



ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE

Fakulta elektrotechnická
Katedra měření

Bakalářská práce

Vestavný systém zpracování obrazu pro počítání objektů

Richard Steidl

Květen 2017

Vedoucí práce: doc. Ing. Jan Fischer, CSc.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Richard Steidl
Studijní program:	Otevřená informatika
Obor:	Počítačové systémy
Název tématu česky:	Vestavný systém zpracování obrazu pro počítání objektů
Název tématu anglicky:	Embedded Image Processing System for Object Counting

Pokyny pro vypracování:

Navrhněte a realizujte systém pro vyhodnocení počtu uzavřených a izolovaných objektů přítomných ve snímané scéně využitím vhodné metody zpracování obrazu. Systém určí počet, velikost a polohu jednotlivých objektů. Využijte moduly s mikrořadičem STM32F4xx s připojeným obrazovým senzorem CMOS. Systém se bude konfigurovat pomocí PC prostřednictvím rozhraní USB. Výstup informace o stavu scény bude prostřednictvím zobrazovače a také hlasovým výstupem. Vytvořte potřebné programové vybavení pro mikrořadiče i aplikaci pro nadřazené PC. Ověřte správnou funkci realizovaného systému při proměnlivých okolních světelných podmínkách.

Seznam odborné literatury:

- [1] Yiu, J.: The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors,
- [2] Dokoupil, V.: Zpracování obrazu vestavným mikrořadičem, bakalářská práce, ČVUT – FEL
- [3] STMicroelectronics: RM0090, STM32F4 Reference manual

Vedoucí bakalářské práce:	doc. Ing. Jan Fischer, CSc.
Datum zadání bakalářské práce:	24. ledna 2017
Platnost zadání do ¹ :	30. září 2018



Prof. Ing. Jan Holub, Ph.D.
vedoucí katedry



Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 24. 1. 2017

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Čestné prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....
Podpis autora práce

Poděkování

Rád bych využil příležitosti zde poděkovat rodině a přátelům za jejich jak psychickou, tak odbornou podporu při tvorbě této práce a rovněž v průběhu celého studia.

Zvláštní poděkování patří též panu doc. Ing. Janu Fischerovi, CSc., jakožto vedoucímu této práce, bez jehož rad, motivace a odborného pohledu na věc by tato práce nejspíše nebyla nikdy dokončena.

Abstrakt

Práce analyzuje jednotlivé standardní metody segmentace obrazu a nastiňuje jejich možnost pro využití za nízkých výpočetních a paměťových kapacit, které vestavné systémy nabízejí. V rámci práce byly některé metody následně modifikovány pro jejich efektivní využití ve vestavných systémech.

V rámci této práce jsou vybrané metody a jejich modifikace testovány na vestavném mikrořadiči s jádrem ARM Cortex – M4, a výsledné zvolené algoritmy jsou proto laděny primárně pro tento mikrořadič a požadují možnost uchovat obraz v paměti (ne však nutně ve vysoké kvalitě).

Rovněž v rámci práce vzniklo demonstrační zařízení, jež detekuje vržené kostky pod senzorem a určuje součet na nich hozený.

Abstract

The thesis analyses standard image segmentation methods and outlines their capability of being used with low computing and memory capacity, that is offered by embedded systems. In the thesis, there were some of the methods modified for their efficient usage in embedded systems.

The thesis also tests some methods, that are selected based on the analysis, and their modifications on ARM Cortex – M4 microcontroller. Algorithms, that are created are harmonized to work on selected microcontroller and they demand capability of storing whole image in memory (even though the resolution of picture does not need to be necessarily high).

Also there was created a device capable of detecting dices (that are rolled under the sensor) and counting the sum of dots.

Obsah

Prohlášení	1
Poděkování.....	1
Anotace (CZ)	2
Annotation (EN)	2
Seznam obrázků	7
Seznam tabulek	8
1. Úvod	9
2. Rozbor	10
3. Využitý hardware	11
3.1 Obrazový senzor	11
3.1.1 Úvodní popis	11
3.1.2 Parametry	12
3.1.3 Důvod výběru	12
3.1.4 Zapojení	12
3.2 Mikrořadič	13
3.2.1 Úvodní popis	13
3.2.2 Parametry	13
3.2.3 Důvod výběru	14
3.2.4 Nevýhody	14
3.3 Displej	14
3.3.1 Úvodní popis	14
3.3.2 HD44780 standard	15
3.3.3 Důvod výběru	15
3.4 Zapojení	15
3.4.1 Seznam vyvedených pinů	16

4. Vývojová prostředí	17
4.1 Matlab	17
4.1.1 Využití	17
4.2 Keil uV5	17
4.2.1 Využití	17
4.3 AC6	18
4.3.1 Využití	18
4.4 Cube	18
4.4.1 Využití	18
4.4.2 HAL knihovny	19
4.5 QT	19
4.5.1 Využití	19
5. Metody zpracování obrazu	20
5.1 Princip konvoluce	20
5.1.1 Popis	20
5.1.2 Příklad	20
5.2 Prahování (thresholding)	21
5.2.1 Popis	21
5.2.2 Příklad	22
5.2.3 Výhody	22
5.2.4 Nevýhody	22
5.2.5 Možnost aplikace na náš problém	23
5.3 Nálepkování (labeling)	23
5.3.1 Popis	23
5.3.2 Výhody	24
5.3.3 Nevýhody	24
5.3.4 Možnost aplikace na náš problém	24
5.4 Hranová detekce (edge detection)	24
5.4.1 Popis	24
5.4.2 Příklad	25

5.4.3 Výhody	26
5.4.4 Nevýhody	26
5.4.5 Možnost aplikace na náš problém	27
5.5 Cannyho hranový detektor	28
5.5.1 Popis	28
5.5.2 Výhody	28
5.5.3 Nevýhody	28
5.5.4 Možnost aplikace na náš problém	29
5.6 Metody aplikované na náš problém	29
6. Algoritmus	30
6.1 Teoretická rovina	30
6.2 Praktická rovina	31
6.2.1 Korekce osvětlení	31
6.2.2 Úprava obrázku	33
6.2.3 Vyhodnocení obrázku	33
6.2.3.1 Detekce hrany	34
6.2.3.2 Kontrola oblasti	34
6.2.3.3 Zaplavování oblasti	35
6.2.4 Nakládání s výsledkem	36
7. Aplikace pro nadřazené PC	37
7.1 Důvod	37
7.2 Možnosti aplikace	37
8. Zhodnocení výstupu zařízení	39
8.1 Uživatelská přívětivost	39
8.2 Robustnost	39
8.2.1 Souměrné osvětlení	39
8.2.1.1 Standardní stav	39
8.2.1.2 Přesvětlení	41

8.2.1.3 Podsvětlení	41
8.2.2 Nesouměrné osvětlení	41
8.2.2.1 Silná hrana mezi objekty	42
8.2.2.2 Silná hrana procházející objektem	42
8.2.2.3 Slabá hrana	43
9. Závěr	44
Zdroje	45
Příloha A	i

Seznam obrázků

Obr. 1, CMOS obrazový senzor MT9V032	11
Obr. 2, Vývojový kit STM32F4 – Discovery, převzato z [6]	13
Obr. 3, Využívaný LCD displej 8 × 2	14
Obr. 4, Propojovací plošný spoj (po řadě TOP, BOTTOM), převzato z [3]	15
Obr. 5, Příklad konvoluční masky	21
Obr. 6-1, Původní obrázek [9]	21
Obr. 6-2, Po aplikaci konvoluční masky	21
Obr. 7-1, Původní obraz	22
Obr. 7-2, Po aplikaci prahování	22
Obr. 8, Prewittové operátory (po řadě Horizontální a Vertikální)	25
Obr. 9-1, Původní obraz	25
Obr. 9-2, po aplikaci horizontálního operátoru	25
Obr. 9-3, Po aplikaci vertikálního operátoru	25
Obr. 9-4, po sečtení abs. Hodnot obrazů 9-2 a 9-3	25
Obr. 10-1, Původní obraz	27
Obr. 10-2, Po aplikaci Prewittové operátorů	27
Obr. 10-3, Po aplikaci konvoluční masky z obr. 10-5	27
Obr. 10-4, Po aplikaci Prewittové operátorů na obr. 10-3	27
Obr. 10-5, Konvoluční maska	27
Obr. 11, Graf běhu algoritmu	30
Obr. 12, Graf průběhu doby expozice za měnících světelných podmínek	32
Obr. 13, Aplikace pro nadřazené PC	37
Obr. 14-1, Původní obraz, dotyk kostek	40
Obr. 14-2, Po thresholdu, dotyk kostek	40
Obr. 15-1, Původní obraz, silné zastínění	41
Obr. 15-2, Po thresholdu, silné zastínění	41
Obr. 16-1, Původní obraz, stín v objektu	42
Obr. 16-2, Po thresholdu, stín v objektu	42
Obr. 17-1, Původní obraz, slabé zastínění	43

Obr. 17-2, Po thresholdu, slabé zastínění	43
Obr. 18, Výsledné zařízení	i
Obr.19, Senzor zapojen na kitu	i

Seznam tabulek

Tab. 1, Parametry MT9V032, převzato z data sheetu senzoru [2]	12
Tab. 2, Základní parametry STM32F407VG, zdroj [4]	13
Tab. 3, Vyvedení pinů propojovacím plošným spojem, převzato z [3]	16
Tab. 4, Události znázorněné v grafu na obr. 12	32

1. Úvod

Díky rostoucímu výkonu a snižování cen mikrořadičů se jejich působení rozšiřuje téměř do všech odvětví. Jedním z těchto odvětví je i zpracování obrazu. To přichází s podporou od výrobců typu STmicroelectronics, kteří na své kity připojují i periferie, jež jsou primárně využívány pro přenos obrazu (například DCMI). Vzhledem k rychle rostoucí výpočetní kapacitě mikrořadičů je nyní možné za jejich využití zpracovávat obraz.

Cílem bakalářské práce je tvorba zařízení, které bude snímat obraz a následně tento obraz za využití mikrořadiče vyhodnocovat. Vyhodnocení bude zaměřeno na detekci uzavřených objektů.

Jelikož bude významnou součástí tohoto zařízení i interakce s uživatelem, bude muset zpracování obrazu probíhat s adekvátní rychlostí. Program bude muset rovněž brát v potaz proměnné osvětlení pro možnost aplikace v reálných podmínkách.

Důležitou podmínkou je též obecnost řešení, které bude zajišťovat nezávislost na konkrétních podmínkách.

Výsledné demonstrační zařízení bude snímat kostky vržené pod senzor a bude vyhodnocovat součet na nich hozený, nezávisle na počtu kostek. Demonstrační zařízení bude sloužit jako reprezentační předmět pro Katedru měření Elektrotechnické fakulty ČVUT.

2. Rozbor

Zařízení bude složeno z mechanického uchopení, obrazového senzoru, mikrořadiče a LCD displeje pro interakci s uživatelem.

Pro komunikaci s jednotlivými částmi zařízení a pro zpracování výsledků byl zvolen discovery kit STM32F407VGT6 od společnosti STmicroelectronics. Ke snímání obrazu a jeho prvotním modifikacím bude sloužit CMOS obrazový senzor MT9V032 a pro interakci s uživatelem poslouží LCD displej, obsluhující 8×2 znaky s mikrořadičem S6A0069.

Obrazový senzor bude součástí plošného spoje, jehož návrh vytvořil Ing. Ondřej Pribula. K propojení plošného spoje senzoru s kitem bude poté využita propojovací deska, za jejímž návrhem stojí Ing. Vojtěch Dokoupil [3].

Ovládání a komunikace se senzorem bude zajištěna pomocí ovladače, jež vznikl jako součást diplomové práce Ing. Vojtěcha Dokoupila [10].

Program bude psán v jazyce C za využití knihoven HAL od STmicroelectronic, a to pro zajištění možnosti jednoduché přenositelnosti mezi kity od zmíněné společnosti.

Program se bude potýkat s relativně nízkou operační pamětí mikrořadiče, a proto bude muset probíhat nad samotným sejmutým obrazem bez možnosti uložení většího množství mezivýsledků. Rovněž bude program narážet i na nízkou výpočetní kapacitu (v porovnání s běžně využívanými přístroji pro zpracování obrazu), a jelikož bude významnou součástí zařízení též interakce s uživatelem, musí vyhodnocení probíhat adekvátní rychlostí. Tyto problémy budou řešeny volbou vhodných algoritmů a jejich modifikací.

Problémem pro vyhodnocovací algoritmus může být mimo jiné také proměnné osvětlení. To bude řešeno korekcí doby expozice a gainu, aby si zařízení před každým vyhodnocováním zajistilo podmínky co nejbližší ideálu. Dalším krokem ke zvýšení robustnosti bude rovněž autodetekce chybného, či potenciálně chybného vyhodnocení.

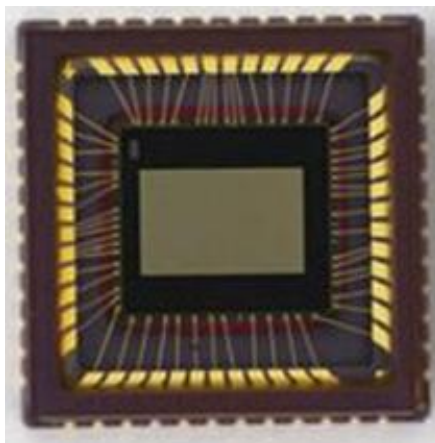
3. Využitý hardware

V této kapitole budou vyjmenovány významné hardwarové části zařízení a jejich stručný popis.

3.1 Obrazový senzor

3.1.1 Úvodní popis

Jako obrazový senzor bude sloužit MT9V032 (obr. 1). Jedná se o CMOS obrazový senzor, detekující na černobílé škále v rozmezí 0 – 255. Podporuje řadu možností pro úpravu snímaného obrázku (kupříkladu zaměření snímané oblasti, modifikaci rozlišení či korekci jasu) během samotného snímání konfigurací klíčových registrů.



Obr. 1, CMOS obrazový senzor MT9V032

3.1.2 Parametry

Parameter	Value
Optical Format	1/3-inch
Active Imager Size	4.51 mm (H) × 2.88 mm (V) 5.35 mm diagonal
Active Pixels	752H × 480 V
Pixel Size	6.0 μm × 6.0 μm
Color Filter Array	Monochrome or color RGB Bayer Pattern
Shutter Type	Global Shutter
Maximum Data Rate Master Clock	26.6 MPS/26.6 MHz
Full Resolution	752 × 480
Frame Rate	60 fps (at full resolution)
ADC Resolution	10-bit column-parallel
Responsivity	4.8 V/lux-sec (550 nm)
Dynamic Range	>55 dB; >80 dB-100dB in HDR mode
Supply Voltage	3.3 V ± 0.3 V (all supplies)
Power Consumption	<320 mW at maximum data rate; 100 μW standby Power
Operating Temperature	-30°C to + 70°C
Packaging	48-Pin CLCC

<- Monochrome

Tab. 1, Parametry MT9V032, převzato z data sheetu senzoru [2]

3.1.3 Důvod výběru

Hlavním důvodem výběru tohoto senzoru byla dostupnost navazujícího hardwaru k tomuto snímači. Plošný spoj pro daný senzor byl dostupný přímo na katedře a stejně tak tomu bylo s propojovacím tištěným spojem mezi modulem senzoru a využívaným kitem.

Senzor však rovněž podporuje úpravu obrazu během snímání, což je velká výhoda pro aplikace s nízkou výpočetní a paměťovou kapacitou.

3.1.4 Zapojení

Senzor je umístěn na plošném spoji, vytvořeném Ing. Ondřejem Pribulou. A celý modul je poté spojen s kitem pomocí propojovacího tištěného spoje, jež je popsán níže.

3.2 Mikrořadič

3.2.1 Úvodní popis

Výběr mikrořadiče byl spojen s otázkou, zda využít samotný mikrořadič, nebo využít vývojový kit. Nakonec bylo rozhodnuto zvolit vývojový kit z rodiny STM32F4 – Discovery s mikrořadičem STM32F407VG.



Obr. 2, Vývojový kit STM32F4 – Discovery, převzato z [6]

3.2.2 Parametry

Parametry mikrořadiče jsou dostupné v data sheetu, a zde budou uvedeny pouze základní parametry (tab. 2).

Základní parametry	
Kit	STM32F407VGT6
Jádro	32-bit ARM Cortex-M4F core
RAM	192 kB
Hodiny	Až 168 MHz
Flash	1 MB

Tab. 2, Základní parametry STM32F407VG, zdroj [4]

3.2.3 Důvod výběru

Důvodů pro výběr bylo několik. Hlavním důvodem je podpora DCMI pro komunikaci se senzorem, dále podpora USB komunikace (včetně portu) pro zasílání dat do nadřazeného počítače. Další výhodou tohoto vývojového kitu je i on-board DA převodník CS43L22, který ve spolupráci s portem pro jack 3,5mm měl zajišťovat audio výstup.

Ostatní v zařízení využívané periferie (I2C, DMA, atd.) jsou na kitech od STM vesměs běžné.

3.2.4 Nevýhody

Velkým nedostatkem je však operační paměť. Velikost 192 kB je pro paměťově náročné algoritmy (jako jsou právě algoritmy pro zpracování obrazu) značně omezující, a proto musela být brána v potaz velikost každé proměnné.

3.3 Displej

3.3.1 Úvodní popis

V zařízení je rovněž využíván displej pro interakci s uživatelem. Je jím displej 8 × 2 znaků s obecným řadičem S6A0069. Jeho parametry jsou dostupné v data sheetu [8]. Tento řadič odpovídá standardnímu HD44780 řadiči.



Obr. 3, Využívaný LCD displej 8 × 2

3.3.2 HD44780 standard

HD44780 je pravděpodobně nejznámějším řadičem. Jeho hojně rozšíření zajistila především jeho obecnost. Obecností je myšlena nízká závislost na konkrétních parametrech, jakou je napájecí napětí, počet komunikačních vodičů, velikost ovládaného displeje, či počet pixelů na jeden znak. Jeho bližší popis je dostupný z jeho datasheetu [7].

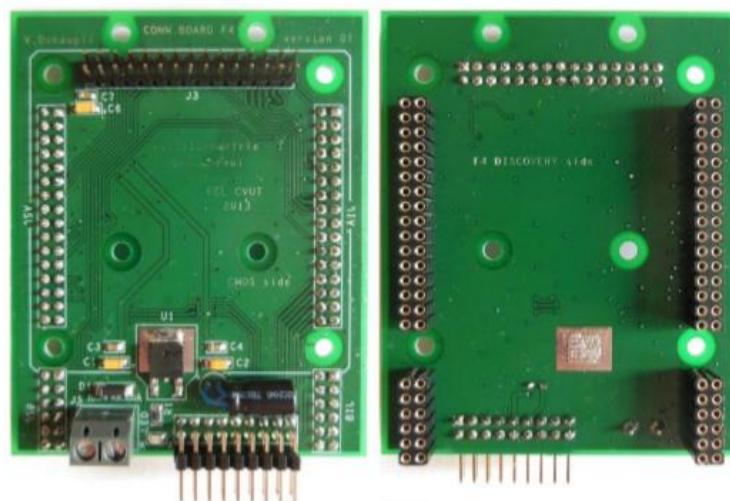
3.3.3 Důvod výběru

Displej byl vybrán především z důvodu jeho rozměrů, které vizuálně nejvíce seděly k výslednému zařízení. Jediným funkčním požadavkem na displej bylo ovládání odpovídající standardu HD44780, a to z důvodu možnosti jednoduchého přepojení na displej jiný.

3.4 Zapojení

K propojení discovery kitu s modulem senzoru byl využit plošný spoj, vytvořený Ing. Vojtěchem Dokoupilem v rámci jeho bakalářské práce [3].

Tento spoj zajišťuje vyvedení pinů z kitu do modulu senzoru, vyvedení pinů pro komunikaci s LCD displejem a zároveň i poskytuje možnost externího napájení.



Obr. 4, Propojovací plošný spoj (po řadě TOP, BOTTOM), převzato z [3]

3.4.1 Seznam vyvedených pinů

Zapojené piny a jejich využití je znázorněno v tab. 3.

P1		P2		DISCOVERY SILKSCREEN	function CMOS	other connector
GND	GND	GND	GND	PC1	DCMI_LEDOUT	
		SV	SV	PC2	DCMI_OE	
GND				PA1	DCMI_EXPOSURE	
PC1				PA3	DCMI_STANDBY	
				PA2	DCMI_RESET	
PA1				PA4	DCMI_HSYNC	
PA3	PA2			PA6	DCMI_PIXCLK	
	PA4			PB0	DCMI_SYSCLK	
	PA6			PE7		LCD_D4
	PB0			PE8		LCD_D5
GND				PE9		LCD_D6
PE7	PE8			PE10		LCD_D7
PE9	PE10			PE11		LCD_RS
PE11	PE12			PE12		LCD_E
PE13	PE14			PE13		LCD_RW
				PE14		LCD_BACKGLIGHT
				PD8		USART3_TX
				PD9		USART3_RX
				PE6	DCMI_D7	
				PE4	DCMI_D4	
				PE5	DCMI_D6	
				PB8	I2C_SCL	
				PB9	I2C_SDA	
				PB6	DCMI_D5	
				PB7	DCMI_VSYNC	
				PB4		SPI3_MISO
				PB5		SPI3_MOSI
				PB3		SPI3_SCK
				PC8	DCMI_D2	
				PC9	DCMI_D3	
				PC6	DCMI_D0	
				PC7	DCMI_D1	

Tab. 3, Vyvedení pinů propojovacím plošným spojem, převzato z [3]

4. Vývojová prostředí

4.1 Matlab

Vývojové prostředí Matlab bylo vyvinuto společností MathWorks. Jeho hlavní výhodou je široká podpora práce s maticemi, a díky široké škále knihoven je zároveň uživatelsky velice přívětivé.

Pro prvotní zkoušku algoritmů se tedy jednoznačnějevilo jako ta nejlepší volba.

4.1.1 Využití

Prostředí bylo využito pro tvorbu, modifikaci a testování algoritmů. Díky funkci `imshow()` a celkově velice jednoduché práci nad obrázky a maticemi bylo z možných řešení jednoznačně nejvýhodnější. Tyto výhody umožnily testovat algoritmy na volitelně zašuměných obrázcích a poté snadno vyhodnocovat jak mezivýsledky, tak i samotný konečný výsledek.

4.2 Keil uV5

Vývojové prostředí Keil pochází od stejnojmenné společnosti. Konkrétní verze Micro Vision 5 je verzí nejnovější. Toto prostředí má z volně dostupných vývojových prostředí nejlepší debugger, a to díky real-time sledování periférií atp. Významnou nevýhodou tohoto prostředí je ovšem omezení na velikost programu při využívání trial verze, a to na 32kB. Toto je řešitelné využitím optimalizace kompilátoru na velikost, ovšem to přináší výrazné zhoršení samotného debugování, neboť poté často nejsou mezivýsledky zobrazovány.

4.2.1 Využití

Prostředí bylo využito pro samotné programování kitu STM32F407VG. Z důvodu výše zmíněných výhod bylo primárním využitým prostředím.

4.3 AC6

AC6 je vývojové prostředí na bázi Eclipse. Bylo vyvinuto společností STmicroelectronics pro programování kitů od téže společnosti.

Jeho výhodou je, že se jedná o freeware a není tedy omezeno na velikost kódu. Majoritní nevýhodou jsou však problémy s kompilací. Gcc kompilátor má občas problémy s kódy vygenerovanými přes Cube, které využívají knihovny HAL. Tento problém propojení jednotlivých kroků programování kitů od STM v doporučených vývojových prostředích se nepodařilo vyřešit a proto bylo od programování v AC6 ustoupeno ve prospěch Keil uV5.

4.3.1 Využití

Prostředí bylo prvotně využíváno jako primární prostředí pro programování vývojového kitu. Po detekci výše zmíněných problémů bylo ovšem od dalšího programování v tomto vývojovém prostředí upuštěno.

4.4 Cube

Cube je prostředí od společnosti STmicroelectronics, jehož hlavním úkolem je vygenerovat prvotní kód pro inicializaci periférií na mikrořadičích zmíněné společnosti.

K inicializaci periférií Cube využívá knihovny HAL (více o knihovnách níže) od téže společnosti.

Jedná se tedy významné zjednodušení prvního kroku a z tohoto důvodu byl Cube při tvorbě našeho zařízení využit.

4.4.1 Využití

Cube byl využit pro korektní inicializaci periférií a pro nahrání využívaných knihoven HAL.

4.4.2 HAL knihovny

Knihovny HAL (High abstraction layer) byly vytvořeny společností STmicroelectronics jako nástupce SPL (standard peripheral library), jež už nejsou u nově vyvíjených kitů nadále podporovány. Cílem těchto knihoven bylo dosáhnout takové míry abstrakce, aby nebyl program závislý na konkrétním mikrořadiči. To se do značné míry i podařilo a byť je po přechodu na jiný mikrořadič nutná kontrola zdrojového kódu, z větší části zůstane nepozměněn.

4.5 QT

QT je vývojové prostředí od stejnojmenné společnosti, distribuované jako freeware. Jeho hlavní výhodou je vysoká míra abstrakce při programování GUI aplikací. Je tomu tak díky silné základně knihoven.

4.5.1 Využití

Toto prostředí bylo využito pro vytvoření počítačové aplikace pro komunikaci s vyvíjeným zařízením. Díky jeho zaměření na GUI aplikace bylo ideální volbou.

Zároveň je z důvodu podpory programování v C++ i dostatečně otevřené pro volitelnou komunikaci přes USB, jež je v zařízení využita.

5. Metody zpracování obrazu

Zpracování obrazu lze provést mnoha způsoby. Jelikož se jedná o klíčovou část této práce, budou zde zmíněny ty nejvýznamější a zároveň bude stručně poreferována jejich možnost aplikace v tomto projektu při omezených zdrojích, které mikrořadič STM32F407VG nabízí.

5.1 Princip konvoluce

5.1.1 Popis

V prvé řadě zde vysvětlím princip konvoluce a konvolučních masek, které jsou u dalších metod často skloňovány. Jde o vyhodnocení daného pixelu v závislosti na jeho okolí. Konvoluční masky (operátory) se mohou v libovolně lišit, avšak princip zůstává stejný.

5.1.2 Příklad

Za použití konvoluční masky z obr.5 na původní obrázek (obr. 6-1) získáváme jako výsledek jeho částečně rozmazanou modifikaci (obr. 6-2). Vyhodnocovaný pixel je standardně ve středu a jeho hodnota je násobena osmi. Hodnota pixelu o řádek výš je násobena čtyřmi a tento násobek je přičten k předchozímu číslu. Ostatní pixely jsou vyhodnoceny analogicky. Výsledný součet je poté vydělen sumou masky (v našem případě se jedná o 32) a tato hodnota je uložena na souřadnice středového pixelu v nově vznikajícím obrázku.

Nový obrázek vzniká proto, aby byla maska vždy aplikována na originální obrázek. Samozřejmě je také možné přepisovat přímo vyhodnocovaný obrázek a průběžně si ukládat předchozí hodnoty stranou pro optimalizaci paměťové náročnosti, ale pro příklad je lepší vytvářet obrázek zcela nový.

2 4 2
4 8 4
2 4 2

Obr. 5, Příklad konvoluční masky



Obr. 6-1, Původní obrázek [9]



Obr. 6-2, Po aplikaci konvoluční masky

5.2 Prahování (thresholding)

5.2.1 Popis

Nezákladnější metodou segmentace obrazu je metoda prahování (thresholding). Ta je založena na ohodnocení každého pixelu v závislosti na thresholdu (prahu). V nejjednodušší formě se často jedná o rozdělení pixelů na foreground-background, tedy popředí-pozadí. Cílem je tedy rozdělit oblasti obrázku podle jejich hodnoty (světlosti) na dvě skupiny. To se dá poté využít například k lokalizaci objektů, určení jejich velikostí či pozic. Tato metoda má samozřejmě i formy složitější, ať už jde o dynamický práh (například závislý na pozici pixelu, popřípadě na průměrné světlosti obrázku), či o více prahů a tím dělení do více než dvou skupin.

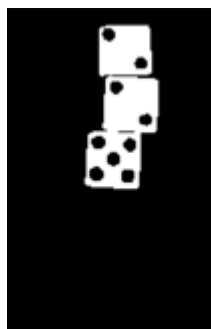
5.2.2 Příklad

Ukázkou výstupu po zavolání thresholdovací metody na obr.7-1 je obr.7-2.

Jako práh byla zvolena hodnota 90. Tedy všechny pixely se světlostí nižší než je práh jsou zobrazeny černě, s vyšší nebo rovnou pak bíle.



Obr. 7-1, Původní obraz



Obr. 7-2, Po aplikaci prahování

5.2.3 Výhody

Hlavní výhodou metody je její jednoduchost. A to nejen z pohledu implementačního, ale i z pohledu práce procesoru. Protože vyhodnocovaný pixel není závislý na jeho okolí, je možné provádět toto vyhodnocení již v průběhu snímání obrázku. To zajišťuje možnost vyhodnocovat obraz v reálném čase. Další výhodou, vyplývající z jednoduchosti, je i naprostá předvídatelnost algoritmu a jeho jednoduchý debugging.

5.2.4 Nevýhody

Jednoduchost algoritmu má ovšem i svou cenu. Tou je v první řadě nespolehlivost či v lepším případě alespoň částečná nepřesnost v neoptimálních podmínkách. V případě, že je například obraz částečně zastíněn, může mít nedynamický práh velké problémy, kdy objekty, jež měly být označeny jako foreground, spadnou do backgroundu. Stejně tak se děje v případě různých odlesků, kde se naopak background jeví jako foreground.

Tyto nevýhody jsou však částečně řešitelné. Částečně zastínění lze řešit kupříkladu již zmiňovaným dynamickým prahem, kde na základě

souřadnic pixelu se program podívá na referenční body a z nich určí, jaký by měl být threshold na dané pozici. To už ovšem nese zase další nevýhodu, a tou je závislost na referenčních bodech. Nevýhodu odlesků pak částečně řeší rozmazání obrázku, ať už pomocí konvoluční masky, binningu, nebo jiným způsobem. To opět přináší nevýhodu, kterou je zvýšená náročnost a snížení rozlišení obrázku, což může způsobit nesprávné ohraničení objektů a tím změnu jejich rozměrů, popřípadě propojení dvou blízkých objektů.

5.2.5 Možnost aplikace na náš problém

Přes zmíněné nevýhody je tato metoda v zařízení využita. Tyto nevýhody jsou částečně potírány. Problém s odlesky je řešen snížením rozlišení obrazu, které podporuje přímo zvolený senzor a tedy probíhá bez jakékoliv zátěže pro procesor.

Problém se stínem pak pokrývá jak korigovaná doba expozice, tak dynamický práh, závislý na N nejsvětlejších a N nejtmavších pixelech v obrázku. Touto metodou se dostáváme nejlépe k ideálu bez závislosti na referenčních bodech, které se jeví jako přílišná újma na obecnosti řešení.

5.3 Nálepkování (labeling)

5.3.1 Popis

Metoda labelingu je velice široký pojem. Cílem metody je olabelovat (onálepkovat) jednotlivé oblasti na základě jejich shodných vlastností. Jelikož existuje nepřehledné množství klíčů, podle kterých se jednotlivé nálepky rozdělují, zmíníme zde pouze přímo v našem zařízení využívané a jejich konkrétní využití bude rozvedeno v bodě **Využívané metody**.

Za nejjednodušší formu labelingu lze označit i metodu prahování, kde jsou klíčem hodnoty prahů. S těmi jsou porovnávány jednotlivé pixely a na základě tohoto porovnání je jim udělena odpovídající nálepka.

Další v zařízení využitou možností je udělení klíče na základě okolních pixelů, čímž lze definovat například podmínku „dostaneš svou nálepku L, pokud susedíš s pixelem oblasti A, popřípadě pokud se dokážeš přes přímé sousedství dostat přes pixely označené L až k oblasti A, a zároveň momentální oblast L není příliš velká a nikdo z L nesousedí s nikým jiným než s A nebo L“.

Podmínky jsou tedy, jak je vidno, neomezené, a tato konkrétní podmínka je v algoritmu přímo využita, byť ještě drobně rozšířena.

5.3.2 Výhody

Výhodou této metody je její naprostá otevřenost. Možností zadefinovat si podmínku podle potřeb konkrétní aplikace získává na atraktivitě. Díky tomu se zároveň při správnem zadefinování podmínek stává přesně únosně náročnou a poměr cena-výkon je zcela v rukou uživatele.

5.3.3 Nevýhody

Ne vždy však musí tato metoda stačit. Jejím výstupem je pouze rozdělení na jednotlivé oblasti, což nemusí být žádaným výstupem. Další nevýhodou je, že při složitějších podmínkách může dojít k nedostatku nálepek, což může být neřešitelný problém – v závislosti na konkrétní aplikaci například můžeme narazit na to, že nesmíme nechat labely přetéct a zároveň si nemůžeme dovolit mít nálepkou větší než 8bitovou (což se děje přímo v našem zařízení).

5.3.4 Možnost aplikace na náš problém

Jak již bylo zmiňováno v předchozích bodech, je tato metoda v našem zařízení využívána. Je tomu tak právě kvůli její otevřenosti, která nás nechala si konkrétně zadefinovat naše specifické podmínky.

Pravděpodobnost možného problému s přetečením nálepek je značně snížena předchozím zpracováním obrazu. V případě, že při přechodných jevech přesto nastane (například při zachycení vrhané kostky může být obraz značně zmatený), je problém detekován a ošetřen.

5.4 Hranová detekce (edge detection)

5.4.1 Popis

Poněkud pokročilejšími metodami jsou metody založené na detekci hran, tedy analýzy gradientu úrovně šedi (světlosti). Hrana samotná se nejjednodušším způsobem detekuje pomocí konvolučních masek.

Oblíbeným operátorem je například Prewittové operátor (obr. 8), jak zmiňuje například [1]. Cílem je vytvořit obraz, složený z hodnot růstu (poklesu) úrovně šedi ve směru derivace. Hrany kolmé na směr jsou vždy pomíjeny, proto je často žádoucí provádět detekci více směry.

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

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

Obr. 8, Prewittové operátory (po řadě Horizontální a Vertikální)

5.4.2 Příklad

Jako příklad nám poslouží aplikace horizontálního (výstup v obr. 9-3) a vertikálního (obr. 9-2) Prewittového operátoru na obr. 9-1. Sečtením jejich absolutních hodnot poté získáváme obr. 9-4, který obsahuje již všesměrovou detekci.



Obr. 9-1, Původní obraz



Obr. 9-2, po aplikaci horizontálního operátoru



Obr. 9-3, Po aplikaci vertikálního operátoru



Obr. 9-4, po sečtení abs. Hodnot obrazů 9-2 a 9-3

5.4.3 Výhody

Samotná detekce je relativně nenáročná operace. S předchozími zmiňovanými metodami se ovšem nedá srovnávat, neboť vyžaduje buď častý přepis „odkládacího“ pole (výpočetní náročnost), popřípadě vytvoření zcela nového výsledného obrázku (paměťová náročnost).

Jejím primárním účelem ovšem je, že se jedná o první krok pro pokročilejší algoritmy závislé na detekci hrany.

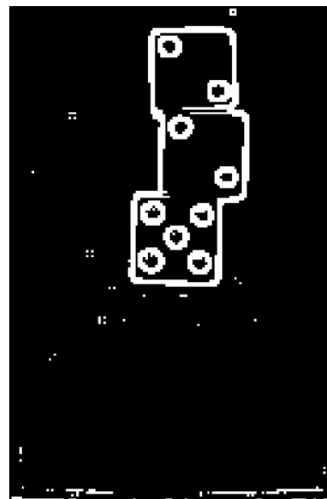
5.4.4 Nevýhody

Nevýhodou je nedostatečný výstup. Po surové detekci hran je velice pravděpodobné, že z výstupu nedokážeme obrázek uspokojivě vyhodnotit. Proto na tuto metodu často navazují metody další.

Další nevýhodou je výskyt nechtěných hran v místech odlesků, zastínění a podobně (obr. 10-2). Toto je řešitelné rozmazáním obrázku – ať už konvoluční maskou (obr. 10-5), popřípadě v předchozí metodě zmiňovaným binningem. To ovšem zároveň výrazně rozšiřuje reálné hrany, a slévá blízké objekty v jeden (obr. 10-4).



Obr. 10-1, Původní obraz



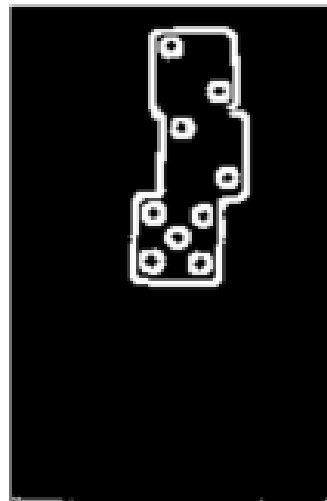
Obr. 10-2, Po aplikaci Prewittové operátorů

```
1 2 3 2 1
2 5 8 5 2
3 8 16 8 3
2 5 8 5 2
1 2 3 2 1
```

Obr. 10-5,
Konvoluční maska



Obr. 10-3, Po aplikaci
konvoluční masky z obr. 10-5



Obr. 10-4, Po aplikaci Prewittové
operátorů na obr. 10-3.

*(Z důvodu grafického znázornění byla po detekci hran ještě zavolána
thresholdovací metoda s prahem 30)*

5.4.5 Možnost aplikace na náš problém

Aplikace detekce hran v našem zařízení byla vážně zvažována a testována. Bohužel však její výstup se jevil jako nedostatečný, především kvůli možnosti neuzavření celé hrany a kvůli slévání blízkých objektů do jednoho. Požadavky na následné zpracování získaných dat už byly pro využívaný mikrořadič nereálné.

5.5 Cannyho hranový detektor

5.5.1 Popis

Tato metoda je široce rozšířena pro její vysokou přesnost. V první řadě rozmazává obrázek, což lze udělat již mnohokrát zmiňovanými metodami. Poté následuje detekce hran obrázku, jež je popsána výše.

Jak je zmíněno v bodě **Detekce hran**, je z důvodu rozmazání výstupem obrázek s neúměrně širokými hranami. Tyto hrany je poté nutné vyhodnotit.

Prvním krokem v následném vyhodnocení hran bývá nejčastěji jejich ztenčení. To lze udělat mnoha způsoby, například uznáním pouze nejsilnějšího gradientu (nejvyšší hodnoty). Jako hrana je uznán pouze pixel, v jehož okolí jsou gradienty nominálně nižší, než má on sám.

Poté přichází poslední krok metody, kterým je uznání hrany. To je nejčastěji děláno pomocí prahování. Obvykle jsou zadefinovány dva prahy T_1 a T_2 ($T_1 \geq T_2$). Pixel překračující práh T_1 je označen za silnou hrana. Pixel nepřekračující T_1 , avšak překračující T_2 je označen za slabou hrana, avšak pouze za předpokladu, že sousedí s hranou silnou.

5.5.2 Výhody

Hlavní výhodou je vysoká spolehlivost detekce a zěštíhlenost hran, což může být pro mnohé navazující algoritmy (například algoritmus pro doplňování chybějících hran) výrazné zrychlení, často to bývá dokonce i požadavek.

5.5.3 Nevýhody

Hlavní nevýhoda algoritmu je ovšem naprosto zřejmá. A tou je jeho náročnost. Pracuje s několikanásobným průchodem obrázkem a jeho výstup navíc stále není řešením, ale jeho výstup ještě potřebuje další netriviální zpracování.

5.5.4 Možnost aplikace na náš problém

Jak již bylo zmíněno v bodě **Nevýhody**, je tento algoritmus pro využití za omezených kapacit našeho mikrořadiče nevýhodný.

5.6 Metody aplikované na náš problém

Ne všechny metody jsou vhodné pro využití za nízkých kapacit, které kit STM32F407VG nabízí. V tomto bodě budou blíže rozebrány metody, které byly skutečně využity.

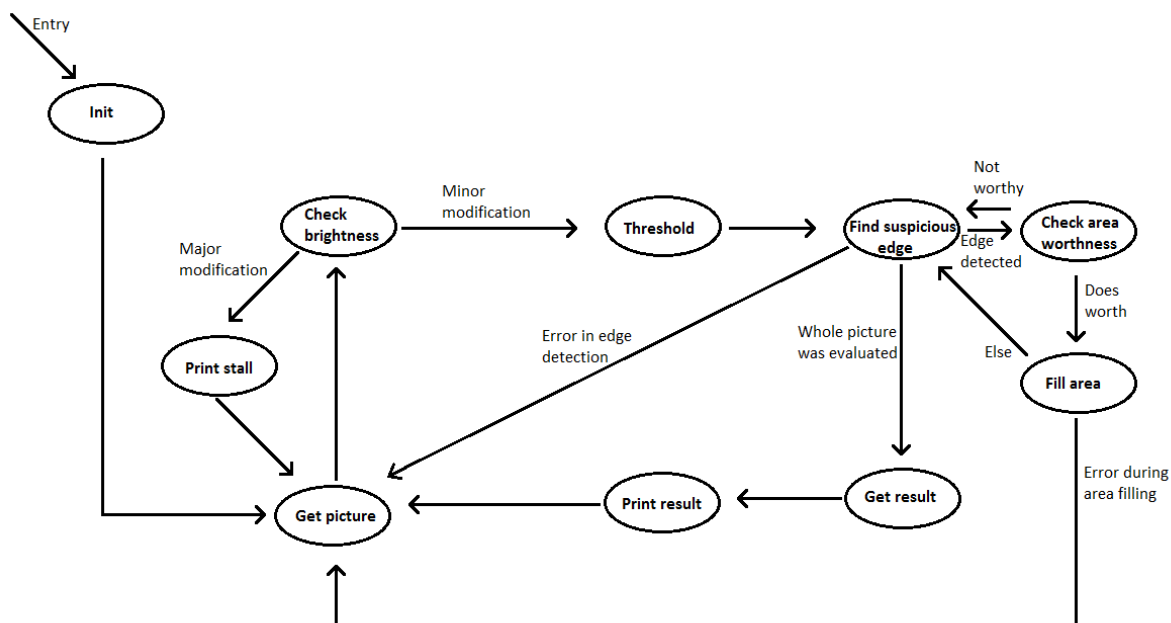
Celý program provází metoda labelingu. Ta je díky výše uvedeným výhodám ideální pro aplikaci na využívaném kitu. S první formou labelingu se setkáváme v jeho nejjednodušší formě, ve formě thresholdingu. Zde rozdělujeme obrázek na dvě separátní nálepky. Na černou (0) a bílou (255). Thresholdovací hodnota je v našem případě dynamická a je závislá na N nejsvětlejších a N nejtmavších pixelech.

Druhé využití labelingu je poté s o poznání složitějším klíčem. Jeho cílem už je samotná detekce uzavřených objektů. Jednotlivými podmínkami klíče jsou:

- 1) buď černý a soused s bílou, nebo soused s pixelem se stejnou nálepkou, jakou máš dostat ty (tzn. dokážeš se přes svou nálepku dostat až k bílé)
- 2) jediní tví sousedé mohou být bílá, černá a pixely se stejnou nálepkou, jakou máš dostat ty
- 3) ještě nebylo rozdáno maximální množství dané nálepky
- 4) pokud nesplníš některou z podmínek, všechny pixely s tvou nálepkou jsou považovány za pozadí

6. Algoritmus

6.1 Teoretická rovina



Obr. 11, Graf běhu algoritmu

Algoritmus sestává z více částečně provázaných kroků. Prvním krokem je korekce osvětlení využitím korekce doby expozice a úpravou gainu („Get picture – Check brightness – Print stall“ smyčka v obr. 11).

Ve druhém kroku je sejmout již správně osvětlený obrázek. Ten je ještě jednou zkontrolován na osvětlení a jsou uloženy klíčové parametry osvětlení obrázku („Get picture – Check brightness“ v obr. 11).

Poté následuje úprava obrázku pomocí metody prahování, která využívá v předchozím kroku získané parametry („Threshold“).

Na tuto úpravu navazuje metoda nálepkování, jejímž výstupem už je přímo číslo, kolik objektů bylo nalezeno („Find suspicious edges – Check area worthenss – Fill area“ smyčka v obr. 11).

Nakonec přichází využití výsledku, které už je zcela závislé na konkrétních požadavcích po daném zařízení, u nás je to výpis na LCD displej (Print result).

6.2 Praktická rovina

6.2.1 Korekce osvětlení

Ke korekci světelných podmínek slouží úprava doby expozice a gainu. To je prováděno zpětnovazebně využitím dělení intervalu (na obr. 12 je vidět konkrétní vzhled úpravy v závislosti na změnách osvětlení).

Je zadefinována ideální hodnota rozdílu nejsvětlejších a nejtmavších N pixelů v obrázku (číslo N je opět definováno a může se v závislosti na aplikaci lišit). V obrázku je nalezeno těchto N hodnot a ty jsou vyprůměrovány. Po odečtení průměrného nejsvětlejšího a průměrného nejtmavšího pixelu získáváme momentální rozdíl. Hodnota průměrného nejsvětlejšího a průměrného nejtmavšího pixelu je uložena stranou, neboť je využívána v dalších krocích.

Rozdíl je porovnán s definovanou ideální hodnotou, a je naordinována korekce, jež je závislá na vypočítaném rozdílu. Je spočítána jeho vynásobením s zadefinovaným multiplifikátorem. Je tomu tak z důvodu maximálního možného rozdílu 255 – doba expozice je tedy rozdělena na 255 diskrétních možností.

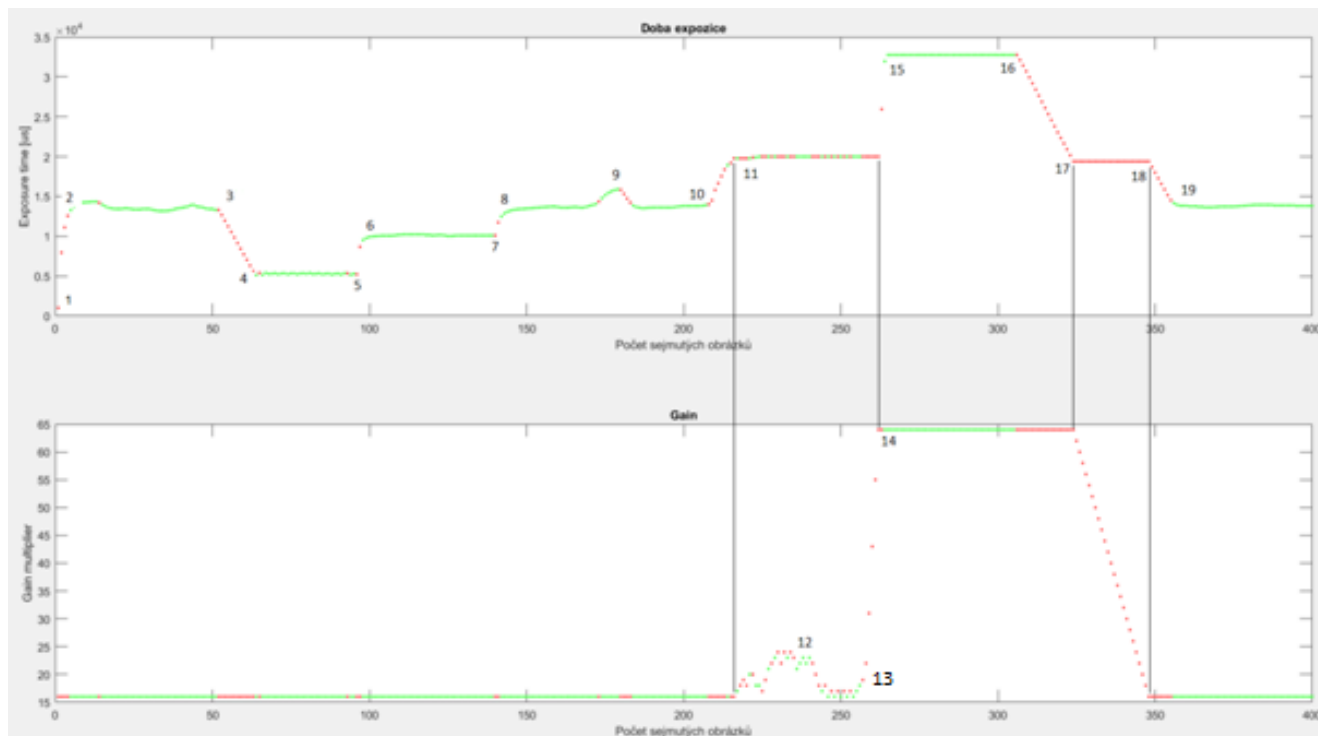
V případě, že je již doba expozice příliš vysoká a cílem korekce je další zesvětlení, je využita úprava gainu. Děje se tak z důvodu dynamického zpracování obrazu, které je při maximálních hodnotách doby expozice pro uživatele nepříjemně dlouhé.

Gain je podobně jako doba expozice rozdělen na omezené množství diskrétních hodnot. V případě gainu je toto množství 48.

Velikost korekce je poté vydělena třemi (dělení intervalu) a příslušné registry jsou o tuto hodnotu upraveny. Podle velikosti korekce se rozhoduje, zda se jednalo o velkou či o kosmetickou změnu. V případě velké je funkce pro korekci osvětlení volána znovu. Pokud se jednalo pouze o kosmetickou změnu, program pokračuje k dalšímu kroku.

Dále je ještě vyhodnocována jedna možnost, a tou je absence objektů pod senzorem. To je zjišťováno porovnáním průměrných hodnot nejsvětlejších a nejtmavších pixelů. Na to program reaguje delayem, nicméně i přesto osvětlení koriguje. Děje se tak z důvodu, že například při nárazové a výrazné změně osvětlení by si mohl program myslet, že pod senzorem nejsou

žádné objekty. V tomto extrémním případě se tak program nedostane do nežádoucího stavu bez možnosti návratu, pouze mu bude trvat návrat značně déle.



Obr. 12, Graf průběhu doby expozice za měnících světelných podmínek.

Nahoře Doba expozice, dole Gain. Zelené body znamenají spokojenost s nastavením, červené požadavek na další úpravu

1	Úprava světla od inicializační hodnoty na základě momentálních podmínek	11	Úprava doby expozice dosáhla maximální hodnoty pro rychlé zpracování obrazu, bude upravován gain
2	Ustálení doby expozice	12	Pohyb mezi senzorem a zdrojem osvětlení
3	Výrazné zvýšení osvětlení	13	Výrazné zastínění (senzor je překryt krabicí)
4	Ustálení doby expozice	14	Gain je na maximum, nastává úprava doby expozice
5	Částečné snížení osvětlení	15	Doba expozice i gain je na maximum, senzor už nedokáže zlepšovat podmínky
6	Ustálení doby expozice	16	Návrat k původnímu osvětlení (odsundání krabice)
7	Snížení osvětlení na původní hodnotu	17	Návrat k mezi rychlého zpracování, bude snižován gain
8	Ustálení doby expozice	18	Gain je na minimum, dále se bude snižovat doba expozice
9	Krátkodobé zastínění (pohyb mezi senzorem a zdrojem světla)	19	Ustálení doby expozice
10	Výrazné zastínění		

Tab. 4, Události znázorněné v grafu na obr. 12

6.2.2 Úprava obrázku

Po sejmutí korektně osvětleného obrázku přichází jeho úprava. Ta je dělána metodou prahování (thresholdingu).

Cílem je rozdělit pixely na dvě separátní skupiny, na čistě černou (0) a na čistě bílou (255). To je provedeno průchodem obrázku a porovnáním každého pixelu s thresholdovací hodnotou. Využívaná thresholdovací hodnota je dynamická a je závislá na průměrném nejsvětlejším a průměrném nejtmavším pixelu, získaném v předchozím kroku.

Dynamický threshold byl zvolen z důvodu zvýšení robustnosti, kdy i v extrémních podmínkách, ať už při přesvětlení či naopak podsvětlení, vyhodnocuje algoritmus pixely správně. Částečně tato volba thresholdu řeší i neúplné zastínění obrázku (například když je zastíněna pouze polovina). Efektivnějším řešením pro volbu thresholdu by byla funkce, závislá na souřadnicích pixelu, kde by tato funkce byla vytvořena na základě referenčních bodů, avšak s ohledem na obecnost řešení (resp. jeho nezávislost na referenčních bodech) byla tato možnost vyhodnocena jako neoptimální pro tuto úlohu.

Výstupem tohoto kroku je obrázek složený pouze ze dvou nálepek (0 a 255).

6.2.3 Vyhodnocení obrázku

Posledním krokem algoritmu je pak vyhodnocení předzpracovaných dat. Cílem je najít všechny černé oblasti, uzavřené v oblasti bílé. K tomu je využita metoda labelingu (nálepkování).

Návratovou hodnotou z této části už je číslo, jež označuje počet nalezených uzavřených objektů, splňujících parametry na velikost. Pokud však v průběhu vyhodnocování nastane chyba, vrátí funkce chybový výstup. Děje se tak v případě přetečení využívaných nálepek nebo přetečení zásobníku využívaného při zaplavování oblasti.

6.2.3.1 Detekce hrany

Program prochází po řádcích obrázků a sleduje přechod z bílé na černou. V případě, že na něj narazí, zkontroluje tuto oblast, zda stojí za zaplavování (bod **Kontrola oblasti**). V případě, že kontrola proběhne úspěšně, začne do hloubky zaplavovat danou oblast unikátní nálepkou (bod **Zaplavování**).

Zaplavování zároveň informuje o úspěchu/neúspěchu při hledání uzavřené oblasti a proto je v rámci této funkce rovněž inkrementován čítač nalezených vhodných oblastí.

6.2.3.2 Kontrola oblasti

Ke kontrole, zda oblast stojí za pokus o její zaplavení, dochází z důvodu algoritmické náročnosti samotného zaplavování oblasti, ke kterému se chceme dostávat ideálně pouze v případech, kdy narazíme na skutečný objekt. Rovněž je výhodné nerozdávat nálepky zadarmo, neboť je jich omezené množství, konkrétně 254 (0 a 255 jsou již zabrány).

Kontrola je provedena jednoduchým pohledem až o N pixelů ve všech směrech dál, zda narazí na bílou. Číslo N je zadefinováno a může se v závislosti na aplikaci a na očekávaných velikostech objektů lišit. Pohled směrem doleva není uskutečňován, neboť je tato funkce volána na pixel, o němž víme, že s ním nalevo přímo sousedí bílý pixel.

Za předpokladu, že ve všech směrech narazíme na bílou, pokračuje se k zaplavování oblasti labelem. V opačném případě je oblast brána jako nevhodná, nedochází k jejímu zaplavení a pokračuje se dál v průchodu (bod **Detekce hrany**).

6.2.3.3 Zaplavování oblasti

Samotné zaplavování oblasti probíhá do hloubky za využití zásobníku. Zásobník je staticky zaalokován a jako první prvek je mu přidělen výchozí pixel. Poté se vstupuje do cyklu.

Cyklus je tvořen těmito kroky:

- a. seber vrchní pixel ze zásobníku*
- b. zkontroluj, zda nejsi okrajový pixel (v případě, že se objekt dotýká rámu obrázku, nejedná s o uzavřený objekt a tedy není brán v potaz. Pokud se jedná o okrajový pixel, je cyklus ukončen s návratovou hodnotou informující o nekorektní oblasti)*
- c. zkontroluj, že jsi skutečně černý (může totiž nastat, že pixel byl přidán vícekrát a už byl vyhodnocen. Tímto se algoritmus vyhýbá nekonečné smyčce. Režie nad kontrolou nepřidání již přítomného prvku by byla příliš drahá)*
- d. zkontroluj své sousedy ve všech čtyřech směrech. V případě, že je pixel černý, přidej ho na zásobník. Pokud je bílý nebo má stejnou nálepku, jako máš dostat ty, nic nedělej. Pokud má cizí nálepku, ukonči cyklus s odpovídajícím výstupem (ve chvíli, kdy program narazí na cizí nálepku, je jisté, že tato předchozí nálepka neproběhla úspěšně a tedy ani oblast momentálně zaplavovaná nebude vyhodnocena jako korektní)*
- e. zkontroluj, zda jsi nepřesáhl maximální velikost oblasti (v opačném případě ukonči cyklus a vrať adekvátní výstup. Velikost oblasti je zadefinována a je závislá na konkrétní aplikaci)*
- f. označ se nálepkou*
- g. zkontroluj, zda zásobník není prázdný. Pokud není, pokračuj v cyklu. Pokud je, ukonči cyklus a pokroč k další fázi.*

Pokud cyklus úspěšně doběhne, je vyhodnocena velikost nalezené oblasti. V případě, že splňuje i požadavky na zadefinovanou minimální velikost, je návratovou hodnotou úspěch a počet nalezených objektů je inkrementován. Rovněž je možné (avšak v našem případě nevyužívané) vracet i velikost pole a souřadnice jeho počátečního pixelu.

6.2.4 Nakládání s výsledkem

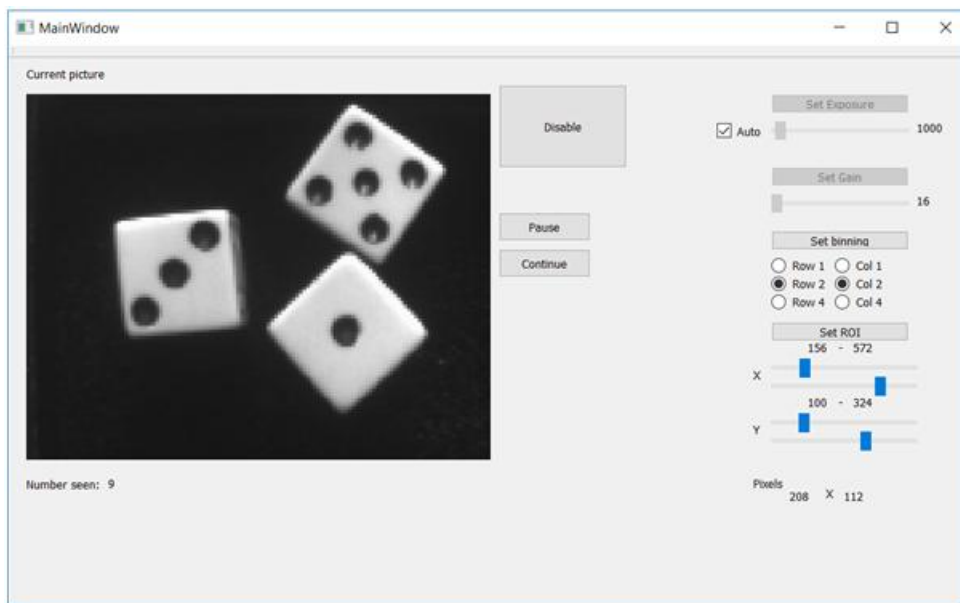
Výstup z bodu **Vyhodnocení obrázku** je zkontrolován, zda se nejedná o chybový výstup. V tom případě je chyba adekvátně ošetřena.

V opačném případě je výsledek porovnán s předchozím výsledkem. Pokud se dvě vyhodnocení v řadě shodují, je číslo vypsáno.

Dvojitá kontrola výsledku byla zvolena z důvodu zvýšení robustnosti a je zcela dostačující. Původně zamýšlená kontrola třech výsledků tudíž není potřeba. Hlavní silou tohoto opakovaného vyhodnocování je, že každé vyhodnocení bude s velice vysokou pravděpodobností probíhat za různého osvětlení.

Rozdílnost světlosti nastává, neboť přestože bylo v bodě **Korekce osvětlení** rozhodnuto obraz vyhodnocovat, přesto se doba expozice (popříadě gain) minoritně upravila. Rovněž se v daném časovém úseku, dělicím sejmutí těchto dvou obrázků, upraví i vnější světelné podmínky.

7. Aplikace pro nadřazené PC



Obr. 13, Aplikace pro nadřazené PC

V rámci bakalářské práce byl též vyvinut program pro nadřazené PC (obr. XX) propojené pomocí micro USB s kitem. Tato aplikace umožňuje uživateli přístup ke všem klíčovým parametrům pro snímání obrazu.

7.1 Důvod

Cílem je, aby v případě změněných požadavků na aplikaci uživatel nemusel přímo zasahovat do zdrojového kódu, ale aby bylo možné změnit jednotlivé registry dálkově v průběhu běhu programu.

7.2 Možnosti aplikace

Program v okně zobrazuje aktuálně snímáný obraz a nabízí možnost jeho zapauzování. Rovněž vypisuje aktuálně viděný počet objektů.

Nabízí též možnost úpravu snímání oblasti, tedy oblast, která bude ze senzoru brána jako validní a která bude vyhodnocována. Tato možnost může být využita jak k zaměření snímání oblasti v případě částečné deformace ramena zařízení.

V rámci programu je možná také úprava doby expozice a gainu, popřípadě je možné nastavit automatickou úpravu (která je popsána výše v kapitole **Algoritmus**) a je doporučeno využívat automatické nastavení. V určitých případech je však možné, že manuální nastavení bude efektivnější. Jedná se o případy za velice stabilního vnějšího osvětlení, kdy bude díky manuálnímu nastavení program rychlejší, ovšem nebude schopen reagovat na změněné podmínky. Rovněž v extrémních případech silného přesvětlení nebo podsvětlení (které senzor není schopen korigovat) bude program částečně zrychlen. Ani v jednom případě však nebude zrychlení výrazné.

V neposlední řadě poté PC aplikace nabízí možnost úpravy binningu. A to jak horizontálního, tak vertikálního. Například v případě zaměření se pouze na určitou oblast může být žádoucí využít plné rozlišení a proto je tato možnost aplikací nabízena.

8. Zhodnocení výstupu zařízení

8.1 Uživatelská přívětivost

Zařízení je z hlediska uživatele velice nenáročné. Jediným požadavkem na uživatele je jeho zapojení do napájení pomocí USB kabelu a poté hod kostkou.

Zařízení samotné má přednastavené všechny parametry pro standardní běh. V případě požadavku na změnu klíčových parametrů je implementován PC program, jehož prostřednictvím se přes druhé USB komunikuje s zařízením. Tento program intuitivně nabízí úpravu všech významných parametrů a je tedy pro jeho operátora opět velice nenáročný. Požaduje pouze znalost pojmů „Exposure time“, „Region of interest“ a podobně.

8.2 Robustnost

Ohodnocení robustnosti proběhne nad rozdílnými světelnými podmínkami. Významné stavy osvětlení budou popsány a bude ohodnocena práce zařízení při těchto podmínkách.

Standardní osvětlení, neboť se jedná o stav osvětlení, v němž by mělo zařízení pracovat, bude navíc rozšířeno o statistický vzorek pokusů

8.2.1 Souměrné osvětlení

Za souměrné osvětlené označujeme osvětlení, při němž v sejmutém obrazu není detekována žádná významná hrana stínu.

8.2.1.1 Standardní stav

Za standardní stav osvětlení označujeme takový, který je označitelný za souměrný a zároveň je senzor schopen si úpravou doby expozice a úpravou gainu zajistit podmínky žádané pro běh algoritmu.

V tomto případě bylo na staticky hodnotných 500 hodech zařízení testováno. Ve všech pěti stech náhodných hodech zařízení

správně vyhodnotilo součet a dá se tedy předpokládat, že úspěšnost se bude blížit 100%.

Rovněž byly v rámci testování zkoušeny potenciálně problémové případy, jakými je kostka v rohu snímané oblasti, různá vzájemná poloha kostek a podobně.

Jako jediný problémový stav se ukázal takový, kdy kostky mezi sebou vzájemně dokonale uzavírají oblast pozadí (obr. 14-1 a -2), kde je daná uzavřená oblast rovněž vyhodnocena jako validní tečka.



Obr. 14-1, Původní obraz,
dotyk kostek



Obr. 14-2, Po thresholdu,
dotyk kostek

Pravděpodobnost tohoto jevu je však velice nízká (energie vložená do hodů kostkami zajistí, že při jejich sražení dojde k odrazu a tak nenastane přímý dotyk, obzvláště pak když je tento dotyk požadován nejméně ve třech bodech a navíc s podmínkami na minimální a maximální vzdálenost těchto bodů).

S ohledem na pravděpodobnost jevu tedy zařízení tuto možnost nekontroluje, neboť zvýšení robustnosti by bylo ve velice nízkých řádech, zatímco algoritmická náročnost tohoto ošetření by byla znatelná.

8.2.1.2 Přesvětlení

Za přesvětlení jsou označovány takové podmínky, při kterých není senzor schopen ani snížením doby expozice a gainu na minimum zabránit přílišnému jasů nejsvětlejších pixelů.

Za těchto podmínek je zařízení schopno pracovat s vysokou úspěšností. Narušení úspěšnosti nastává pouze při odrazu světla, kdy se černý objekt vliem odrazu tváří jako bílý, popřípadě je vlivem odrazu velikost černého objektu neúnosně malá nebo je odrazem rozdělen ve více menších.

8.2.1.3 Podsvětlení

Za podsvětlení je označován takový stav, kdy je doba expozice na maximu, gain taktéž a přesto nemá snímaný obraz požadovaný jas.

V tomto případě nastávají dva problémy. Prvním je doba zachycení obrazu, která je společně se zpracováním uživatelsky nepříjemně vysoká. Druhým problémem je pak samotná robustnost zpracování obrazu, kdy je zařízení výrazně náchylnější na stabilitu osvětlení.

8.2.2 Nesouměrné osvětlení

Za nesouměrné osvětlení je považováno takové, kde sejmutým obrazem prochází významná hrana stínu.

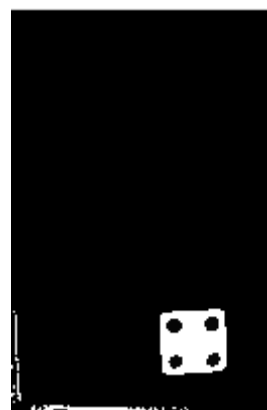
V tomto případě jsou v závislosti na lokaci dané hrany a na rozdílu osvětlení možné tři výstupy.

8.2.2.1 Silná hrana mezi objekty

Prvním je, hrana prochází někde v prostoru mezi kostkami a rozdíl je takový, že druhá kostka je zcela nedetekována (obr. 15-1 a obr. 15-2). V tomto případě dochází k vyhodnocení pouze osvětlené kostky.



Obr. 15-1, Původní obraz,
silné zastínění



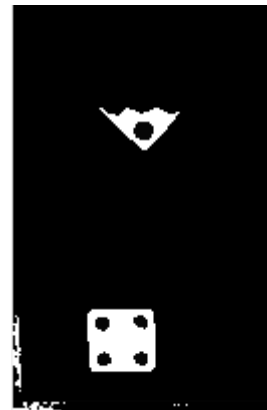
Obr. 15-2, Po thresholdu,
silné zastínění

8.2.2.2 Silná hrana procházející objektem

Druhým možným stavem je detekce části kostky, což je způsobeno průchodem hrany stínu jednou z kostek (obr. 16-1 a obr.16-2). V tomto případě je vyhodnocena celá osvětlená kostka a zároveň část neosvětlené.



Obr. 16-1, Původní obraz,
stín v objektu



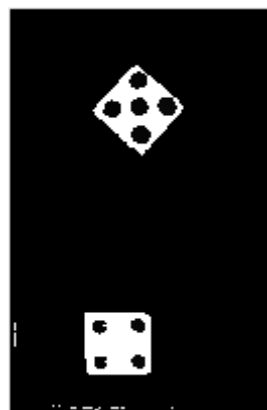
Obr. 16-2, Po thresholdu,
stín v objektu

8.2.2.3 Slabá hrana

Poslední možností je poté existence hrany rozdílu osvětlení, avšak tato hrana není tak výrazná, aby úspěšné detekci zastíněné kostky zabránila (obr. 17-1 a obr. 17-2). V tomto případě je součet vyhodnocen správně.



Obr. 17-1, Původní obraz,
slabé zastínění



Obr. 17-2, Po thresholdu,
slabé zastínění

9. Závěr

V bakalářské práci „Vestavný systém zpracování obrazu pro počítání objektů“ byly zpracovány všechny body zadání s výjimkou jednoho, a to audio výstupu, který bude zpracován dodatečně. Určení velikosti a pozic objektů je v rámci demonstračního zařízení využito, avšak tyto hodnoty nejsou uživateli z důvodu přehlednosti prezentovány. Další body, definované v průběhu řešení problému, byly zpracovány rovněž.

Práce podrobně analyzuje jednotlivé metody segmentace obrazu a nastiňuje jejich možnost pro využití za nízkých výpočetních a paměťových kapacit, které vestavné systémy nabízejí. Na základě této analýzy byly vybrány v zařízení využívané metody. Tyto metody rovněž prodělaly modifikace pro minimalizaci jejich náročnosti na dostupném mikrořadiči, zároveň pro zvýšení jejich robustnosti a pro zvýšení rychlosti vyhodnocení.

Silný důraz byl kladen na možnost aplikace zařízení v reálném prostředí, která byla zajištěna minimalizací závislosti zařízení na okolních světelných podmínkách. Této nezávislosti bylo docíleno dynamickou konfigurací doby expozice (a při extrémních podmínkách také gainu) využívaného obrazového senzoru.

Velký důraz byl rovněž kladen na robustnost zařízení. Té bylo docíleno již zmíněnou nízkou závislostí na vnějším osvětlení. Rovněž je posílena vhodnou modifikací využívaných metod pro autodetekci chybných stavů a implementací řešení takových problémů.

Jelikož je významnou součástí zařízení také interakce s uživatelem, byly využívané metody rovněž modifikovány pro zvýšení rychlosti, a to především předcházením aplikace časově náročnějších algoritmů na místa obrazu, kde by to jinak bylo zbytečné. Jako příklad této zrychlující modifikace budiž implementace kontroly před aplikací metody labelingu, zda má daná oblast potenciál být skutečně korektním uzavřeným objektem.

Zdroje:

[1] Mr. Salem Saleh Al-amri , Dr. N.V. Kalyankar, Dr. Khamitkar S.D: IMAGE SEGMENTATION BY USING EDGE DETECTION , publikováno: 2010, zobrazeno: květen 2017, dostupné z:

<http://www.enggjournals.com/ijcse/doc/IJCSE10-02-03-112.pdf>

[2] MT9V032 1/3-Inch Wide VGA CMOS Digital Image Sensor, publikováno: květen 2017, zobrazeno: květen 2017, dostupné z: <http://www.onsemi.com/pub/Collateral/MT9V032-D.PDF>

[3] Vojtěch Dokoupil, Zpracování obrazu vestavěným mikrořadičem, Bakalářská práce, publikováno: 2013

[4] STmicroelectronics, DM00037051 Datasheet, Revize: 8, zobrazeno: květen 2017, dostupné z:

<http://www.st.com/content/ccc/resource/technical/document/datasheet/ef/92/76/6d/bb/c2/4f/f7/DM00037051.pdf/files/DM00037051.pdf/jcr:content/translations/en.DM00037051.pdf>, 15.05.2017)

[5] STmicroelectronics, RM0090 Reference manual, Revize 14, zobrazeno: květen 2017, dostupné z:

http://www.st.com/content/ccc/resource/technical/document/reference_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00031020.pdf

[6] STmicroelectronics, Data brief DM00037955, Revize 6, zobrazeno: květen 2017, dostupné z:

http://www.st.com/content/ccc/resource/technical/document/data_brief/09/71/8c/4e/e4/da/4b/fa/DM00037955.pdf/files/DM00037955.pdf/jcr:content/translations/en.DM00037955.pdf

[7] HITACHI, Data sheet HD44780U (LCD-II), Revize 0.0, zobrazeno: květen 2017, dostupné z:

<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

[8] SAMSUNG, Data sheet S6A0069, Revize 0.0, zobrazeno: květen 2017, dostupné z:

<http://www.cloverdisplay.com/pdf/S6A0069.pdf>

[9] MathWorks, Matlab, obrázky distribuované společně s programem

[10] Vojtěch Dokoupil, Kamera CMOS a STM32F429 pro výuku, Diplomová práce, publikováno: 2015

Příloha A

Výsledné zařízení:



Obr. 18, Výsledné zařízení



Obr. 19, Senzor zapojen na kitu