmy MathWorks. MATLAB a Simulink nabízí přímé napojení na otevřenou datovou platformu *ThingSpeak* (platformu pro internet věcí).

MATLAB je výkonný programovací jazyk a interaktivní prostředí pro vědecké a technické výpočty, analýzu dat, vizualizaci a vývoj algoritmů. Data lze do MATLABu importovat ze souborů, jiných aplikací nebo externích zařízení a následně je analyzovat pomocí vestavěných inženýrských a matematických funkcí, grafů a vizualizací. Jazyk MATLABu podporuje vektorové a maticové operace, které tvoří základ řešení technických a vědeckých problémů. Příkazy mohou být zadány jeden po druhém a okamžitě vrací výsledky, což umožňuje zkoumat více možností a dojít tak k optimálnímu řešení. V MATLABu lze pomocí skriptů a funkcí vytvářet komplexní programy či aplikace.

Simulink je nadstavba MATLABu, která poskytuje možnost uživateli tvořit modely dynamických soustav ve formě blokových schémat, jež jsou přehledným grafickým zobrazením výpočtů, analýz dat a algoritmů. Modely mohou být vyobrazeny rovnicemi nebo mohou být sestavené z bloků, reprezentujících prvky reálných systémů. Je možné sestavovat nejen modely fyzikálních soustav, ale také algoritmy řídicích systémů včetně jejich automatického ladění, systémy pro zpracování signálu, komunikace a zpracování obrazu [11], [12] a [13].

2.7 Metody detekce osob

Automatická detekce a sledování jedné nebo více osob ve videích je jedním z hlavních výzkumných zájmů v aplikacích založených na počítačovém vidění. Existuje mnoho metod, které se od sebe liší v principu různých algoritmů, popisu objektů a hlavně svými výsledky. V současné době nejlépe pracují detektory osob v obraze založené na metodách *HOG – Histogram of Oriented Gradients, HAAR Cascade Classifie* a *ACF – Aggregate Channel features.*

Metoda *Histogram of Oriented Gradients*: základ metody tvoří histogramy orientovaných gradientů; objekt je detekován pomocí intenzity gradientů a hran objektu, přestože není známá přesná poloha objektu.

Metoda *HAAR Cascade Classifie:* podstatou metody jsou tzv. *Haarovy příznaky*, které nabývají hodnoty v závislosti na pixelech světlé a tmavé části; světlé a tmavé části vznikají podle kontrastních odchylek mezi sousedními obdélníkovými skupinami pixelů.

Metoda *ACF – Aggregate Channel features:* metoda je založená na podstatných informacích o obrazu, tzv. příznacích; metoda je podrobně popsána v kapitole 5.1, neboť záměrem bylo v této bakalářské práci použít již vytvořený detektor společností MathWorks, pracující na metodě ACF [16] a [17].

3 Spojení a nastavení hardwaru a softwaru

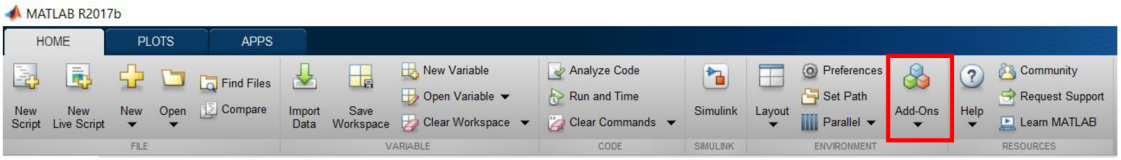
Počítač Raspberry Pi, vybavený webkamerou, je pomocí LAN sítě připojený k internetu. Algoritmus detekce a počítání procházejících osob byl vytvořen v Simulinku, včetně některých funkcí psaných v MATLABu. Odsud byl následně vygenerován kód, který byl nahrán do Raspberry Pi, kde funguje jako samostatná aplikace. Funkcí algoritmu je detekce osoby v obraze, rozhodnutí, zdali je osoba příchozí nebo odchozí, a odeslání informace na portál ThingSpeak, kde dochází k vizualizaci výsledků.

3.1 Vytvoření spojení Raspberry Pi a MATLABu/Simulinku

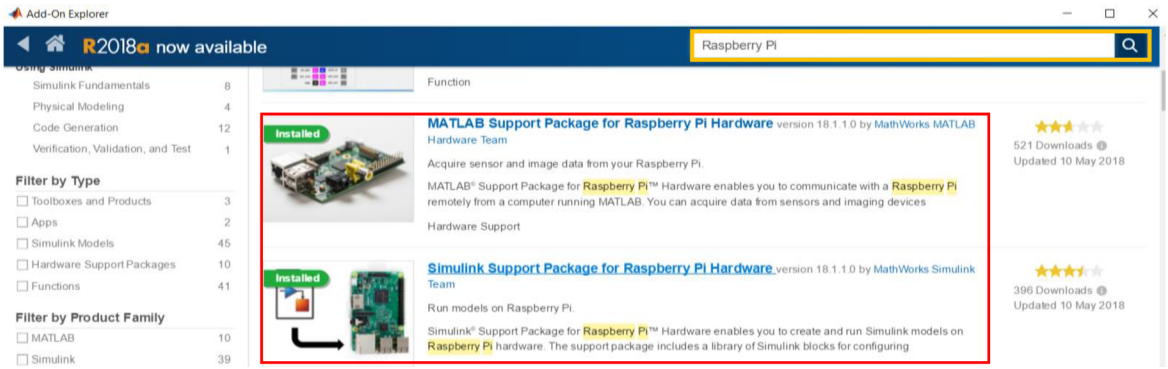
Pro spojení Raspberry Pi a MATLABu je vyžadováno následující příslušenství:

- Počítač Raspberry Pi
- SD karta (4GB nebo větší)
- 5V micro USB síťový adaptér
- Ethernetový kabel, přístup k bezdrátovému připojení pomocí síťové adresy (IP)

Spojení Raspberry Pi s MATLABem/Simulinkem je realizováno pomocí podpůrných balíčků *MATLAB Support Package for Raspberry Pi Hardware* a *Simulink Support Package for Raspberry Pi Hardware*, vytvořených firmou MathWorks. Podmínkou je mít nainstalovaný program MATLAB verze R2014a a vyšší. Podpůrné balíčky *MATLAB Support Package for Raspberry Pi Hardware* a *Simulink Support Package for Raspberry Pi Hardware* umožňují komunikovat pomocí vytvořených funkcí s hardwarem a získávat data z periferních zařízení (například z kamery). Instalaci poskytuje defaultně adresář předpokládaného umístění MATLABu. Na kartě *HOME* se odklikne ikona *Add-Ons* a ve vyhledávači nadepíše *Raspberry Pi,* následně se zobrazí nabídka produktů a po kliknutí na již zmiňované balíčky se zobrazí okno s ikonou *Install*. Po kliknutí na tuto ikonu se spustí instalace souborů k jednotlivým balíčkům. Po instalaci se ikona *Install* změní na ikonu *Manage* (obr. č. 4) [14] a [15].



Obr. č. 2: Karta HOME s označenou ikonou Add-Ons



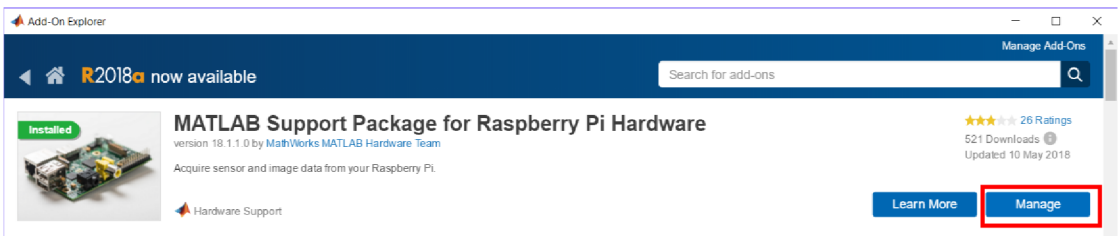
Obr. č. 3: Okno s podpůrnými balíčky a vyhledávačem hardwaru

3.2 Spojení Raspberry Pi s hostitelským počítačem

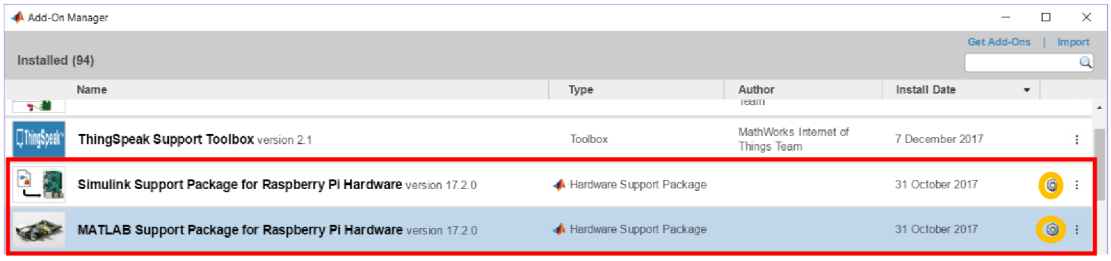
3.2.1 Přes kabel RJ45 přímo s hostitelským počítačem

Spojení Raspberry Pi s hostitelským počítačem lze provést až po instalaci podpůrných balíčků *MATLAB Support Package for Raspberry Pi Hardware* a *Simulink Support Package for Raspberry Pi Hardware*. Spojení je realizovatelné v podpůrném balíčku při kliknutí na ikonu *Manage* (obr. č. 4). Dále se zobrazí seznam podpůrných balíčků s tzv. volbou *Setup* (obr. č. 5). Po kliknutím v tomto místě pravým tlačítkem myši se objeví nabídka podporovaných hardwarů, kde stačí označit Raspberry Pi Model B (obr. č. 6). Dalším krokem po kliknutí na ikonu *Next* je výběr operačního systému z nabídky. Možnostmi výběru je *MathWorks Raspbian Image* nebo přizpůsobení se existujícímu, již nainstalovanému, operačního systému na hardwaru. Jak již bylo zmíněno v kapitole 2.2, pro účely této bakalářské práce byl použit operační systém Raspbian (obr. č. 7). Následující postup spočívá ve zvolení způsobu síťového připojení. Zde byla zvolena možnost přímého spojení s hostitelským počítačem přes Ethernet kabel (obr. č. 8). V dalším kroku instalátor požaduje vložení paměťové karty microSD, kterou rozpozná a určí, na které jednotce hostitelského počítače se karta nachází (obr. č. 9). V dalším kroku je požadováno potvrzení a pomocí ikony *Write* dojde k zapsání firmwaru na SD kartu (obr. č. 10). Současně program MATLAB varuje o kompletním smazání obsahu na kartě, proto je nutné se přesvědčit, že se nejedná o jiné připojené zařízení např. přes USB kabel.

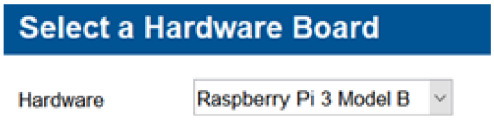
Jakmile je operační systém kompletně nainstalovaný na kartě, další instrukce spočívají v testování spojení: vložit microSD kartu do slotu na spodní straně Raspberry Pi, spojit hostitelský počítač s Raspberry Pi ethernetovým kabelem a spustit Raspberry Pi zapojením síťového adaptéru (obr. č. 11). Následujícím krokem je pokus o načtení tím, že se objeví předposlední okno s výpisem IP adresy a tlačítkem pro otestování spojení. Úspěšné spojení se projeví zeleným kolečkem s bílou " fajfkou" (obr. č. 12) [14] a [15].



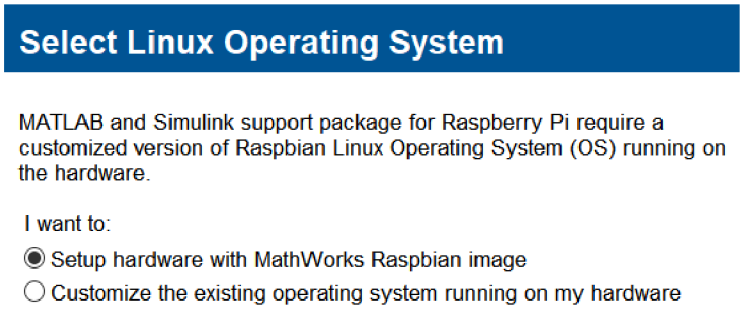
Obr. č. 4: Okno podpůrného balíčku s ikonami Learn More a Manage



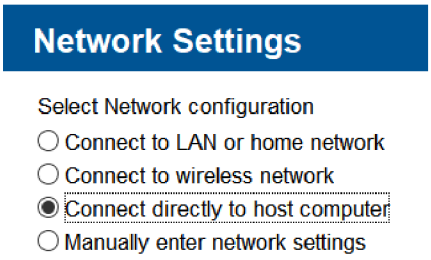
Obr. č. 5: Okno s podpůrnými balíčky a s ikonami Setup



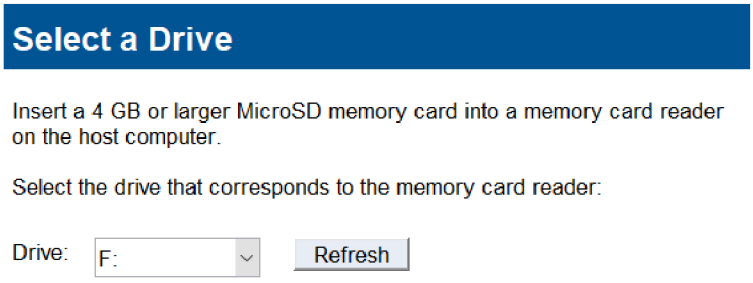
Obr. č. 6: Výběr hardwaru



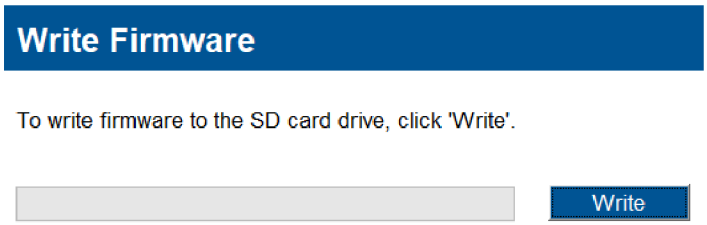
Obr. č. 7: Výběr operačního systému



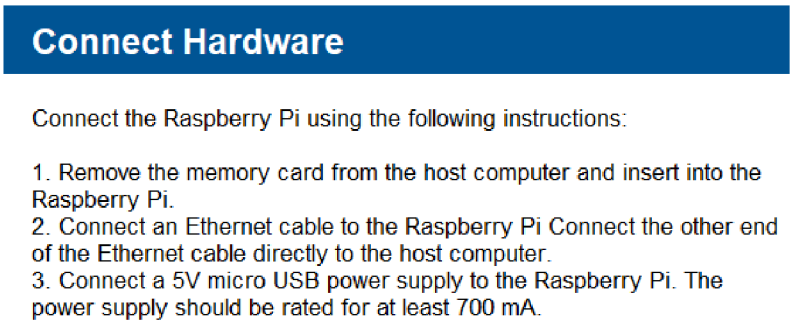
Obr. č. 8: Výběr síťového spojení



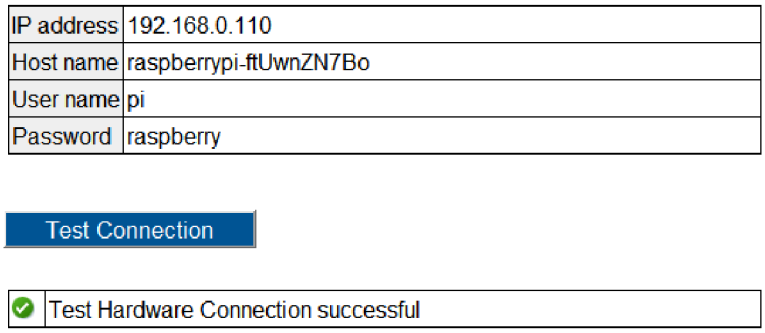
Obr. č. 9: Výběr jednotky



Obr. č. 10: Zapsání firmwaru na SD kartu



Obr. č. 11: Postup zapojení Raspberry Pi



Obr. č. 12: Okno úspěšného spojení Raspberry Pi s Matlabem

3.2.2 Pomocí sítě LAN

Toto spojení je realizováno podobně jako přímé spojení Raspberry Pi s hostitelským počítačem pomocí ethernetového kabelu. Rozdíl je v kroku, ve kterém se konfiguruje síť. Zde je nutné vybrat možnost spojení pomocí *LAN sítě* nebo *domácí sítě* (obr. č.8).

Je třeba si dávat pozor ve chvíli, kdy dochází ve vyhledávání IP adresy Raspberry Pi a na síť je připojeno více Raspberry Pi, neboť může dojít k připojení hostitelského počítače k nechtěnému Raspberry Pi. Tomuto nežádoucímu připojení se může předejít tím, že se zadá IP adresa Raspberry Pi ručně a to ve fázi, kdy se konfiguruje síť. Pokud je připojena pouze jedna adresa Raspberry Pi, může se nechat odsouhlasené automatické spojení. Při správném spojení se v posledním kroku v taktéž objeví zelené kolečko s bílou " fajfkou," stejně jako u přechozího připojení [14] a [15].

3.3 Vytvoření komunikace pomocí MATLAB

Komunikace MATLABu s Raspberry Pi se v matlabovském skriptu vytváří pomocí příkazu *raspi*. Pokud se pracuje s více platformami Raspberry Pi, je nutné do argumentů této funkce ještě napsat IP adresu, přihlašovací jméno a heslo.

3.4 Nastavení modelu Simulinku pro Raspberry Pi

Aby bylo možné použít grafický nástroj Simulink ve spojení s minipočítačem Raspberry Pi, musí být model dobře nastaven, proto je třeba zadat správně na kartě *Hardware Implementation* první položku a to *Raspberry Pi.* Po tomto kroku dochází k inicializaci[1] a updatu[2] modelu. Bude-li hardware Raspberry Pi objeven správně, dole v okně se objeví údaje o Raspberry Pi, jako je IP adresa, přihlašovací jméno a heslo. Důležitým krokem ke správnému nastavení modelu Simulinku je v pravém formuláři ve sloupci *Action,* zatrhnout položku *Simulink a* nastavit režim *Simulation mode* (zobrazování klíčových vlastností) z *Normal* na *External.* (Simulink External Mode slouží k propojení schématu s aplikací, která ovládá model cílového externího hardwaru, blokové schéma se stane uživatelským prostředím pro real-timeovou aplikaci). Následně se odklikne ikona *Deploy to Hardware*, program se uloží na paměťovou kartu Raspberry Pi a dále bude automaticky spouštěn. Program bude probíhat tehdy, když bude minipočítač Raspberry Pi připojen ke zdroji [14] a [15].

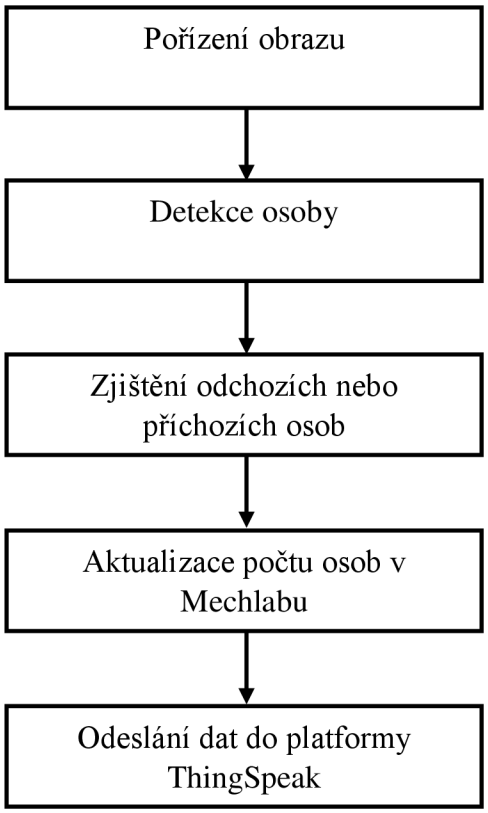
---

[1] Inicializace - zavedení určitého programu do nadřazeného systému nebo spuštění určité funkce, může se překrývat s pojmem instalace. Ovšem inicializace označuje širší spektrum softwarových operací. Inicializací může být spuštění prostředí, ve kterém běží určitý program nebo i běžné načtení souboru [26].

[2] Update = anglicky "aktualizovat" či "aktualizace", jedná se o činnost, při níž se staré nahrazuje něčím novějším (či modernějším) [26].

4 Algoritmus softwaru v modelu Simulinku

V této bakalářské práci byl aplikován algoritmus, jehož schéma je následující:



Obr. č. 13: Schéma algoritmu softwaru v modelu Simulinku

Podle tohoto schématu byl vytvořen model detekce a počítání osob v Simulinku - Příloha č. 1, Příloha č. 2, Příloha č. 3. Schéma zobrazuje jednotlivé části algoritmu a jejich vzájemné propojení. Jednotlivé části jsou demonstrovány v následujících podkapitolách.

4.1 Pořízení obrazu

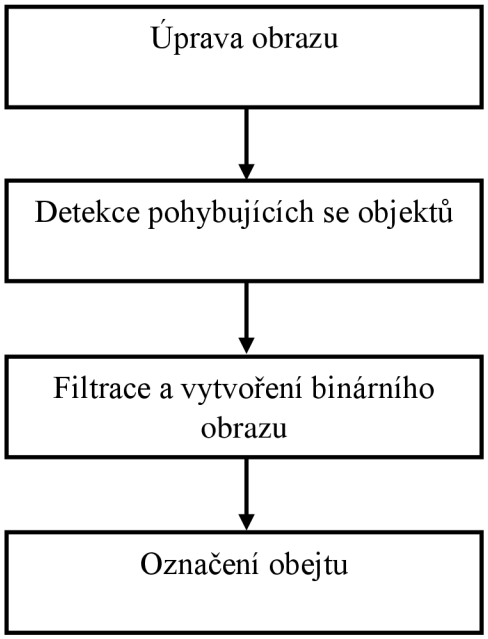
Obraz se dá formulovat jako vícerozměrný signál, jenž nám poskytuje stanovit u daného objektu jeho velikost, jas či polohu [17].

Obraz je v této práci pořizován kamerou s rozlišením 160x120 ve formátu pixelu RGB se snímkovací frekvencí 5 fps, jedná se tedy o dynamický obraz složený z posloupnosti statických obrazů (obr. č. 14).

Obr. č. 14: Blok, ve kterém dochází k pořízení obrazu

4.2 Detekce osoby

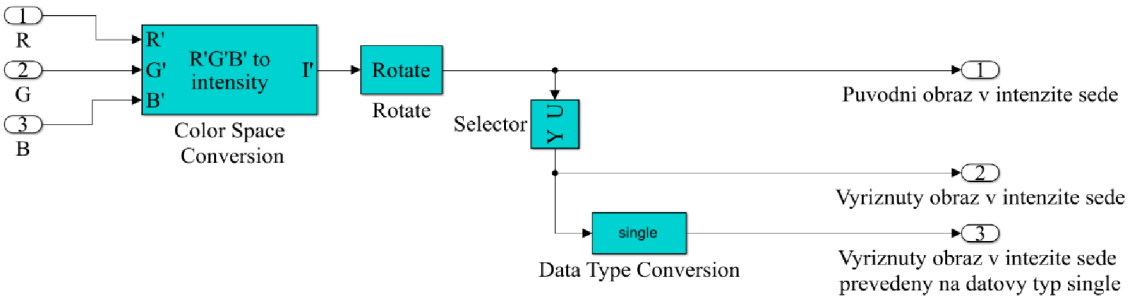
Pokud je třeba v obraze najít nějaký objekt, je nutné, aby obraz prošel určitým procesem, který lze rozdělit do několika částí [17]. Postup procesu zpracování obrazu představuje následující schéma algoritmu:



Obr. č. 15: Schéma algoritmu detekce objektu v Simulinku

4.2.1 Úprava obrazu

Následujícím krokem po pořízení obrazu je jeho úprava. Ta začíná převodem z barevného RGB obrazu do intenzity šedé. Protože obraz kamery je z fyzikálního hlediska skutečný a převrácený, je nutné obraz otočit o 180°. Dále dochází k vystřižení nového obrazu z původního. Vystřižení obrazu spolu s převodem obrazu do intenzity šedé jsou důležité z důvodu zmenšení výpočtové náročnosti (obr. č. 20) a (obr. č. 21). Tato část je ukončena převodem dat do datového typu *Single*, protože vtupem do bloku *Optical Flow* mohou být data, která jsou datového typu *Single* nebo *Double* (obr. č. 16).



Obr. č. 16: Subsystém, ve kterém dochází k úpravě obrazu

4.2.2 Detekce pohybujících se objektů

Optický tok je základem jedné z metod pro zpracování dynamických obrazů. Optický tok popisuje směry a rychlosti pohybu obrazových bodů v časové sekvenci dvou následujících snímků. Výstupem této metody je popis každého bodu obrazu v podobě dvourozměrného vektoru rychlosti, který zahrnuje informaci o směru a velikosti rychlosti pohybu v daném místě obrazu. Jelikož pro účely bakalářské práce byla vybrána kamera statická a objekty v obraze dynamicky mění polohu, může být obraz popsán dvourozměrnou dynamickou jasovou funkcí polohy a času $I(x, y, t)$[18], [19] a [20]. Je-li předpokladem to, že v blízkém okolí posunutého bodu nedojde ke změně intenzity jasu, je možné použít následující vztah [20]:

$$I(x, y, t) = I(x + \partial x, y + \partial y, t + \partial t) \tag{1}$$

Upraví-li se vztah (1) pomocí Taylorova rozvoje a zanedbání členů vyšších řádů, vznikne vztah ve tvaru (2):

$$I_x \cdot v_x + I_y \cdot v_y + I_t = 0 \tag{2}$$

kde v_x a v_y jsou složky vektoru rychlosti obrazového bodu (pixelu).

Tento vztah (2) se může vyjádřit formálně:

$$\nabla I \cdot \vec{v} + I_t = 0 \tag{3}$$

Význam jednotlivých členů vztahu (3) je následující: ∇I je prostorový gradient zohledňující intenzitu jasu, $\vec{v}$ je vektor rychlosti – optický tok pixelu ve směru gradientu a I_t značí časovou derivaci intenzity jasu mezi dvěma snímky.

Obecně platí, že u objektů pohybujících se v menších vzdálenostech od zařízení zaznamenávajících aktuální obraz, se projeví pohyb zřetelněji než u objektů nacházejících se ve větších vzdálenostech od zařízení [20].

K výpočtu optického toku byla zvolena Metoda *Horn-Schunck (H-S)* pomocí parciálních derivací obrazového signálu *I(x,y,t)*. Pro zmírnění příliš velkých změn složek optického toku *(v_x, v_y)* je ke gradientnímu omezení přidán tzv. globální člen vyhlazení. Tato metoda slouží k minimalizaci součtu čtverců celkové chyby ρ_{SH}, která může být vyjádřena následujícím vztahem [20]:

$$\rho_{SH} = \iint_D \left\{ (\nabla I \cdot \vec{v} + I_t) + \lambda^2 \left[\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2 \right] \right\} dx\, dy \tag{4}$$

Význam jednotlivých členů vztahu (4): D je oblast celého snímku, ∇I je prostorový gradient zohledňující intenzitu jasu, $\vec{v}$ je vektor rychlosti – optický tok pixelu ve směru gradientu, I_t značí časovou derivaci intenzity jasu mezi dvěma snímky, v_x a v_y jsou složky rychlosti, x a y jsou složky polohy obrazového bodu (pixelu) a koeficient λ vyjadřuje relativní vliv druhého přidaného členu.

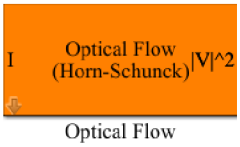
Při použití optického toku dochází k již zmíněným situacím, kdy jednotlivé složky (v_x, v_y) optického toku se mění příliš rychle. Proto se aplikuje tzv. propagace homogenního optického toku do míst s horšími podmínkami a naopak. To znamená, že v místech s menším gradientem se první člen změní málo, tudíž bude kladen větší důraz na druhý člen (homogenitu optického toku). V opačném případě bude kladen větší důraz na první člen vyjadřující gradientní omezení v místech s velkým gradientem [18], [19] a [20].

Ke zjištění minimální chyby se musí vztah (4) derivovat podle jednotlivých složek optického toku a takto vzniklé druhé parciální derivace položit rovno 0. Po úpravě je výsledkem soustava dvou rovnic o dvou neznámých (v_x, v_y), kterou je možné zapsat v maticové podobě [20]:

$$\begin{bmatrix} \lambda^2 + I_x^2 & I_x I_y \\ I_x I_y & \lambda^2 + I_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \lambda^2 \overline{v_x} & -I_x I_t \\ \lambda^2 \overline{v_y} & -I_y I_t \end{bmatrix} \tag{5}$$

Významy členů vztahu (5) λ, I_x , I_y v_x a v_y jsou vysvětleny v předchozích částech této kapitoly. Členy $\overline{v_y}$ a $\overline{v_x}$ vyjadřují vážené průměry jednotlivých složek vypočítané v okolí pixelu o souřadnicích *(x, y)*.

V Simulinku je optický tok vypočítáván blokem *Optical flow*. V bloku byla zvolena metoda výpočtu *Horn-Schunck*, jak již bylo zmíněno (obr. č. 17).



Obr. č. 17: Blok Optical flow

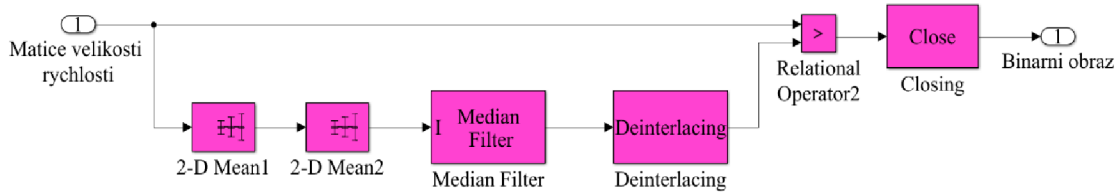
4.2.3 Filtrace a vytvoření binárního obrazu

Výstupem z bloku *Optical flow* je matice velikostí rychlosti. Jelikož v blokcích *2-D Mean1* a *2-D Mean2* je potvrzeno pole *running mean* (obr. č. 18), tak bloky sledují průměrnou hodnotu každého z kanálů v časové sekvenci vstupů. V tomto módu bloky berou každý element matice jako kanál. Dalším použitým blokem je *Median filter*, který vezme každý pixel a jeho okolí v obraze. Z těchto pixelů stanoví medián. Hodnota mediánu se stává novou hodnotou zpracovaného pixelu. Tímto způsobem bloky odstraňují nežádoucích šum v obraze.

Dalším blokem je *Deinterlacing*, jehož funkcí je zvýšit ostrost obrazu odstraněním chyb, které vznikly při prokládání videa. Obraz tvoří viditelné řádky, které technika prokládání videa rozdělí na liché a sudé řádky, a pak je střídavě s určitou frekvencí zobrazuje. Časová prodleva mezi obnovením lichých a sudých řádků vytváří jev rozmazání videa.

Posledními bloky této části algoritmu detekce osoby jsou *Retional Operator2* a *Closing*. Výstupem je **binární obraz**, který vznikl následně: nejprve došlo k porovnání jednotlivých elementů matice rychlosti (výstup optického toku) v logickém operátoru

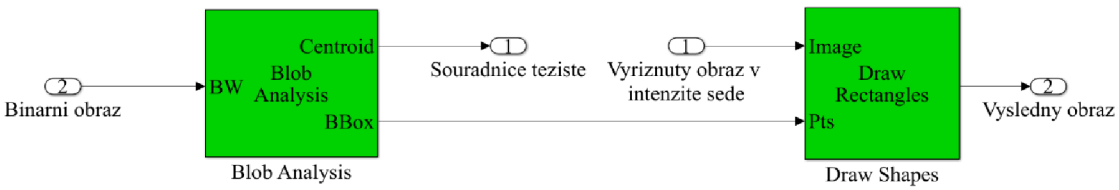
(blok *Retional Operator2*) s hodnotami, které byly upraveny v blocích 2-*D Mean1, 2-D Mean2 , Medián Filter a Deinterlacing*. Potom byl takto vzniklý **binární obraz**, složený z matice obsahují jedničky a nuly, zdokonalen blokem *Closing* (obr. č. 22). Tento blok provádí morfologickou dilataci a morfologickou erosi pro stejné okolí daného pixelu, čímž zdůrazňuje strukturu objektu a zdokonaluje binární obraz[3] (obr. č. 18) [18], [22].



Obr. č. 18: Subsystém, ve kterém dochází k filtraci a vytvoření binárního obrazu

4.2.4 Označení objektu

Označení objektu je realizováno pomocí bloků *Blob Analysis a Draw Shapes*. V bloku *Blob Analysis* je nutné specifikovat označení pouze jednoho objektu (složeného z bílých pixelů) a to v případě, přesáhne-li počet bílých pixelů 500. Toutu specifikací se zajišťuje, aby v obraze nebyly označeny i menší stíny, a že vždy bude v obraze označen pouze jeden objekt, neboť v této bakalářské práci byl učiněn předpoklad, že vyříznutá oblast je takové velikosti, aby se v ní nacházela pouze jedna osoba. Blok *Draw Shapes* objekt označí do černého obdélníku (obr. č. 19).



Obr. č. 19: Subsystém, ve které dochází k označení objektu

[3] Podstatou morfologické analýzy je strukturní element (množina bodů), označován jako maska, jež slouží ke zjištění vlastností zkoumaného objektu. Erozi binárního obrazu vhodnou maskou je možné definovat jako průnik množiny obrazu s množinou všech posuvů masky. Dilatace binárního obrazu vhodnou maskou je naopak sjednocení množiny obrazu s množinou všech posuvů masky [27].

Obr. č. 20: Původní obraz v intenzitě šedé

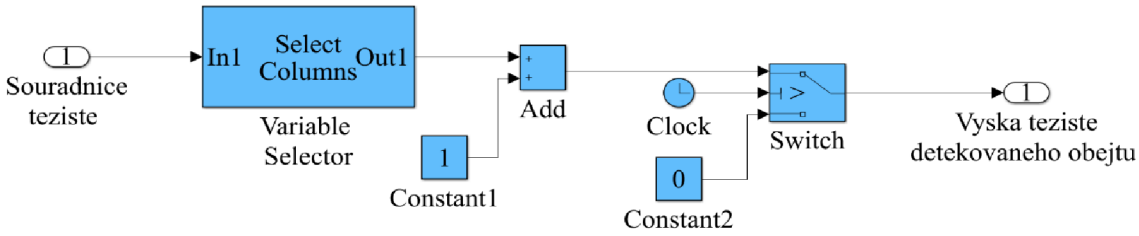
Obr. č. 21: Detekce osoby ve vystřiženém obraze

Obr. č. 22: Binární obraz, ve kterém bílé pixely znázorňují detekovaný objekt

4.3 Zjištění odchozích nebo příchozích osob

Další částí algoritmu počítání osob je zjišťování, zda se jedná o osobu odchozí nebo příchozí. Výstupem z bloku *Blob Analysis* (blok v subsystému detekce osob) je řádková matice, která obsahuje souřadnice těžiště. Výška těžiště se nachází ve druhém sloupci, proto je nutné vybrat tuto souřadnici. Při výpočtu těžiště v bloku *Blob Analysis* osob dochází k trvalé odchylce, proto je nutné tuto chybu kompenzovat připočtením konstanty o velikosti 1.

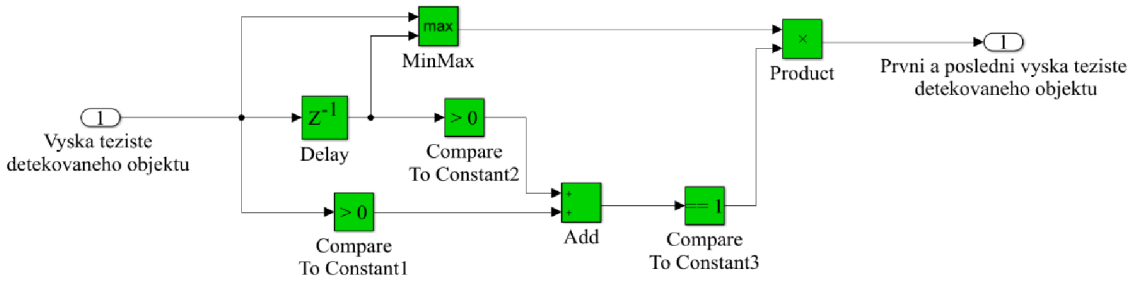
Dále je nutné kompenzovat dobu, po kterou dochází k ustálení hodnot při výpočtu průměrných hodnot ve filtraci. Ta je kompenzována pomocí bloků *Switch, Clock* a *Constant2* a to tak, že první tři sekundy od spuštění modelu je výstupem bloku nula (obr. č. 23).



Obr. č. 23: Subsystém, ve kterém dochází k výběru polohy těžiště

V další části modelu Simulinku je třeba rozlišit, zda se jedná o osobu, která do Mechlabu přichází nebo odchází. Tato část probíhá ve dvou krocích.

Prvním krokem je výběr první a poslední detekované polohy těžiště objektu v obraze (obr. č. 24).

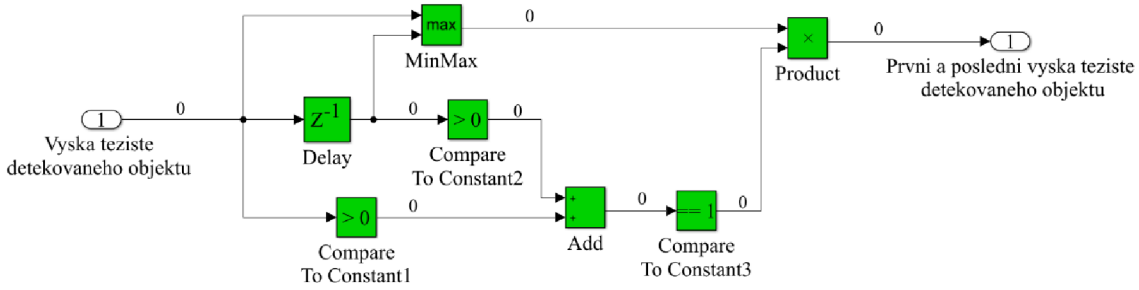


Obr. č. 24: Subsystém, ve kterém dochází k výběru první a poslední polohy detekovaného objektu

Tento úsek modelu pracuje následovně: výška těžiště detekovaného objektu v obraze se v bloku *MinMax* porovnává s výškou těžiště z minulého z cyklu. Výstupní hodnotou tohoto bloku je větší z obou výšek. Hodnotu výšky z minulého cyklu zajišťuje blok *Delay*, jehož výstupní hodnotou je vždy hodnota výšky těžiště z minulého cyklu. Tato hodnota je následně porovnávána v bloku *Compare To Constant2*. Výstupní hodnotou tohoto bloku je vždy 0 nebo 1 podle toho, zda je porovnávaná hodnota vyšší než 0, stejně jako v případě v bloku *Compare To Constant1*, kde je porovnávána aktuální hodnota výšky. Výstupní hodnoty z bloků *Compare To Constant2* a bloků *Compare To Constant1* jsou sčítány v bloku *Add*. Součet je porovnán, zda se rovná 1 v bloku *Compare to Constant3*, jehož výstupní hodnotou může být opět 0 nebo 1. Výstupní hodnota z bloku *Compare to Constant3* je v bloku *Product* násobena s výstupní hodnotou z bloku *MinMax*.

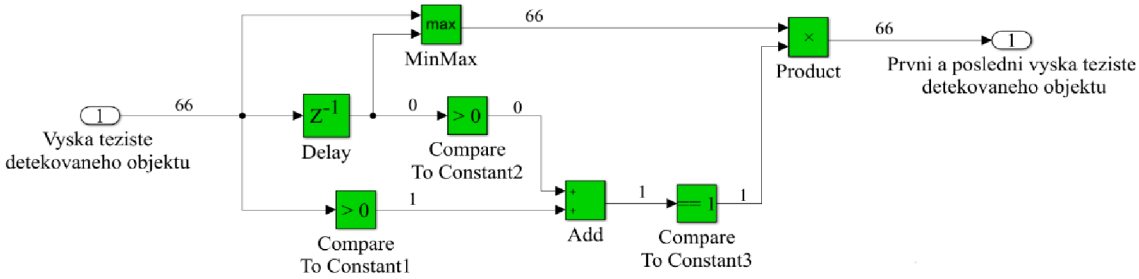
Bude-li uvažován případ, kdy výška těžiště detekovaného objektu bude postupně nabývat hodnot 0, 66, 56, 47 a 0 (odchozí osoba), část modelu se bude chovat následovně:

V prvním cyklu, kdy výška těžiště detekovaného objektu nabývá hodnoty 0 (v obraze se nenachází žádný objekt), budou výstupy jednotlivých bloků vypadat následovně (obr. č. 25):



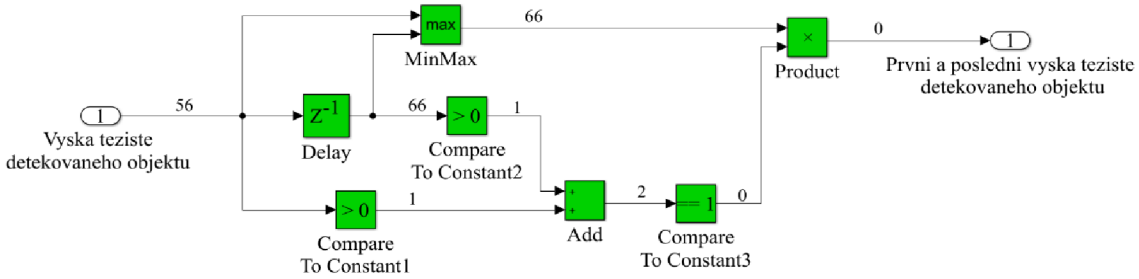
Obr. č. 25: Výstupní hodnoty jednotlivých bloků subsystému v prvním cyklu

Ve druhém cyklu, kdy výška těžiště detekovaného objektu nabývá hodnoty 66 (v obraze se objevil objekt), budou výstupy jednotlivých bloků následující (obr. č. 26):



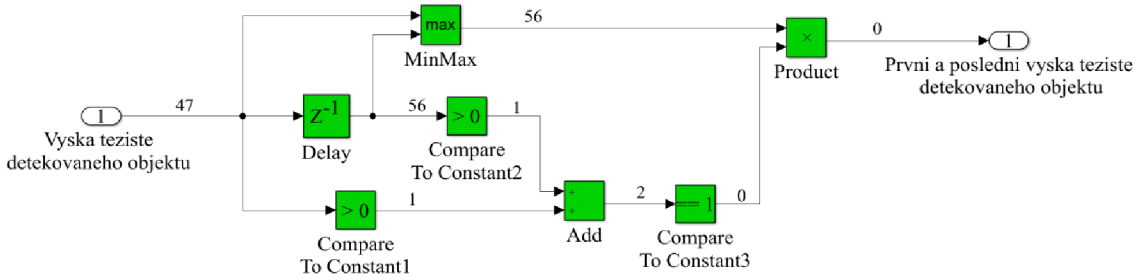
Obr. č. 26: Výstupní hodnoty jednotlivých bloků subsystému ve druhém cyklu

Ve třetím cyklu, kdy výška těžiště detekovaného objektu nabývá hodnoty 56 (v obraze je detekován objekt), budou výstupy jednotlivých bloků vypadat následovně (obr. č. 27):



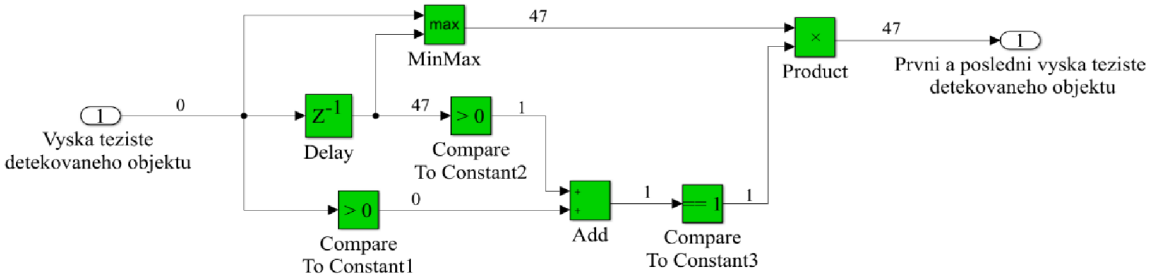
Obr. č. 27: Výstupní hodnoty jednotlivých bloků subsystému ve třetím cyklu

Ve čtvrtém cyklu, kdy výška těžiště detekovaného objektu nabývá hodnoty 47 (v obraze se nachází objekt), budou výstupy jednotlivých bloků vypadat následovně (obr. č. 28):



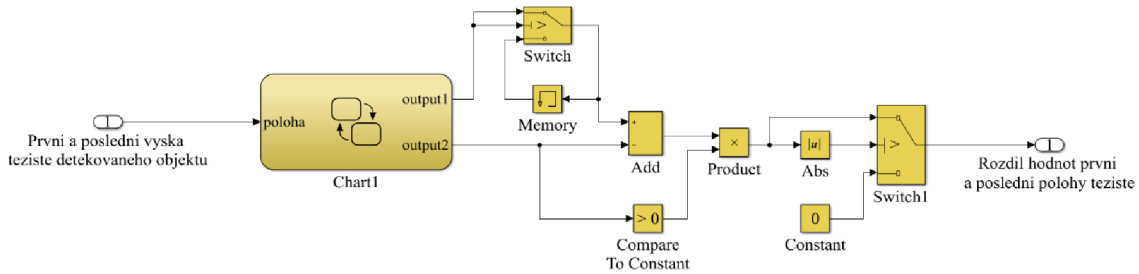
Obr. č. 28: Výstupní hodnoty jednotlivých bloků subsystému ve čtvrtém cyklu

V pátém cyklu, kdy výška těžiště detekovaného objektu nabývá hodnoty opět 0 (v obraze se nenachází žádný objekt), budou výstupy jednotlivých bloků vypadat následovně (obr. č. 29):

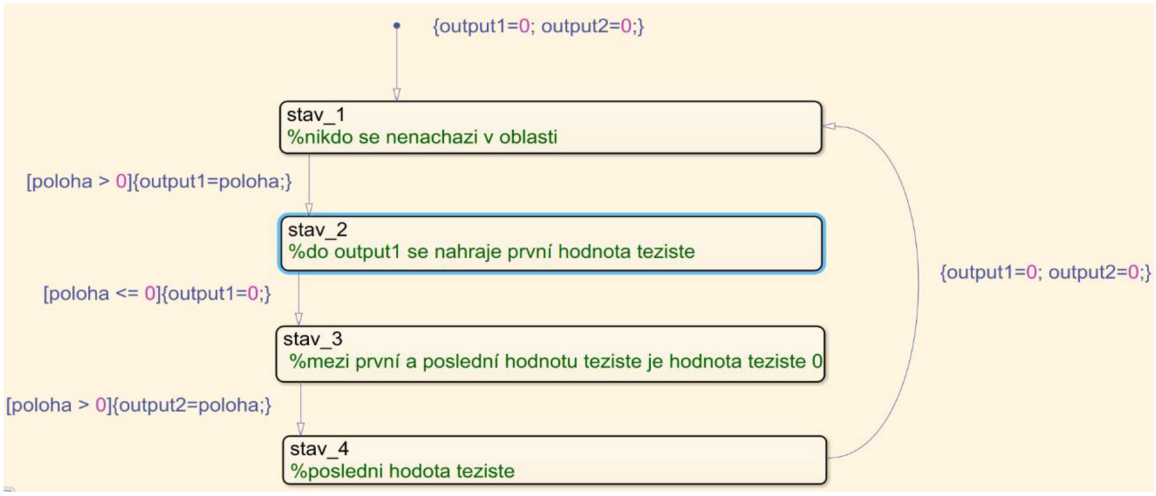


Obr. č. 29: Výstupní hodnoty jednotlivých bloků subsystému v pátém cyklu

Druhým krokem v rozlišení, zda se jedná o odchozí nebo příchozí osobu, je odečtení první a poslední detekované hodnoty polohy těžiště (větší než 0) v obraze. Je-li tento rozdíl kladný, pak se jedná o osobu odchozí, v případě záporného rozdílu se jedná o osobu příchozí (obr. č. 30).



Obr. č. 30: Subsystém, ve kterém dochází k rozdílu první a poslední polohy těžiště



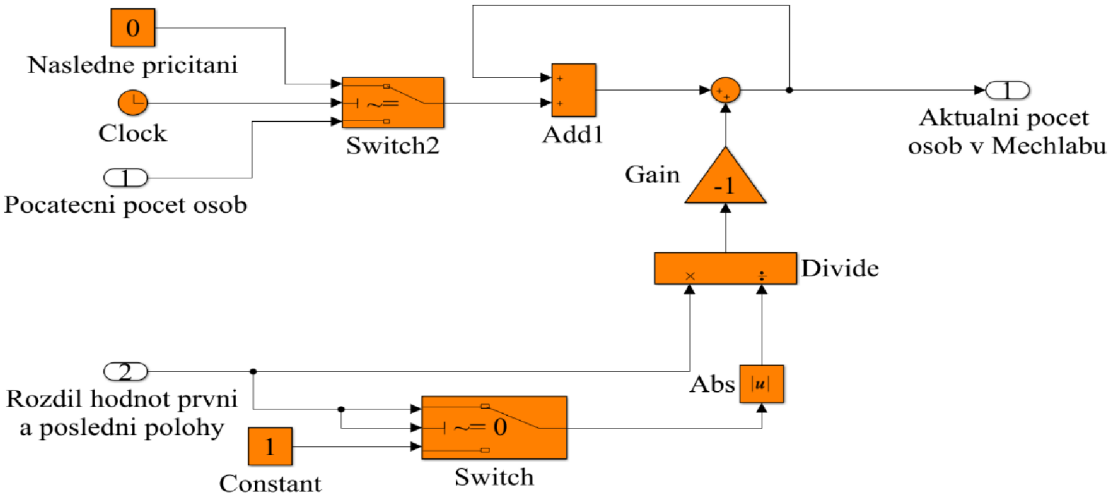
Obr. č. 31: Rozdělení první a poslední polohy v bloku Charat1

Výstupem předešlého subsystému jsou dvě nenulové hodnoty poloh těžiště a několik nulových. V bloku *Chart1* (obr. č. 31) dochází k oddělení těchto dvou nenulových hodnot. Výstup *Output1* (obr. č. 30) nabývá první nenulové hodnoty a výstup *Output2* druhé nenulové hodnoty. Jakým postupem byly tyto hodnoty vybrány je vidět na obr. č. 18. Z obrázku č. 31 vyplývá, že bylo využito čtyř stavů: *První stav* nastane, je-li poloha těžiště rovna 0. *Druhý stav* nastává, je-li splněna podmínka, že poloha je větší než 0, v tomto stavu bude výstupní hodnotou *Output1* aktuální poloha těžiště. *Třetí stav* nastane, je-li poloha rovna 0, v tomto stavu dochází k vynulování *Output1*. *Čtvrtý stav* nastane, je-li poloha těžiště větší než 0 a výstupní hodnotou *Output2* bude druhá nenulová hodnota těžiště. Po tomto stavu následuje opět první stav a dochází k vynulování obou výstupních hodnot *Output1* a *Output2*.

Jelikož mezi nenulovými výstupními hodnotami *Output1* a *Output2* je časový rozdíl, je využíváno bloků *Memory a Switch*. Model Simulinku pracuje následovně: pokud je výstupem *Output1* nenulová hodnota, hodnota se nahraje do bloku *Memory* a setrvává zde do té doby, než je přepsána další nenulovou výstupní hodnotou *Output1*. Než však dojde k přepsání, výstupní hodnota *Output2* nabývá nenulové hodnoty. Od výstupní hodnoty z bloku *Switch1* (první poloha těžiště detekovaného objektu) je odečtena výstupní hodnota *Output2*. Za blokem *Add* následuje blok *Product*. Tato funkce společně s blokem *Compare To Konstant* zajišťuje, že hodnoty se odečítají od sebe pouze jednou. Pokud je výsledkem těchto hodnot hodnota záporná, pak se jedná o příchozí osobu, pokud je výsledná hodnota kladná, jedná se o odchozí osobu.

Poslední část tohoto subsystému zamezuje přičtení nebo odečtení osoby od aktuálního stavu v případech, kdy si detekovaný objekt rozmyslí svůj vstup do Mechlabu a ve vyříznuté oblasti změní směr. Tato situace je ošetřena pomocí bloku *Switch1* následovně: rozdíl první a poslední polohy těžiště detekovaného objektu musí být větší než 9. Pokud tato skutečnost neplatí, výstupem z bloku *Switch1* je 0.

4.4 Aktualizace počtu osob



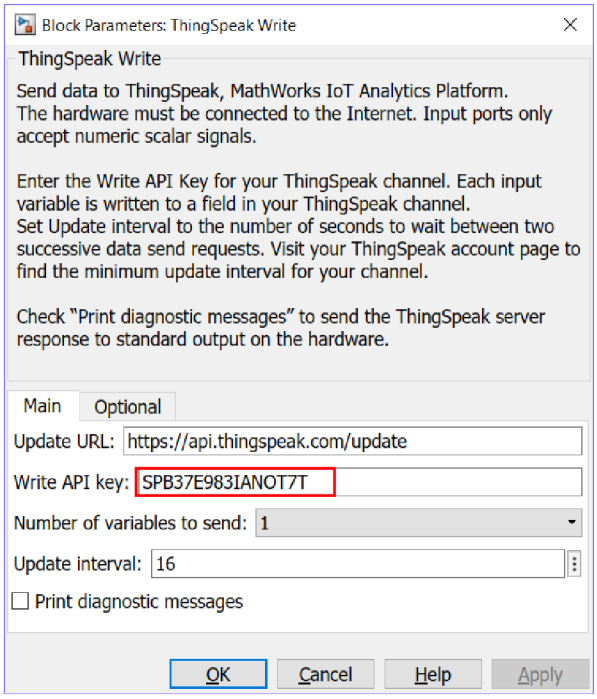
Obr. č. 32: Subsystém, ve kterém dochází k aktualizaci počtu osob

Protože velikost vyříznutého obrazu je určena tak, aby se v ní, pokud možno, nacházela pouze jedna osoba, dochází k přičtení nebo odečtení čísla 1 od aktuálního počtu osob. Zda se jedná o příchozí nebo odchozí osobu zjišťujeme v minulém subsystému odečtením první a poslední polohy těžiště detekované osoby. Vstupem do subsystému *Aktualizace počtu osob* je tedy rozdíl hodnot první a poslední polohy a počáteční počet osob v Mechlabu. Rozdíl hodnot první a poslední polohy může nabývat jak hodnoty 0, tak kladných i záporných hodnot. V případě nulové hodnoty dojde v bloku *Divide* k násobení nulou, takže k aktuálnímu počtu osob je přičtena nula. Pokud je hodnota kladná, dojde v bloku *Divide* k dělení dvou stejných kladných čísel (v blocích *Switch* a *Abs* se totiž hodnota nijak nezmění), jehož výsledkem je 1. Tato hodnota se násobí -1, protože se jednalo o odchozí osobu, takže od aktuálního počtu osob je odečítána 1. Pokud je však hodnota již zmiňovaného rozdílu záporná, dojde v bloku *Abs* k vytvoření absolutní hodnoty tohoto záporného čísla. V bloku *Divide* je tedy děleno kladné číslo záporným, ve zlomku nabývají hodnoty stejných velikostí. Výstupem z bloku *Divide* je -1, která je násobená -1, protože se jedná o přichozí osobu do Mechlabu. K aktuálnímu počtu osob je připočítána 1 nebo -1 pomocí bloku *Sum*.

Počáteční počet osob v Mechlabu je nahrán v čase 0. Návaznost na hodnotu počtu osob v Mechlabu z minulého cyklu je realizováno pomocí bloku *Add,* kde dochází k sečtení počtu osob z minulého cyklu a 0 (hodnota počátečního počtu osob se započítá v čase 0 pouze jednou).

4.5 Odeslání dat do platformy ThingSpeak

Odesílání dat do platformy ThingSpeak je realizováno blokem *ThingSpeak Write*. V tomto bloku je nutné zadat tzv. WriteKey[4], počet proměnných, které chceme odesílat. Jelikož je využíván bezplatný účet, je nutné v bloku zadat časový interval pro odesílání dat do ThingSpeaku větší než 15 sekund. URL adresa byla v tomto bloku vyplněna automaticky (obr. č. 33). WriteKey je možné nálézt v účtu na ThingSpeaku na kartě *API Keys* .



Obr. č. 33: Parametry bloku ThingSpeak Write

Algoritmus detekce příchozích a odchozích osob z Mechlabu, využívající metodu optického toku, byl vytvořen v Simulinku a je nasazen přímo na Raspberry Pi. Tento algoritmus je schopný určit aktuální počet osob v prostorách Mechlabu a tato data potom odeslat do platformy ThingSpeak, kde dochází k jejich zpracování. Přesto přináší určitá omezení. První z nich je, že předpokládáme nenulovou hodnotu gradientu intenzity jasu. Druhé omezení spočívá v předpokladu, že nesmí přicházet nebo odcházet více osob najednou, tedy nesmí být osoby velmi blízko u sebe, neboť dojde ke splynutí osob v binárním obraze a s těmito osobami je v algoritmu počítáno jako s jednou osobou. Třetím omezením je doba, při které dochází k výpočtu a ustálení průměrných hodnot ve filtraci. Čtvrtým omezením je vyšší náročnost výpočtu optického toku.

[4] Klíče API umožňují zapisovat data do kanálu nebo číst data ze soukromého kanálu. Klíče API jsou generovány automaticky při vytváření nového kanálu.

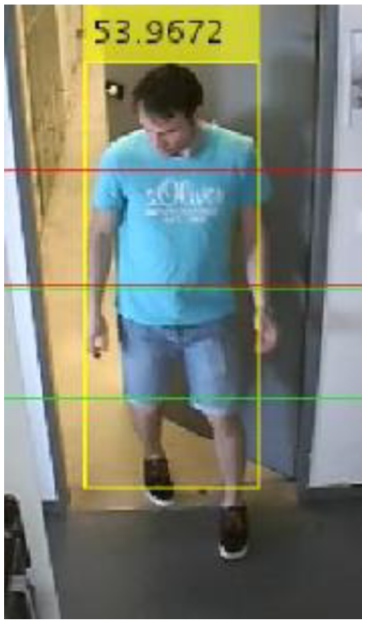
5 Algoritmus softwaru v MATLABu

Schéma algoritmu počítání osob ve skriptu MATLABu je totožné se schématem algoritmu počítání osob v Simulinku. Do tohoto algoritmu byl implementován detektor pracující na základě metody *ACF*. V této části byla využita kamera AirLive POE-260CAM.

5.1 Metoda ACF - Aggregate Channel features

Jak již bylo zmíněno v kapitole *2.7 Metody detekce osob* této bakalářské práce, do skriptu MATLABu byla implementována metoda detekce osob tzv. metoda *ACF - Aggregate Chnnel features*. Metoda *ACF* je vhodná pro detekci objektů, jako jsou například chodci, tváře osob, auta a je založena na tzv. příznacích. Pomocí těchto příznaků jsou reprezentovány podstatné informace o obrazu. Patří sem například normalizovaná amplituda gradientu a histogramy orientovaných gradientů. Pro tuto skupinu implementovaných příznaků je používáno slovo *Feature*. K výcviku *ACF* detektoru byl použit *AdaBoots* algoritmus. Tento učící se algoritmus kombinuje rozhodnutí několika "jednodušších" klasifikátorů a dosahuje tak lepších výsledků, než by bylo možno dosáhnout použitím klasifikátorů samotných. Pro praktickou část bakalářské práce bylo možné požít *HOG* nebo *ACF* detektor, ale byl vybrán detektor založen na metodě *ACF*, protože dosahuje lepších výsledků a je rychlejší, něž detektor založen na metodě *HOG* [23], [24] a [25].

Detektor je možné nalétzt v *Computer Vision Systém Toolbox* a aplikovat jej příkazem *detectPeopleACF*. Do argumentu tohoto příkazu je nutné napsat proměnou, která reprezentuje obraz. Tento příkaz vrací jednořádkovou matici, ve které se nachází čísla vyjadřující souřadnici x, souřadnici y, výšku a délku obedélníku označující osobu v obraze. Souřadnice x a y jsou souřadnicemi levého rohu obdélníku (obr. č. 34).



Obr. č. 34: Ukázka detekce osoby metodou ACF

5.2 Zjištění odchozích nebo příchozích osob

Ve skriptu dochází k detekci osoby v každém cyklu. Pokud bude v cyklu detekována osoba, dojde k jejímu označení do boxu a zjištění těžiště tohoto boxu v obraze. Zjištění, zda se jedná o příchozí nebo odchozí osobu probíhá následovně. Na obrázku č. 45 jsou vidět čtyři čáry - dvě zelené a dvě červené. V oblasti mezi červenými čarami se velikost výšky těžiště v obraze nahraje do proměnné - *první bod*, v případě oblasti mezi zelenými čarami se velikost výšky těžiště v obraze nahraje do proměnné - *druhý bod*.

Níže je uvedená část kódu, která je zodpovědná za detekci, označení osoby, výpočet těžiště označeného objektu v obraze a nahrání hodnoty těžiště do proměnných - *první bod* a *druhý bod*.

ZAČÁTEK ZKRÁCENÉHO KÓDU

```
%detekce osoby
[bboxes2,scores] = detectPeopleACF(vyriznuty_obraz);

%zjištění, zda se v obraze nachází osoba
detekovana_osoba=isempty(bboxes2);

%žádná detekovaná osoba
if detekovana_osoba==1;
vyska_oznaceneho_objektu=0;

%v oblasti byla detekována osoba
else
%souřadnice x, souřadnice y, výška a délka obdélníku označující osobu
v obraze
sour_x=bboxes2(1,1);
sour_y=bboxes2(1,2);
sirka=bboxes2(1,3);
vyska=bboxes2(1,4);

%výpočet těžiště označené osoby
box2=polyshape([sour_x sour_x+sirka sour_x+sirka sour_x],...
        [sour_y sour_y sour_y+vyska sour_y+vyska]);
[delka_oznaceneho_objektu,vyska_oznaceneho_objektu]=centroid(box2);

%označení detekované osoby
imgine2= insertObjectAnnotation(vyriznuty_obraz,'rectangle',...
        bboxes2,scores);
end

%zaznamenání hodnoty výšky detekované osoby mezi červenými čarami
if(vyska_oznaceneho_objektu>y1 & vyska_oznaceneho_objektu<y2)
     prvni_bod=vyska_oznaceneho_objektu;
end

%zaznamenání hodnoty výšky detekované osoby mezi zelenými čarami
if(vyska_oznaceneho_objektu>y3 & vyska_oznaceneho_objektu<y4)
     druhy_bod=vyska_oznaceneho_objektu;
end
```

KONEC KÓDU

Bude-li uvažován případ příchozí osoby do Mechlabu, osoba bude nejprve detekována oblastí obrazu mezi červenými čarami. V tomto úseku jí bude přiřazen status příchozí osoby, jenž spočívá v přiřazení 1 do proměnné s názvem *osoba*. Podmínkou pro přiřazení je, aby proměnná - *první bod* - měla hodnotu jinou než 0 a proměnná - *druhý bod* - měla hodnotu 0. Pokud detekovaná osoba projde oblastí mezi zelenými čarami, velikost polohy těžiště v obraze se nahraje do proměnné - *druhý bod*. Tímto okamžikem bude splněna podmínka pro započtení příchozí osoby a k aktuálnímu počtu osob se připočte 1.

V případě, že osoba bude odcházet z Mechlabu, bude nejdříve detekována oblastí mezi zelenými čarami. V tomto úseku jí bude přiřazen status odchozí osoby, tedy do proměnné s názvem *osoba* se přiřadí -1. Podmínkou pro přiřazení je, aby proměnná - *první bod* - měla hodnotu 0 a proměnná - *druhý bod* - měla hodnotu jinou než 0. V cyklu, kdy detekovaná osoba překročí červenou čáru, velikost polohy těžiště v obraze se nahraje do proměnné - *první bod*. Tímto okamžikem bude splněna podmínka pro odečtení odchozí osoby a od aktuálního počtu osob se odečte 1.

Vždy po odečtení nebo přičtení osoby jsou proměnné - *první bod* a *druhý bod* vynulovány.

Níže je uvedena část kódu, která je zodpovědná za správné přičtení nebo odečtení osoby od aktuálního počtu osob v Mechlabu.

ZAČÁTEK ZKRÁCENÉHO KÓDU

```
%přiřazení statusu odchozí osoby
if(prvni_bod==0 & druhy_bod~=0)
   osoba=-1;
end
%přiřazení statusu příchozí osoby
if(prvni_bod~=0 & druhy_bod==0)
   osoba=1;
end

%aktualizace počtu osob v Mechlabu
if(prvni_bod~=0 & druhy_bod~=0)
     %jedná se o příchozí osobu
     if (osoba>0)
     %k aktuálnímu počtu osob v Mechlabu je připočtena příchozí osoba
     pocet_osob=pocet_osob+1
     end

     %jedná se o odchozí osobu
     if (osoba<0)
     %od aktuálního počtu osob v Mechlabu je odečtena odchozí osoba
     pocet_osob=pocet_osob-1
     end
     %vynulování proměnných
     prvni_bod=0;
     druhy_bod=0;
end
```

KONEC KÓDU

5.3 Odesílání dat do platformy ThingSpeak

Odesílání dat do platformy ThingSpeak je realizováno příkazem *thingSpeakWrite*. Do argumentu je nutné napsat číslo kanálu, do kterého chceme data posílat, název proměnné, jejíž hodnotu chceme odesílat do kanálu ThingSpeaku a tzv. WriteKey[5]. Jelikož je využíván bezplatný účet, byl vytvořen časovač, který vykonává tento příkaz v časových intervalech větších než 15 sekund.

Zde je uvedena část kódu, která je zodpovědná za odesílání dat do platfomy ThingSpeak.

ZAČÁTEK ZKRÁCENÉHO KÓDU

```
% objekt časovače
t = timer;
% režim odesílání dat
set(t,'ExecutionMode','fixedRate');
% perioda odesílání dat
set(t,'Period',20);
% příkaz,který je vykonáván
set(t,'TimerFcn','thingSpeakWrite(419745,pocet_osob,''WriteKey'',''SP
B37E983IANOT7T'')');
%spuštění časovače
start(t);
%aktualizace počtu osob v cyklu
thingspeak.Field1 = pocet_osob;
```

KONEC KÓDU

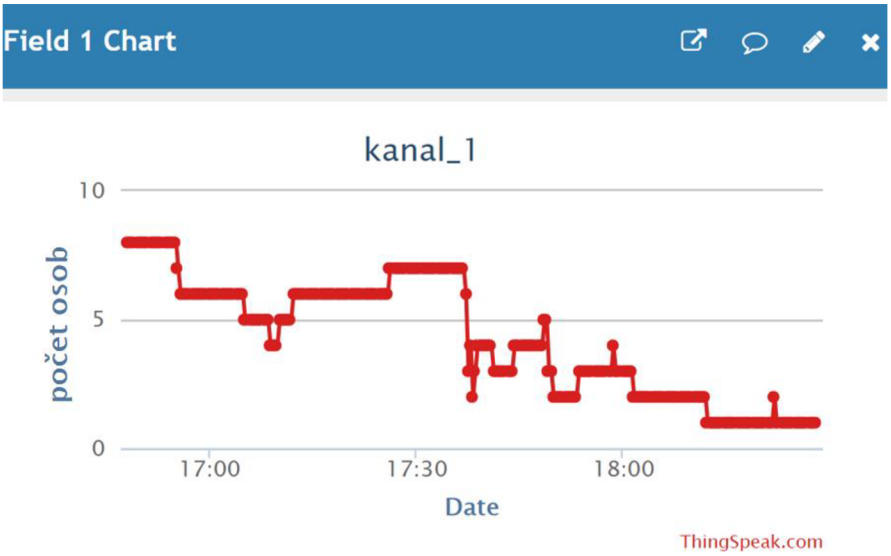
V prostředí MATLAB byl využit vytvořený firmou MathWorks detektor založený na metodě *ACF Aggregate Channel Features* . Jeho algoritmus je opět schopný určit aktuální počet osob v Mechlabu, ale nastane chyba v situaci, kdy osoba má nízké těžiště nebo při odchodu osoba rychle za sebou zavře dveře. Osoba neprojde oběma oblastmi v obraze a není tudíž vůbec započítána. I v případě tohoto detektoru je vyžadována stálá intenzita osvětlení.

Dalším omezením tohoto detektoru je implementace pouze do algoritmu běžícího na hardwaru hostitelského počítače, protože není možné ho generovat v C kódu. Z tohoto důvodu byl stvořen druhý detektor v modelu Simulinku, který je možné zkompilovat[6] a nahrát na minipočítač Raspberry Pi.

[5] Klíče API umožňují zapisovat data do kanálu nebo číst data ze soukromého kanálu. Klíče API jsou generovány automaticky při vytváření nového kanálu.
[6] Úloha kompilace spočívá v převodu zdrojového programu na výsledný [26].

6 Vizualizace dat v platformě ThingSpeak

K vizualizaci dat v platformě ThingSpeak je nutné vytvořit účet na ThingSpeaku. Na tomto účtu byl vytvořen tzv. *kanál,* jehož hlavním úkolem je shromažďovat data. Každý *kanál* obsahuje osm polí, kde je možné posílat data libovolného typu, další tři pole k určení lokalizace pořízených dat a jedno pole pro aktualizaci stavu. Spolu s daty jsou v *kanálu* ukládány časové informace o datech a ID[7]. V případě potřeby je možné data načítat podle času nebo ID. Do *kanálu* je možné posílat data ve formátech JSON, XML a CSV.



Obr. č. 35: Graf vytvořený v platformě ThingSpeak, znázorňující vývoj počtu osob v Mechlabu

Obr. č. 36: Ukázka twitterových statusů publikovaných v platformě ThingSpeak

[7] ID je jedinečné číslo kanálu. Každý kanál má vlastní ID.

7 Závěr

Cílem této práce bylo vytvořit software schopný detekovat osoby, zjistit aktuální počet osob v určeném prostoru a tyto data odeslat do platformy ThingSpeak, kde dojde k jejich zpracování do podoby grafu a statusu na Twitteru.

Teoretické poznatky bakalářské práce jsou zaměřeny na seznámení s levným minipočítaem Raspberry Pi 3, Kamerovým Modulem V2 pro Raspberry Pi, kamerou AirLive POE-260CAM, platformou ThingSpeak, aplikací Internet věcí a s výpočetním a vývojovým prostředím MATLAB a Simulink.

Na teoretickou část navazuje praktická část, v níž je popsáno nejprve propojení počítače Raspberry Pi s platformou ThingSpeak a s hostitelským počítačem. Následuje obeznámení s implementací vytvořeného algoritmu softwarů, které zpracovávají obraz z kamery v MATLABu a Simulinku, jenž jsou v ideálním případě schopné zjistit aktuální počet osob v prostorách Mechlabu a tyto data potom odeslat do platformy ThingSpeak. Údaje shromážděné v platformě ThingSpeak zpracovávají vytvořené aplikace, kde první vykresluje graf vývoje počtu osob v prostorách Mechlabu a druhá píše statusy na sociální síť Twitter.

Test softwarů byl proveden na hostitelském počítači. Při testování byly natočeny různé situace, z nichž je možné vyvodit následující tvrzení: v případě softwaru, ve kterém je aplikován detektor pracující na základě metody *ACF*, nastane chyba pouze v situaci, kdy osoba odchází z Mechlabu v dřepu anebo při odchodu rychle za sebou zavře dveře. Z videa je patrné, že detektor správně označí osobu, avšak z důvodu malé výšky osoby nebo rychlého zavření dveří těžiště neprojde oběma oblastmi v obraze a osoba není vůbec započítána. V případě softwaru pracující s metodou optického toku dochází k chybě v detekci v situacích, ve kterých jsou osoby v obraze příliš blízko u sebe. V tomto případě dochází ke splynutí osob v binárním obraze a s těmito osobami je v algoritmu počítáno jako s jednou osobou.

I když cílem bakalářské práce nebylo vytvoření Stand-alone systému, přesto pokusy se tímto směrem ubíraly. V případě softwaru pracující s metodou *ACF* úsilí selhalo při kompilování, detektor vytvořený firmou MathWorks není možné tedy zkompilovat. Software vytvořený v Simulinku, jenž pracuje s metodou optického toku, byl zkompilován, přesto při dlouhodobém používání software přestal správně pracovat. Bohužel se nepodařilo zjistit příčinu.

8 Použité zdroje a literatura

[1] UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. 2., aktualizované vydání. Přeložil Jakub GONER. Brno: Computer Press, 2016. ISBN 978-80-251-4819-8.

[2] Raspbery Pi 3. CzC.cz. [online]. oXyShop X7, 2018 [cit. 2018-05-22]. Dostupné z: https://www.czc.cz/raspberry-pi-3/187359/produkt

[3] VÍTEK, Jan. Raspberry Pi 3 je tu, co nabídne? Svět hardware [online]. oXyMedia, 2016. [cit. 2018-05-22]. Dostupné z: http://www.svethardware.cz/raspberry-pi-3-je-tu-conabidne/41920

[4] HORÁČEK, Petr. Raspberry π I. – Úvod. Linuxsoft.cz [online]. Praha: Pavel Kysilka, 2012 [cit. 2018-05-22]. ISSN 1801-3805. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=1937

[5] RASPBERRY PI. Raspberrypi.org [online]. RASPBERRY PI FOUNDATION UK REGISTERED CHARITY 1129409 [cit. 2018-05-22]. Dostupné z: https://www.raspberrypi.org/.

[6] NORRIS, Donald. Raspberry Pi: projekty. Brno: Computer Press, 2015. ISBN 978-80-251-4346-9.

[7] HUMUSOFT ve službách vědy a techniky. MATLAB & Simulink. ThingSpeak. Humusoft.cz [online]. © HUMUSOFT 1991 - 2018 . [cit. 2018-05-22]. Dostupné z: http://www.humusoft.cz/matlab/thingspeak/.

[8] VOJÁČEK, Antonín. Základní úvod do oblasti internetu věcí (IoT). automatizace.hw.cz [online]. HW server s.r.o. 2016 [cit. 2018-05-22]. Dostupné z: https://automatizace.hw.cz/zakladni-uvod-do-oblasti-internetu-veci-iot.htm.

[9] Raspberry Pi 8MP Kamerový modul V2. SmarterHOME [online]. [cit. 2018-05-22]. Dostupné z: https://smarterhome.sk/cs/zwave-prislusenstvi/raspberry-pi-8mp-kamerovy-modul-v2-392.html.

[10] AirLive POE-260CAM AirLive POE-260CAM. Alza.cz [online]. [cit. 2018-05-22]. Dostupné z: https://www.alza.cz/ovislink-poe-260cam-d140585.htm.

[11] Matlab a internet věcí. SCIENCEmag.Tiskové zprávy [online]. © Copyright 2018, Nitemedia s.r.o. [cit. 2018-05-22]. Dostupné z: http://sciencemag.cz/matlab-a-internet-veci/.

[12] HUMUSOFT ve službách vědy a techniky. MATLAB & Simulink Matlab. Humusoft.cz [online]. © HUMUSOFT 1991 - 2018 . [cit. 2018-05-22]. Dostupné z: http://www.humusoft.cz/matlab/details/.

[13] HUMUSOFT ve službách vědy a techniky. MATLAB & Simulink. Simulink. Humusoft.cz [online]. © HUMUSOFT 1991 - 2018 . [cit. 2018-05-22]. Dostupné z: http://www.humusoft.cz/matlab/simulink/.

[14] DADEJ, V. Raspberry Pi: programování v prostředí Matlab/Simulink. Brno, 2017. 96 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Dostupné také z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=151082.

[15] HOLOUBEK, T. Raspberry Pi: programování v prostředí Matlab/Simulink. Brno, 2017. 84 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Dostupné také z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=149396.

[16] MAŠEK, Jan Detekce objektů v obraze s pomocí Haarových příznaků. Brno, 2012. 72 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Dostupné také z: https://dspace.vutbr.cz/bitstream/handle/11012/9494/diplomka.pdf.

[17] KŘIVÁNEK, Filip Detekce částí lidského těla v obraze. Brno, 2014. 52 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Dostupné také z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=87892

[18] PRÍTEL, Pavol. Detekce pohybujících se objektů v obraze. Brno, 2014. 48 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Dostupné také z: https://dspace.vutbr.cz/bitstream/handle/11012/31690/Pritel_Pavol_2014.pdf?sequence=2

[19] ZÍTKA M. Detekce pohybu v obraze. Brno, 2008. 44 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Dostupné také z: https://dspace.vutbr.cz/bitstream/handle/11012/13859/xzitka04_BP.pdf?sequence=.

[20] ŠILAR Z. Metody obrazové detekce obsazenosti železničního přejezdu. Pardubice, 2013. 109 s. Disertační práce. Univerzita Pardubice, Fakulta elektrotechniky a informatiky. Dostupné také z: http://dspace.upce.cz/bitstream/handle/10195/54068/SilarZ_MetodyObrazove_MD_2013.pdf?sequence=3.

[21] MathWorks. Documentation. Optical Flow. Estimate object velocities. ch.mathworks.com [online]. © The MathWorks, Inc. 1994 - 2018. [cit. 2018-05-22]. Dostupné z: https://ch.mathworks.com/help/vision/ref/opticalflow.html?searchHighlight=optical%20flow&s_tid=doc_srchtitle.

[22] Získávání a analýza obrazové informace. Operace s obrazem II. Brno: Biofyzikální ústav Lékařské fakulty Masarykovy univerzity Brno. Med.muni.cz [online]. © BFÚ LF MU [cit. 2018-05-22]. Dostupné z: https://www.med.muni.cz/biofyz/Image/prezentace_4.pdf

[23] HOUROVÁ B. Automatická detekce argumentace. Plzeň, 2017. 72 s. Diplomová práce. Západočeská univerzita v Plzni, Fakulta informatiky a výpočetní techniky. Dostupné také z: https://dspace5.zcu.cz/bitstream/11025/27162/1/Barbora%20Hourova%20-%20ZCU%20-%20Diplomova%20prace.pdf.

[24] ŠOCHMAN J. Cvičení z RPZ – AdaBoos. cmp.felk.cvut.cz. [online]. 2005-05-26. [cit. 2018-05-22]. Dostupné z: http://cmp.felk.cvut.cz/cmp/courses/recognition/Labs/adaboost/adaboost.pdf.

[25] HAL archives-ouvertes.fr. A Comparative View on Exemplar'Tracking-by-Detection' Approaches. Hal.archives-ouvertes.fr. [online]. 2015-08-23 [cit. 2018-05-22]. Dostupné z: https://hal.archives-ouvertes.fr/hal-01202882/document.

[26] Vyhledávání ve slovníku IT-Slovník.cz. IT SLOVNÍK.cz [online]. 2018 [cit. 2018-05-22]. Dostupné z: https://it-slovnik.cz/vyhledavani/?text.

[27] HLAVÁČ. V. Matematická morfologie. People.ciirc.cvut.cz [online]. 2018 [cit. 2018-05-22]. Dostupné z: http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/71-3MatMorpholBinCz.pdf

9 Seznam obrázků

Obr. č. 1: Raspberry Pi 3 Model B 10
Obr. č. 2: Karta HOME s označenou ikonou Add-Ons 14
Obr. č. 3: Okno s podpůrnými balíčky a vyhledávačem hardwaru 15
Obr. č. 4: Okno podpůrného balíčku s ikonami Learn More a Manage 15
Obr. č. 5: Okno s podpůrnými balíčky a s ikonami Setup 16
Obr. č. 6: Výběr hardwaru 16
Obr. č. 7: Výběr operačního systému 16
Obr. č. 8: Výběr síťového spojení 16
Obr. č. 9: Výběr jednotky 16
Obr. č. 10: Zapsání firmwaru na SD kartu 17
Obr. č. 11: Postup zapojení Raspberry Pi 17
Obr. č. 12: Okno úspěšného spojení Raspberry Pi s Matlabem 17
Obr. č. 13: Schéma algoritmu softwaru v modelu Simulinku 19
Obr. č. 14: Blok, ve kterém dochází k pořízení obrazu 19
Obr. č. 15: Schéma algoritmu detekce objektu v Simulinku 20
Obr. č. 16: Subsystém, ve kterém dochází k úpravě obrazu 20
Obr. č. 17: Blok Optical flow 22
Obr. č. 18: Subsystém, ve kterém dochází k filtraci a vytvoření binárního obrazu 23
Obr. č. 19: Subsystém, ve které dochází k označení objektu 23
Obr. č. 20: Původní obraz v intenzitě šedé 24
Obr. č. 21: Detekce osoby ve vystřiženém obraze 24
Obr. č. 22: Binární obraz, ve kterém bílé pixely znázorňují detekovaný objekt 24
Obr. č. 23: Subsystém, ve kterém dochází k výběru polohy těžiště 25
Obr. č. 24: Subsystém, ve kterém dochází k výběru první a poslední polohy detekovaného objektu 25
Obr. č. 25: Výstupní hodnoty jednotlivých bloků subsystému v prvním cyklu 26
Obr. č. 26: Výstupní hodnoty jednotlivých bloků subsystému ve druhém cyklu 26
Obr. č. 27: Výstupní hodnoty jednotlivých bloků subsystému ve třetím cyklu 26
Obr. č. 28: Výstupní hodnoty jednotlivých bloků subsystému ve čtvrtém cyklu 27
Obr. č. 29: Výstupní hodnoty jednotlivých bloků subsystému v pátém cyklu 27
Obr. č. 30: Subsystém, ve kterém dochází k rozdílu první a poslední polohy těžiště 27
Obr. č. 31: Rozdělení první a poslední polohy v bloku Charat1 28
Obr. č. 32: Subsystém, ve kterém dochází k aktualizaci počtu osob 29
Obr. č. 33: Parametry bloku ThingSpeak Write 30
Obr. č. 34: Ukázka detekce osoby metodou ACF 31
Obr. č. 35: Graf vytvořený v platformě ThingSpeak, znázorňující vývoj počtu osob v Mechlabu 35
Obr. č. 36: Ukázka twitterových statusů publikovaných v platformě ThingSpeak 35

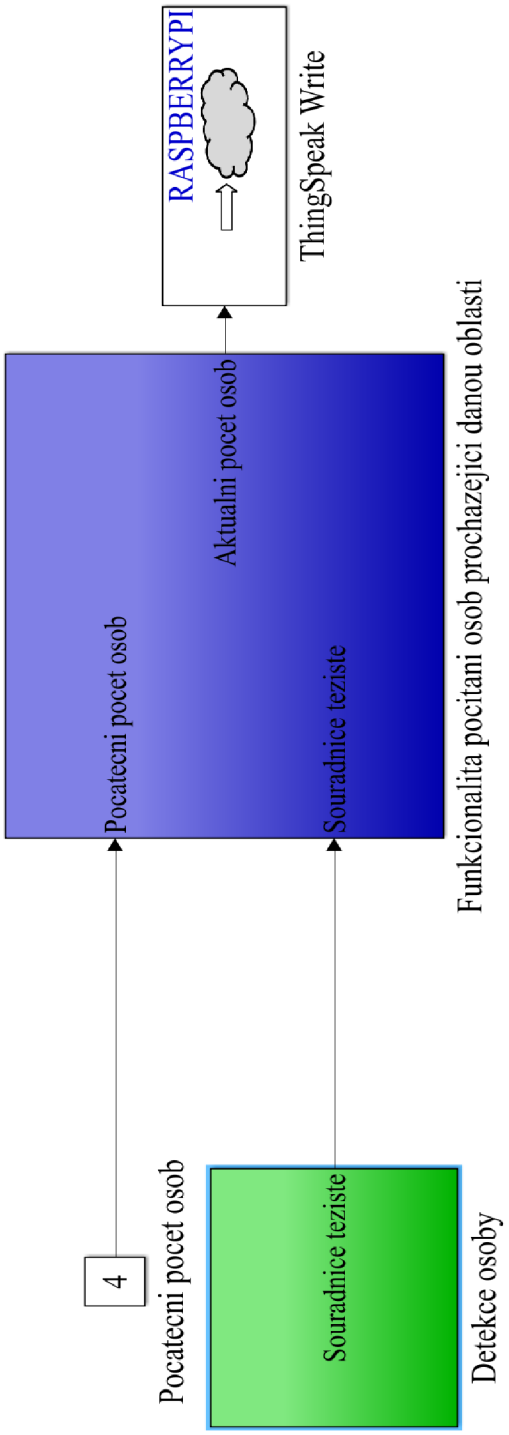
10 Seznam zkratek

Wi-Fi	*Wireless Fidelity*	komunikační standard pro bezdrátový přenos dat
ARM	*Acorn RISC Machine*	označení architektury procesorů
GHz,MHz	*Gigahertz, Megahertz*	jednotky frekvence
GB	*Gigabyte*	jednotky množství informace používané v informatice
SD	*Secure Digital*	paměťová karta
HDMI	*High-Definition Multimedia Interface*	digitální rozhraní používané mezi zdrojem audio a video signálu
GPIO	*General-purpose input/output*	programovatelné piny
HTTP	*HyperText Transfer Protocol - Secure*	bezpečná verze hypertextového přenosového protokolu
PoE	*Power over Ethernet*	technologie, která umožňuje síťovým kabelem přenášet elektrickou energii
IP	*Internet Protocol*	Standardní síťový protokol
URL	*Uniform Resource Locators*	Adresa pro přesnou identifikaci dokumentů na internetu
API	*Application Programming Interface*	Programové rozhraní aplikace
LAN	*Local Area Network*	lokální síť (počítačová)

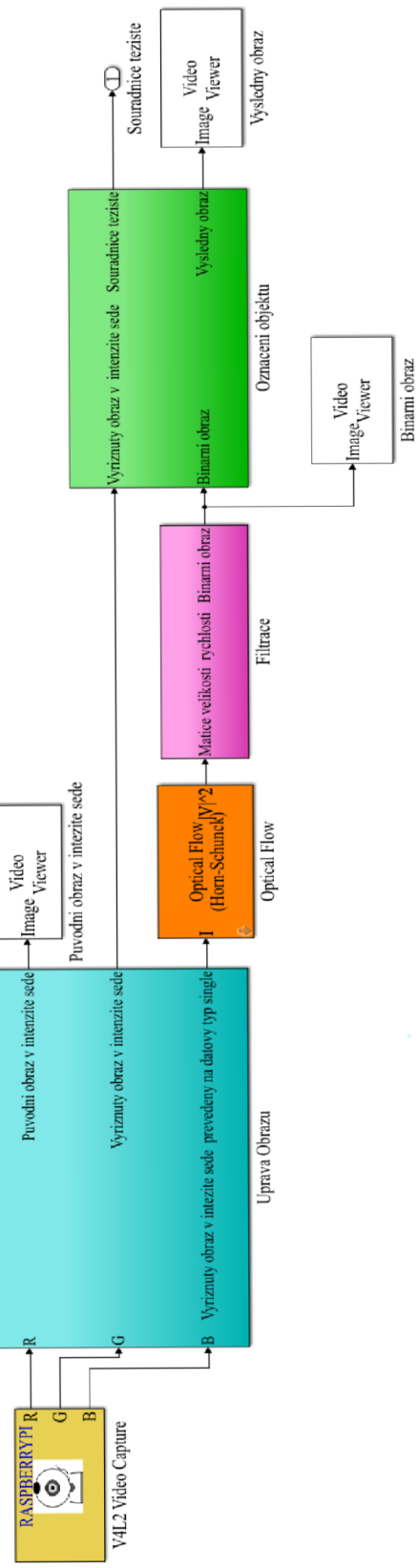
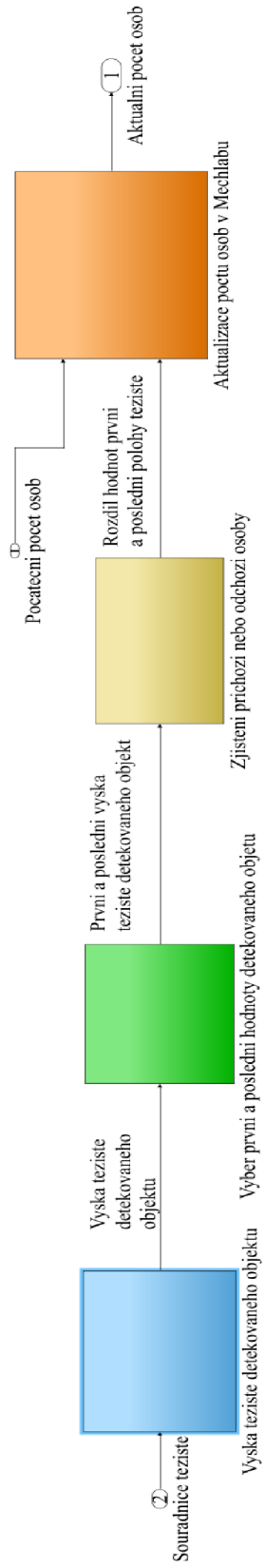
11 Seznam příloh

1. **Příloha č. 1:** Hlavní schéma modelu Simulinku
2. **Příloha č. 2:** Subsytém *Detekce osoby*
3. **Příloha č. 3:** Subsystém *Funkcionalita pocitani osob prochazejici danou oblasti*
4. **CD obsahující**:model Simulinku, skript MATLABu, videa zobrazující detekci osob a jejich počítání

Příloha č. 1: Hlavní schéma modelu Simulinku



Příloha č. 2: Subsytém *Detekce osoby*



Příloha č. 3: Subsystém *Funkcionalita pocitani osob prochazejici danou oblasti*